MO71

IBM MQ for Windows GUI Administrator User Guide

Version 9.3.3

22nd April 2024

Paul Clarke

MQGem Software Limited
support@mqgem.com

Take Note!

Before using this User's Guide and the product it supports, be sure to read the general information under "Notices".

Ninety Eighth Edition, April 2024

This edition applies to Version 9.3.3 of *IBM MQ for Windows GUI Administrator* and to all subsequent releases and modifications until otherwise indicated in new editions.

(c) Copyright MQGem Software Limited 2012, 2024. All rights reserved.

Notices

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

MQGEM SOFTWARE LIMITED PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

The information contained in this document has not been submitted to any formal test and is distributed AS IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by MQGem Software for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

- The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:
  - IBM MQ
  - IBM
  - AIX
  - IBM i
  - MVS
  - z/OS

- The following terms are trademarks of the Microsoft Corporation in the United States and/or other countries:
  - Windows 95, 98, Me
  - Windows NT, 2000, XP, 7, 8, 10
  - Powershell
- Intel is a trademark of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Table of Contents

Notices iii
Figures xiii
Introduction xiv
Main changes from previous version xv
Chapter 1.IBM MQ GUI Administrator 1
1.1.Overview 1
1.2.Installation 2
1.3.MO71 Versioning 2
1.4.Running on Linux 3
1.4.1.Installing on Linux 3
1.4.2.Running under Linux 3
1.4.3.Minor Differences when running under WINE 4
1.4.4.Support 4
Chapter 2.Licensing 5
2.1.Licence File Location 5
2.2.Multiple licences 6
2.3.Licence Renewal 6
2.4.Changing your licence file 6
Chapter 3.Getting Started 7
3.1.Running the Administrator 7
3.1.1.Importing local Queue Managers 8
3.2.Adding a Queue Manager Entry 9
3.2.1.Connecting via a Client 9
3.2.2.Connecting via a different Queue Manager 10
3.2.3.Creating two locations for the same Queue Manager 10
Chapter 4.Issuing Commands 11
4.1.The Command window 11
4.1.1.Fields 11
4.1.2.Status window 12
4.1.3.Action buttons 12
4.1.4.Command Keys 12
4.1.5.List Windows 12
4.2.Dialog Examples 14
4.2.1.Displaying a list of Queues 14
4.2.2.Displaying Queues for multiple Queue Managers 15
4.2.3.Defining a Queue 17
4.2.4.Defining a Queue like another Queue 17
4.2.5.Starting a Channel 18
4.3.The Pop-up menu 19
4.4.Auto Refresh Dialog 21
4.4.1.Auto Refresh Export Fields 21
4.5.Alter list dialog 23
4.5.1.MULTIQM or QSGQMNAME field 23
4.6.Hostname Support 24
4.6.1.DNS System Values 24
4.6.2.DNS User Values 25
4.7.MQSC Window 26
4.7.1.MQSC Examples 27
4.7.2.MQSC Clipboard 27

Chapter 5.Filtering....28
5.1.Column Filters....28
5.1.1.Column Filter Types....29
5.1.2.Column Filtering Positioning....30
5.1.3.Date&Time Column Filters....30
5.1.3.1.Now....30
5.1.3.2.Local Time....31
5.2.Filter Expressions....31
5.2.1.List Variables....33
5.2.2.Operators & Flow control....34
5.2.3.System Variables....35
5.2.4.User Variables....36
5.2.5.User Variable Inserts....36
5.2.6.Functions....37
5.2.6.1.Function - usrcol(Name, Value, Options, MaxValue, Colour1, Colour2, FgColour, BgColour)....44
5.2.6.2.Positioning your user column....49
5.2.6.3.Using usrcol() without displaying the column....50
5.2.6.4.Using usrcol() in practice....50
5.2.7.Output Format String....51
5.3.Filter Manager....52
5.3.1.Comments....52
5.3.2.Shared Filters....53
5.4.Filter Examples....54
5.5.Filter Troubleshooting....59
Chapter 6.Queue Manipulation....60
6.1.Browse Queue....60
6.1.1.Message Selection....60
6.2.Browse Message....61
6.3.Message Detail....62
6.4.Fields....63
6.4.1.Display Range....63
6.4.2.Selector....63
6.4.3.Search....63
6.4.4.Max Message Size....63
6.4.5.Target Queue Manager....64
6.4.6.Target Queue....64
6.5.Pop-up Menu....64
6.5.1.Next/Prev....64
6.5.2.Display Format....64
6.5.3.Detail Level....65
6.5.4.Copy to clipboard....65
6.5.5.Copy shown to clipboard....65
6.5.6.Copy all to clipboard....65
6.5.7.Edit....65
6.5.8.Copy....65
6.5.9.Move....65
6.5.10.Delete....65
6.5.11.Put Message....65
6.5.12.Load/Unload Messages....65
6.5.13.New....65
6.5.14.Convert....66
6.5.15.Message Summary....66
6.5.16.Properties....66
6.5.17.Strip Headers....66
6.5.18.Ignore CR/LF....66
6.5.19.Multi Transaction....66
6.5.20.Search Offset....66

6.5.21.Message Selection....66
6.5.22.Context....67
Chapter 7.Copying, Moving and Deleting messages....68
7.1.Browsing the queue....68
7.2.Queue load/unload facility....68
7.2.1.File Format....71
7.2.1.1.Example - Changing the user ID....71
7.2.1.2.Attribute Format Reference....72
7.2.1.3.Recognised file formats....72
7.3.Progress Dialog....73
7.4.Long Operations....73
Chapter 8.Naming Objects....74
8.1.Object Name Patterns....75
8.1.1.Global Object Name Patterns....75
8.1.2.Global Object Name Pattern Specification....76
8.1.3.Local Object Name Pattern Specification....77
8.2.Object Define Name Checking....77
8.3.Object Name Health Check....77
8.4.Object Name Matching....78
Chapter 9.Network View....79
9.1.Network Menu....79
9.1.1.File....79
9.1.2.Commands....79
9.1.3.Action....79
9.1.4.View....80
9.1.5.Network Layout....80
9.2.Network Display....81
9.2.1.Moving and Selecting Objects....81
9.2.2.Display Format....81
9.3.Verify Display....83
9.3.1.Verify Display Search....84
9.3.2.Automatic Search Wildcards....84
9.3.3.Case Sensitivity....84
9.3.4.Search Qualifiers....84
9.4.Queue Managers Display....85
9.5.Levels Display....85
9.6.Problem Selection....86
9.7.Network Names....88
Chapter 10.Application Monitoring....89
10.1.Application Monitoring Configuration....89
10.2.Application Definition....90
10.3.Channel Mask Processing....92
10.3.1.Channel Mask Priority....92
10.4.Application Group Record....93
10.5.Application Object Record....94
10.6.Application Record Actions....95
10.6.1.Application connections dialog....95
10.7.Application View....96
10.7.1.Moving and Selecting Objects....97
10.7.2.Application View Menu....97
Chapter 11.General Actions....98
11.1.Dates and Times....98
11.1.1.Recent Dates....98

11.2.Units *99*
11.2.1.Types *99*
11.2.2.Style *100*
11.2.3.Location *100*
11.2.4.Data Entry *100*
11.2.5.Special Value *100*
11.3.Exporting Definitions *101*
11.3.1.Export MQSC format *102*
11.4.MQ Maintenance *103*
11.4.1.Display latest MQ Maintenance levels *103*
11.4.2.Check latest MQ Maintenance for a Queue Manager *103*
11.4.3.Network View Problem Analysis *103*
11.5.Splash Screens *103*
11.5.1.Configuration *104*
11.5.2. Location and Size *104*
11.6.Master Configuration File *105*
11.7.Licence Command *106*
11.7.1.Command Inserts *106*
11.7.2.Command File Exit Code *106*
11.8.Printing *107*
11.8.1.Margins *108*
11.8.2.Font *108*
11.8.3.Title, Header, Footer *108*
11.8.4.Printing Lists *108*
Chapter 12.Queue Manager Monitoring **109**
12.1.Command Strings *109*
12.1.1.Format *110*
12.1.2.Substitution Characters *110*
12.1.3.Examples *110*
Chapter 13.Attribute Monitoring **111**
13.1.Monitors *111*
13.1.1.Monitor Fields *111*
13.1.2.Actions *112*
13.2.Monitor Status *113*
13.2.1.Actions *113*
Chapter 14.Graphing **114**
14.1.Graph Options Dialog *115*
14.1.1.Data Sets *115*
14.1.2.Pie Charts *119*
14.1.3.Graph *119*
Chapter 15.MQEV Event Monitoring **123**
15.1.Licensing *123*
15.2.Administering MQEV *123*
15.3.Viewing MQEV data *123*
15.3.1.Multiple dialog fields *124*
15.3.2.Date Range selection *124*
15.3.3.Summing data and other selection controls *125*
15.4.Graphing your MQEV data *125*
15.5.Menu visibility *125*
Chapter 16.Direct Event Monitoring **126**
16.1.Event Example *126*
16.1.1.Queue Manager Name *126*
16.1.2.Event Queue Name *126*
16.1.3.Type *126*

16.2.Event Fields....127
16.2.1.Queue Manager Name....127
16.2.2.Location....127
16.2.3.Type....128
16.2.4.Filter....128
16.2.5.Filter Priority....128
16.2.6.Discard....128
16.2.7.Command....128
16.2.8.Log Line....128
16.2.9.Requeue....128
16.2.10.Log File....129
16.2.11.Log Event....129
16.2.12.Auto Start....129
16.2.13.Console Priority....129
16.2.14.Console Type....129
16.3.Operational Considerations....129
16.4.User Events....130
16.4.1.Operation....130
Chapter 17.API Exerciser....132
17.1.Main Layout....132
17.2.Walk-through....133
17.2.1.MQCONN....133
17.2.2.MQOPEN....133
17.2.3.Object Descriptor (MQOD)....134
17.2.4.Using other connection handles....134
17.2.5.Opening other object types....134
17.2.6.MQPUT....135
17.2.7.Message Descriptor (MQMD)....135
17.2.8.Put Message Options (MQPMO)....135
17.2.9.Message Buffer....135
17.2.10.MQGET....136
17.2.11.Message Descriptor (MQMD)....136
17.2.12.Get Message Options (MQGMO)....136
17.2.13.Message Buffer....137
17.3.Walkthrough with multiple instances....137
17.3.1.MQSUB....137
17.3.2.Subscription Descriptor (MQSD)....137
17.3.3.Using MQCHARV data types....138
17.3.4.Creating a Message handle....138
17.3.5.MQGET from a handle returned from MQSUB....138
17.3.6.MQPUT1....138
17.3.7.MQINQMP....139
17.3.8.MQCLOSE....139
17.4.MQCONNX....140
17.4.1.Channel Definition (MQCD)....140
17.4.2.SSL Configuration Options (MQSCO)....141
17.4.3.Connection Security Parameters (MQCSP)....141
17.5.Asynchronous Consume....142
17.6.The MQI Verb set....143
17.7.Advanced Mode....144
17.7.1.MQCONN....144
17.7.2.Connection Handle fields....144
17.7.3.Options fields....144
17.8.Troubleshooting....144
17.8.1.Why are my object name drop-down lists empty?....144
17.8.2.Why is the object handle I just opened not in the drop-down list?....144

17.8.3.Why is the queue I just defined not in the drop-down list?....144
Chapter 18.Comparing Queue Manager Objects....145
18.1.Synchronise....146
18.2.Queues....147
18.3.Channel Authentication....147
18.4.Subscriptions....147
Chapter 19.Trace Message....148
19.1.Commands....149
19.2.Send....149
19.3.Open <object type>....149
19.4.Display Type....149
19.5.Display Options....150
19.6.Show Pie Chart....150
19.7.Pie Chart Labels....151
19.8.Export....151
19.9.Copy text to Clipboard....151
19.10.Delete....152
19.11.Limitations....152
Chapter 20.Web Access....153
20.1.Getting Started....153
20.2.Examples URLs....154
MQ Commands....155
20.2.1.MQEV Commands....156
20.3.Dialog Fields....157
20.3.1.Examples....157
20.4.HTML Files....158
20.4.1.HTML Inserts....159
20.4.1.1.%[qmlist] HTML Insert....160
20.5.Using HTTPS....163
20.5.1.Trying it out with a self-signed certificate....163
20.5.2.Advanced Configuration....164
20.6.Limiting what is displayed....165
20.7.Who is connected ?....165
20.7.1.Limiting connections....165
20.8.Running MO71 as a service....167
20.8.1.Running MO71 in background mode....167
Chapter 21.Predefined Dialogs....168
21.1.Definition....168
21.2.Menu....171
21.3.Refresh Group....171
Chapter 22.User Commands....172
22.1.User Command Fields....172
22.2.Command Inserts....174
Chapter 23.Activity Trace....175
23.1.Configuration....175
23.2.Switching Tracing on and off....175
23.3.Activity Trace Dialog....176
23.3.1.Input....177
23.3.2.Filter....177
23.3.3.Output....179
23.3.4.API Selection....179
23.3.5.Actions....179

23.3.6.Status Tab....180
23.3.7.Output Tab....181
23.3.8.Health Tab....182
23.3.9.Graph Tab....184
23.3.9.1.Selection Fields....184
23.3.9.2.Graph....184
23.3.10.Context Menu....185
Chapter 24.Customization and Reference....187
24.1.Menu options....187
24.1.1.File....187
24.1.2.Commands....189
24.1.3.Action....189
24.1.4.View....190
24.1.5.Monitor....192
24.1.6.Windows....192
24.2.Preferences Dialog....193
24.2.1.General....193
24.2.2.Dialogs....194
24.2.3.Lists....198
24.2.4.Browse....200
24.2.5.Connection....201
24.2.6.Network....202
24.2.7.Network UI....204
24.2.7.1.Icons....204
24.2.7.2.Media Titles....204
24.2.8.Applications....205
24.2.9.Application Icon....206
24.2.10.Export....206
24.2.11.Naming....206
24.2.12.Sounds....207
24.2.13.Display....207
24.2.14.Events....208
24.2.15.Console....209
24.2.16.Units....209
24.2.17.Time....209
24.2.18.HTTP....210
24.2.18.1.General....211
24.2.18.2.IP Address List....212
24.2.18.3.Proxy Settings....212
24.2.18.4.HTTP Server buttons....212
24.2.19.Graph....212
24.2.20.Help....213
24.3.Location Dialog....214
24.3.1.Connection....214
24.3.2.Security....215
24.3.3.Time....216
24.3.4.Options....216
24.3.5.Monitoring....218
24.3.6.Export....219
24.3.7.Pub/Sub....221
24.3.8.Network UI....221
24.3.9.Applications....221
24.3.10.Naming....221
24.4.Password Cache File....222
24.4.1.Configuration....222
24.4.2.Specifying the Cache File Name....223
24.4.3.Cache File Format....223
24.5.Key Database File Password....225

24.6.Container Windows ... 226
24.6.1.Keyboard Control ... 226
24.7.Text Inserts ... 227
24.7.1.Conditional Text Inserts ... 228
24.8.File Name Inserts ... 228
24.9.Reply Queue Details ... 229
24.9.1.Model Reply Queue ... 229
24.10.Colour Dialog ... 229
24.10.1.Colour Schemes ... 229
24.10.2.Extended Selection ... 229
24.11.Multiple Location Change Dialog ... 230
24.12.Time Interval Format ... 232
24.13.MO71 Parameters ... 233
Chapter 25.Usage Tailoring ... 235
25.1.Object Tailoring Keywords ... 236
25.2.Menu Tailoring Keywords ... 238
25.2.1.File Menu ... 238
25.2.2.Action Menu ... 239
25.2.3.View Menu ... 239
25.2.4.Monitor Menu ... 239
25.3.Dialog Menus ... 240
25.4.Miscellaneous Tailoring Keywords ... 240
Chapter 26.Security ... 241
26.1.Example security commands for Distributed Platforms ... 241
26.2.Example security commands for z/OS (using RACF) ... 241
Chapter 27.Troubleshooting ... 242
27.1.Troubleshooting interacting with IBM MQ ... 242
27.1.1.The Command server for the Queue Manager is not running ... 242
27.1.2.The channels to and from the remote Queue Manager are not running ... 242
27.1.3.The user does not have security access to the Queue Manager ... 242
27.1.4.Message "MO71 configuration file version is newer than program" ... 242
27.1.5.Message "Problem not checked because of incomplete definitions" ... 243
27.1.6.An SSL/TLS connection fails and MO71 shows a Message Box ... 243
27.2.Troubleshooting Web Access ... 243
27.2.1.Twisties not working ... 243
27.2.2.Navigation twisty does not surround all its queue managers ... 244
27.2.3.HTTPS setup ... 244
27.2.4.HTTPS connectivity ... 244
27.2.5.sChannel function 'AcquireCredentialsHandle' failed RC(0x00000057) The parameter is incorrect ... 244
27.3.Service ... 245
27.3.1.Mini-Dump ... 245
Chapter 28.Changes made in previous versions ... 246
Changes made in Version 9.3.2 ... 246
Changes made in Version 9.3.1 ... 246
Changes made in Version 9.3.0 ... 247
Changes made in Version 9.2.3 ... 247
Changes made in Version 9.2.2 ... 248
Changes made in Version 9.2.1 ... 248
Changes made in Version 9.2.0 ... 249
Changes made in Version 9.1.4 ... 250
Changes made in Version 9.1.3 ... 250
Changes made in Version 9.1.2 ... 250
Changes made in Version 9.1.1 ... 251

Changes made in Version 9.1.0 ... 251
Changes made in Version 9.0.6 ... 252
Changes made in Version 9.0.5 ... 252
Changes made in Version 9.0.4 ... 252
Changes made in Version 9.0.3 ... 253
Changes made in Version 9.0.2 ... 254
Changes made in Version 9.0.1 ... 254
Changes made in Version 9.0.0 ... 255
Changes made in Version 8.0.4 ... 255
Changes made in Version 8.0.3 ... 256
Changes made in Version 8.0.2 ... 257
Changes made in Version 8.0.1 ... 257
Changes made in Version 8.0.0 ... 258
Changes made in Version 7.5.2 ... 258
Changes made in Version 7.5.1 ... 260
Chapter 29.Migration from previous versions ... **261**
Migrating from a version prior to Version 9.3.3 ... 261
Migrating from a version prior to Version 9.3.2 ... 261
Migrating from a version prior to Version 9.3.1 ... 261
Migrating from a version prior to Version 9.2.3 ... 261
Migrating from a version prior to Version 9.2.1 ... 261
Migrating from a version prior to Version 9.2.0 ... 262
Migrating from a version prior to Version 9.1.3 ... 262
Migrating from a version prior to Version 9.1.2 ... 262
Migrating from a version prior to Version 9.1.1 ... 262
Migrating from a version prior to Version 9.0.3 ... 262
Migrating from a version prior to Version 9.0.0 ... 263
Migrating from a version prior to Version 8.0.3 ... 263
Migrating from a version prior to Version 8.0.2 ... 264
Migrating from a version prior to Version 8.0.0 ... 264
Migrating from a version prior to Version 7.5.2 ... 265
Migrating from a version prior to Version 7.5.1 ... 266
Appendix A: Icons ... **267**
A.1. Queue Manager Icons ... 267
A.2. Queue Icons ... 267
A.3. Channel Icons ... 268
A.4. Other Object Icon ... 268
A.5. Network View Icons ... 269
A.6. Container Icons ... 269
A.7. User Icons ... 269
Appendix B: Constants ... **270**
Appendix C: BB Codes ... **274**
Appendix D: Filter function 'set' - Field names ... **275**
Appendix E: Display National Language Characters ... **276**

Figures

Figure 1: Main Window....7
Figure 2: Add Location Dialog....9
Figure 3: Queue List Dialog....14
Figure 4: Multi Queue Manager List....15
Figure 5: Queue Dialog....17
Figure 6: Channel List Dialog....18
Figure 7: Alter List Dialog....23
Figure 8: MQSC Window....27
Figure 9: Filter manager Dialog....52
Figure 10: Queue List with colour filter....55
Figure 11: Queue List with cell colour filter....56
Figure 12: Message List Dialog....60
Figure 13: Message Dialog....61
Figure 14: Message Detail....62
Table 1: Meaning of column one symbol in file format....70
Table 2: Message descriptor attribute representations....71
Figure 15: Network Window....78
Figure 16: Verify Window....82
Figure 17: Queue Managers Display....84
Figure 18: Network Levels Display....84
Figure 19: Problem Selection Dialog....85
Figure 20: Problem Description Dialog....86
Figure 21: Print Dialog....106
Figure 22:File Name Inserts....227
Figure 23:Icons....268
Figure 24:Codepages....275
www.mqgem.com....276

Introduction

This program was originally written as a monitoring program for the 1996 Olympic Games in Atlanta. The requirement was for an OS/2 PM program which would actively monitor a number of remote Queue Managers and give visual and audible feedback if a problem was detected. At the time I was interested in gaining experience of PCF (Programmable Command Format) messages. Since the program already had the basic structure required I decided to add command support to it allowing you to issue virtually any MQ command against any number of Queue Managers in a dialog.

For many years the program was enhanced in my spare time and released as SupportPac MO71. In October 2012, after working at IBM for 28 years on product development, I decided to leave IBM to pursue a different career writing tools and utilities for IBM MQ. To that end I founded MQGem Software (www.mqgem.com). IBM agreed that it was in customers best interests for the program to continue to be developed and in August 2013 IBM agreed to let me continue my work on MO71. Future support and development of the program will therefore be made by MQGem Software and all support questions should be directed to support@mqgem.com. Since 2013 we have extended MO71 greatly and added many features which were not feasible to do when it was just a spare time activity.

We hope you enjoy the product and agree with us that is makes many MQ tasks much simpler and easier. Of course I could not have written such a program without help from a variety of people.

My thanks go to Morag Hughson for considerable assistance with this manual, for numerous usability suggestions, for keeping me sane when nothing ever seems to go right, as well as tirelessly testing many buggy versions of the program!

My thanks to Jan Fluitsma for providing "*Displaying National Language Characters*".

My thanks to Stefan Raabe for his many suggestions and testing of the programmable filters.

I would also like to express my gratitude to the many people over the years who have written to me to express their thanks for the program and make suggestions. It makes all the difference in the world to know that people are using the program and finding it useful. When I first wrote the program there were few alternatives to MQSC for Queue Manager Administration but now there are a number of solutions available. We will continue enhancing the program while ever our customers tell us it is useful to them.

Throughout this book look-out for boxes such as this. These contain a link which takes you to our blog with filtering applied to show you further examples and explanations of the topic at hand.

Main changes from previous version

Unless otherwise stated the behaviour of the previous version should be maintained and, if all goes well, enhanced. While every effort is made to try and ensure that there are as few bugs as possible it would be surprising if some problems didn't leak out. We would recommend that users keep their previous version of MO71 handy so that they can revert to it if a bug is found. Needless to say, please report any incompatibilities to us if they are found and we will do our best to provide a fix.

The full list of changes are:

1. **Improved support for MQEV**
   - Addition of MQEV commands to the Predefined Dialog list
   - Standard MQEV dialogs can now show multi-Queue Manager displays
   - The addition of a MQEV List dialog allowing you to see the status of all your MQEVs in one dialog
   - MQEV Accounting and Statistics list dialogs can be invoked via context menus taking the queue or channel selected
   - MQEV Dialogs can now be exported in MQSCX format
   - Ability to have MQEV alerts routed to the MO71 console
     For more information please see the description in the Options tab of the Location Dialog.
2. **Commands menu added to all Queue Manager dialogs**
   So, for example, you can start the Channel List dialog directly from the Queue List dialog
3. **New Console fields 'category' and 'text' have been added**
   - In conjunction the csl() function has been altered to add these new fields - see page 38.
   - The MO71Event program has also been enhanced to add these two fields.
4. **Enhancements to Predefined Dialogs**
   - **Changes to the Predefined Context Menu field**
     Previously you could ask for the dialog to be included or not ie. **YES** or **NO**. In this MO71 version you have the additional choices of **MAIN**, **QMGR**, **OBJECT** or **PREFIX** allowing the choice to invoke predefined dialogs with an object name as part of the context.
   - **Addition of Context Level**
     By default predefined dialogs, when shown on a context menu, are shown in a folder as a sub-menu off the commands menu. Choose to position your most commonly used ones on a higher up part of the context menu with this field.
   - **Addition of Title control**
     Chose what you want the window title to be when a predefined dialog is shown. For example, to differentiate between lists of the same object type with different filters, e.g. transmission queues, and Dead-letter queues.

   For more information on all of these see 21.1. Predefined Dialogs Definition on page 168
5. **User Command changes**
   - **An additional 'Object Type' field**
     Therefore you can both show and invoke user commands based on the type of object selected.
   - **An additional 'Keep Window' option.**
     If you specify YES then command window which starts will not be ended after completion of the command.
   - **Addition of Context Level**
     As above for predefined dialogs.

   For more information on all of these see 22.1.User Command Fields on page 172.
6. **New export time formats**
   You can now have an export date format of something like 23-Feb-2024.

7. **Licence reminder added to the console.**
   Whenever **MO71** issues a licence reminder pop-up dialog it will also add an entry to the console. If a new licence is installed, with a longer expiry interval, then the console entry will be removed.

8. **=mqev** and **=mqsc** supported in MQSC window

Chapter 1. IBM MQ GUI Administrator

This document describes the functions available in the program.

1.1. Overview

The main window displays a list of Queue Managers. Typically, this list would be the local Queue Manager and a number of other 'remote' Queue Managers that the machine has access to via MQ channels. The user can then perform the following functions against the Queue Managers in the list:-

1. **Issue commands**
   By selecting a Queue Manager in the list and invoking one of the Command menu options the user can issue commands against that Queue Manager. Note that the Queue Manager can be any platform that has a command server (including z/OS). Commands may also be entered in MQSC format from the MQSC window described later.

2. **Filtering**
   When dealing with a list of objects as the result of an issued command, it can be useful to filter out some of these objects to reduce the size of the list displayed.

3. **Queue Manipulation**
   The queue being accessed can be either on the local Queue Manager or a remote Queue Manager provided a program like MO71 is running on the remote Queue Manager. Connecting to the Queue Manager as a client will allow queue manipulation on any remote Queue Manager since, as far as MO71 is concerned, the Queue Manager is local.

4. **Communication**
   By selecting Talk from the menu, a dialog is shown which allows the user to type a text message that can be sent to the target Queue Manager. The receiving Queue Manager must be running the monitoring program, in which a dialog shows the text that the original Queue Manager sent.

5. **Monitoring Queue Managers**
   With monitoring enabled, the program will periodically send 'monitor' messages to any enabled Queue Manager. You can monitor a Queue Manager that the application is directly connected to but it is more useful to monitor a remote Queue Manager. This essentially involves sending a message to a named queue ‘the monitor queue’ on the remote machine and waiting for the message to be returned.

6. **Monitoring Queue Manager objects**
   IBM MQ provides quite a number of 'event' messages which are generated when certain key events occur during the life of the Queue Manager. For example, an event message can be generated when the depth of a queue reaches a certain threshold value. IBM MQ itself does not act upon these messages, they are purely generated for some monitoring tool to receive and perform an appropriate action. The appropriate action will vary depending on the installation, the object involved and the event type generated.
   MO71 can be configured to read these event messages from the event queues and can be configured to perform a variety of actions such as issue commands or write to a log file

7. **Monitoring Applications**
   Periodically monitor what applications are running and how many connections then have. The application data can be displayed in a dialog list or in an application view diagram.

8. **Try out MQI verbs**
   The API Exerciser allows the user to try out most of the MQI verbs (MQCONN, MQOPEN, MQPUT, MQGET etc) and see their behaviour.

1.2. Installation

Create a directory, say MO71, and copy the installation zip file into it. This is probably called mo71.zip. You now need to unzip this file; you should be able to use any of the regular zip utilities to do it. For example, in Windows Explorer you should be able to right click on the file and select 'Extract All..'.

Once you have unzipped the file you should end up with a directory containing the following:

File	Description
mqmonntp.exe	The actual MO71 program itself
mo71start.cmd	A simple command file for starting the program
mqmon.pdf	The MO71 manual....probably what you are now reading
mo71event.exe	A program which enables the user to raise an MO71 user event to have text displayed on the MO71 console.
html	A directory containing HTML files. These files are only used if you choose to allow access to MO71 from browsers.

MQMONNTP is the main executable of the MO71 program. Note that the directory where you place the .EXE program is the default location where the program will store its configuration file MQMON.CFG and the object definition file MQMON.DFN. You generally need not worry about these files but you should be aware that they are written and therefore the program needs write access to the directory. Alternatively you can use the '-f' parameter to the program to give the directory name the program should use for these files.

The MO71 program will dynamically load either the MQ client or MQ Queue Manager libraries as required. So, in order to successfully connect to a Queue Manager at least one of these IBM MQ products must be installed. The windows IBM MQ client is available for free download at:-

https://www-01.ibm.com/support/docview.wss?uid=swg24044791[1]

1.3. MO71 Versioning

Before we start it is probably worth saying a few words about MO71 versioning. The version of MO71 and the version of IBM MQ need not be the same but they are related. Essentially the first two digits of the MO71 version tells you the level of MQ that MO71 understands. So, for example, both MO71 version 8.0.0 and 8.0.1 know about the MQ objects and their definitions up to the level of MQ 8.0. When IBM brings out a new version of MQ, let's say MQ 8.1, then MO71 8.0.1 will almost undoubtedly run on that version but it will not know about any new objects or attributes that have been introduced. If MO71 detects a Queue Manager which is at a later version than itself it will present a warning dialog. You may choose to ignore this dialog but be aware that some objects or attributes of objects may not be displayed.

For this reason it is always advisable to run a version of MO71 where the first two digits are equal to or greater than the version of MQ you are running. So, to reiterate, you can run any version of MO71 on virtually any version of IBM MQ. The later the version of MO71 the more features you will have. It is perfectly reasonable to run MO71 8.0.1 against MQ 5.3 or 6.0. The only thing you should try and ensure is that the version of MO71 you are using is late enough to understand all the MQ Objects and their attributes. And the simplest way to check that is to check the first two digits of the version.

Of course it is always recommended to use the latest version of MO71 available since a new version will almost always mean new features. You can even get MO71 to tell you when a new version has become available by setting the value of 'Notify of program updates' in the General tab of the Preferences Dialog.

[1]Note that there may be a more recent version of the MQ client (or indeed an earlier version) available on the MQ SupportPac web sites. It is recommended to use the latest version possible, however, MO71 should work with any version.

1.4. Running on Linux

Although MO71 is a Windows program it is possible to run it on Linux using WINE. WINE is a fabulous piece of free software that comes from the WINE project (https://www.winehq.org/). This software allows a huge range of Windows applications to run, virtually unchanged, on an Intel Linux machine[2]. The software is not an emulator but rather translates Windows API calls to POSIX calls on the fly which means that it is fast and efficient. The level of compatibility is very good and it is well supported with regular updates.

1.4.1. Installing on Linux

Installing MO71 on Linux is very easy. You just need to follow the following steps:

- **Installing WINE**
  MO71 is a 32-bit Windows program and as such requires the 32-bit WINE package to be installed. There is a good chance that your Linux machine already has WINE installed but if it doesn't a simple command such as the following will likely do the job.

  ```
  sudo yum install wine.i686
  ```

  Of course, as is the nature with Linux, different distributions are often slightly different. Refer to your system documentation and the WINE documentation to discover the best way to install WINE.

- **Installing MO71**
  Download or copy the MO71 zip file to your chosen Linux directory. Now unzip the file making sure to maintain the zip file directory structure.

- **Installing the IBM MQ interface libraries**
  MO71 is a Windows program and yet we need it to call the Linux MQ libraries (libmqm_r.so or libmqic_r.so). This is done by installing MQ wrapper libraries (mqm.dll.so and mqic32.dll.so). These libraries can be downloaded from www.mqgem.com. They are contained in a zipped tar file **mq_wine.tgz**. Untar the file using a command such as

  ```
  tar -xvzf mq_wine.tgz
  ```

  You need to put these libraries where WINE will find them. Your two choices are:

  - Put them in /usr/lib/wine
  - Put them where ever you like but ensure that the path is added to WINEDLLPATH environment variable.

1.4.2. Running under Linux

Assuming that you have followed all the advice in 'Installing on Linux' then invoking MO71 should be a simple matter of entering the command:

```
wine mqmonntp [ MO71 parameters ]
```

Once you are comfortable that MO71 starts and runs correctly you may want to set up an icon on your desktop to issue this command and possible run it in the background.

File Names

When running on Linux you can either use the normal Linux way of referring to OS files or you can use the Windows file format as described in the WINE File Management documentation.

[2]It also allows Mac OSX and BSD.

1.4.3. Minor Differences when running under WINE

For the most part running MO71 on Linux under WINE will have very few differences in function over running it on Windows. Listed here are any deliberate minor differences:

- When using a re-distributable client, the location of the error log and ini file are found in $HOME/IBM/MQ/data. Unfortunately the HOME environment variable is not propagated into the WINE environment (along with PATH, SYSTEM and TEMP[3]). If you don't have MQ_DATA_PATH set, MO71 will attempt to use /home/$USERNAME/IBM/MQ/data. If this is not the correct place to find your error log and ini file, please set the MQ_DATA_PATH environment variable.

1.4.4. Support

Each version of Windows behaves slightly differently and WINE is just the same. Although the folks at WINE have done a fabulous job of implementing the Windows interface there are bound to be subtle differences. In some cases these differences can be allowed for in the MQGem products and in some cases they will surface as a different behaviour. However, in general WINE can be treated by MQGem as just another Windows release. If you see a behaviour that you believe is not correct then by all means send us an email and we'll see what we can do to fix the problem. If necessary we can also contact the WINE development team and ask them to look in to the issue. Look on our web site for the current list of known problems.

[3]See https://www.winehq.org/docs/wineusr-guide/environment-variables

Chapter 2. Licensing

To access all the features of **MO71** you will require a licence file. The licence file also entitles the user to email support. If you would like to try out **MO71** for free then a 1-month trial licence can be obtained by sending a note to support@mqgem.com.

Each licence is for a certain period of time, usually one year rather than for a particular version of **MO71**. There are number of advantages of this scheme:

- **Customers of MO71 get more dedicated support**
  Resources can be targeted towards customers who have made a financial contribution to the development of **MO71**.

- **Purchasing decision is simpler**
  The **MO71** licence covers a period of time not a release. It is therefore not necessary to concern oneself about whether a bigger, better version is about to come out soon since whatever licence you buy now will also work for that version.

- **Features are available sooner**
  Using this model it is not necessary for us to collect a large group of features together to 'justify' a new release of **MO71**. Instead a new release can be made available whenever a new feature is added which is regarded as sufficiently useful since all current users will be able to migrate to the new version at no cost to themselves.

There are five types of licences which allow different levels of flexibility about who can run the program. Essentially this is controlled by the presence, or not, of userid, machine or location fields.

Type	Fields Set	Description
Emerald	userid, machine	MO71 is only supported on one machine using one userid. One concurrent browser connection is supported.
Ruby	machine	MO71 is supported by any number of users using it on the same machine. Up to 5 concurrent browser connections is also supported
Sapphire	userid	MO71 is supported on any number of machines using the same userid. Up to 5 concurrent browser connections are supported
Diamond	location	MO71 is supported by any number of users at the same site on any set of machines. The location field gives the location, for example “London, England” of where the licence is based. Unlimited concurrent browser connections are supported.
Enterprise	None	MO71 is supported across your whole enterprise. This means any number of users at any number of locations. An Enterprise licence can be bought by purchasing just 3 Diamond licences.

Depending on the licence type the Userid and Machine values must match the environment where **MO71** is running. At any time, even without a valid licence, you can run MO71 and bring up the About box by selecting '***File → About...***' from the main menu. This will show a dialog with information about the version of the program, the userid and machine name it is running on and information about the current status of the licence. If you are about to buy a non-Diamond licence you should use these values in your purchase order to ensure that the licence will work in your environment.

2.1. Licence File Location

If a licence file is bought you will be sent an **mqgem.lic** file. All you need to do is place this licence file in the same directory as the mqmonntp.exe program. It is not necessary to end the program when installing a new licence file.

Alternatively you can set environment variable MQGEML to point to the directory where the licence file can be found.

2.2. Multiple licences

If you have multiple licences then they can be concatenated into a single **mqgem.lic** file. This can be done using simple OS commands such as **copy** or by using your favourite editor. Ensure that all lines are copied in their entirety, including the carriage return character at the end. Blank lines can be added to the file as required.

2.3. Licence Renewal

Licence renewal is never automatic. **MO71** will start issuing warning messages as you get close to the licence expiry date. These reminders will be in the form of pop-up dialogs if running in interactive mode or will be written to the background file if running in background mode. **MO71** will also add a reminder line to the console display.

By default MO71 will start to remind the user about an expiring licence some 30 days before it expires. However, you can adjust this interval, if you like, using the environment variable, MQGEMLR. Set this environment variable to the number of days notice you want prior to expiry. For example **MQGEMLR=60** will request that MO71 starts to notify of licence expiry 60 days before it expires. Setting the value to 0, **MQGEMLR=0**, will stop MO71 issuing any licence expiry warnings. If you choose to do this, make sure you have some other way to remember when your licence will expire, to ensure that you have continuous licence coverage.

A new licence can be purchased at any time and a new licence file will be sent which extends the current licence by whatever period has been purchased. There is therefore no concern about losing time by renewing early.

It is no longer necessary to end the program to pick up the new licence. Just replace the **mqgem.lic** file and within the next 24 hour period **MO71** will notice, the About box will be updated, and licence reminders will stop.

2.4. Changing your licence file

The licence file is a simple text file. Generally speaking if you change the contents of the licence file you will invalidate it and it will cease to work. However, there are some minor changes you can make if you wish. Naturally it is always recommended that you keep a copy of the original unchanged file.

- You can change the case of any of the values.
- You can add or remove white space such as blanks
- You can add or change any lines which start with '*' since these are comment lines.

Remember that more than one licence can be contained within the single **mqgem.lic** file if required.

Chapter 3. Getting Started

This section will explain how to run the Administrator program and provide an example to add the first Queue Manager to be administered.

3.1. Running the Administrator

All that is needed to run the Administrator is to run the mqmontp.exe program. However, you do need to ensure that the correct MQ environment is set-up before the program is run. If you have installed MQ in the non-default location or you have multiple MQ installations then you will need to ensure the right installation is in effect before you run the mqmontp.exe program.

There are essentially two ways you can start MO71 depending on your MQ installation

1. **If you have a single MQ installation in the default location**
   You should be able to run MO71 by issuing the command '**mqmonntp**' or double clicking on the **mqmonntp.exe** in the Explorer window.
2. **If you have multiple MQ installs or MQ is not at the default location**
   In this case you have to set-up the right MQ environment before calling MO71. The simplest way of doing this is to call the MQ command 'setmqenv'. To this end we have provided you with a simple command file called **mo71start.cmd**. Edit this command file to change the installation directory to the location of MQ installation you want to MO71 to use. You should now be able to start MO71 by issuing the command '**mo71start**' or double clicking on the '**mo71start.cmd**' in the Explorer window.

Once you have verified that your chosen method correctly starts MO71 then you may find it convenient to add a desktop or Quick launch icon. This is achieved very simply by dragging either the mqmonntp.exe or mo71start.cmd to the location where you wish there to be a start icon.

The program does not require any parameters but you can pass a Queue Manager location in on the command line if you wish. See *"MO71 Parameters"* on page 233 for more information.

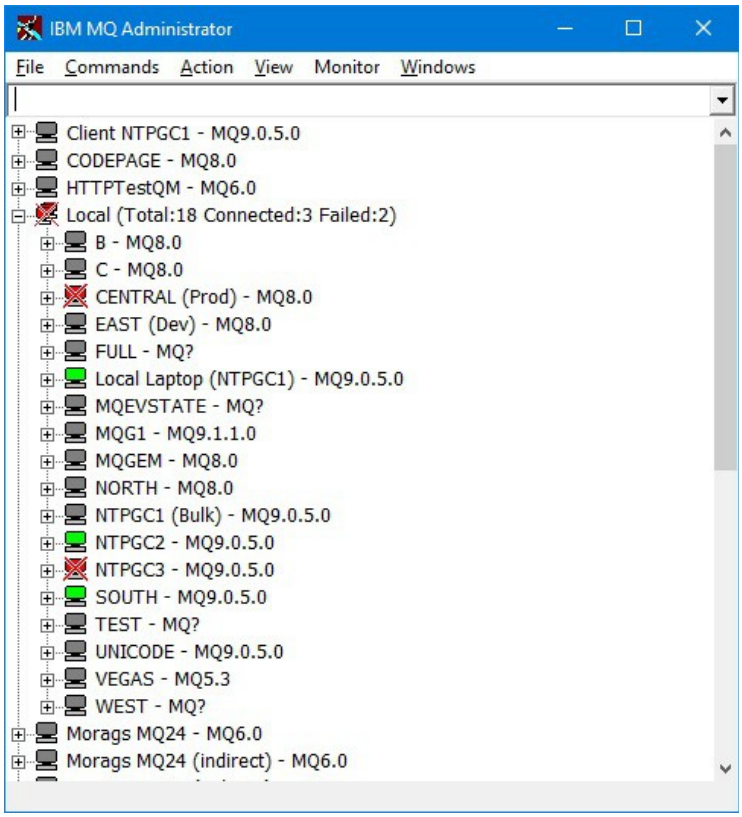
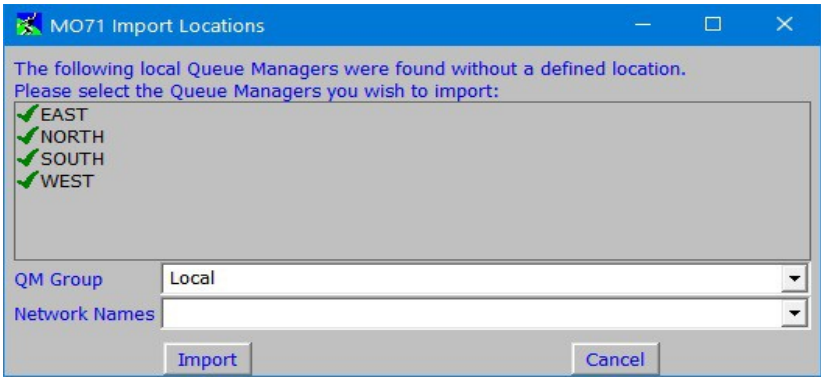


Figure 1: Main Window

3.1.1. Importing local Queue Managers

By default, each time MO71 starts it will check whether all of the local Queue Managers on the Windows machine are defined in MO71. If there are some that aren't defined a dialog will be shown which will give the user the opportunity to import definitions to each of the Queue Managers. For example, you could be shown a dialog like the one below:



In this example we can see that four Queue Managers have been detected which are not defined in the MO71 configuration. Note that importing a definition is doing nothing more than making a Location definition in MO71 to access that Queue Manager. No MQ objects are imported, the Queue Manager itself is neither accessed nor modified at this time.

If presented with this list the user can now decide which, if any, of these Queue Managers should be imported. Each entry is shown with a check (tick) mark or a cross. Only the entries with check (tick) mark will be imported. Clicking on the entry will change between the two. The Group on the main window can be expanded or collapsed either by clicking on the button the left of the window or by double clicking anywhere on the location Group line.

The other thing a user can specify (by typing in a name, or by selecting it from the drop down) is the name of the QM Group which should be used for the imported Queue Managers. The QM Group is merely a grouping concept on the main window of MO71. For example, you may wish to have all your local Queue Managers in a group called 'Local' as in this example. This can make finding particular Queue Managers much easier, particularly if you have a lot of them.

Lastly you can, if you wish, specify a list of Network names that these Queue Managers should be associated with. Network names are a useful way of grouping Queue Managers with a similar property – for example Production or Test or an Application group.

When the selection is correct the user should press the 'Import' button to have the location definitions automatically created. Conversely by pressing 'Cancel' none of the Queue Managers will be imported.

The imported location definitions will naturally contain default values for the various options you can specify for a location. If you wish to change these values then you can select 'Open location...' against each of the locations to make the modifications.

By default each time MO71 starts a check is made. If required this check can be suppressed in the 'General' tab of the Preferences Dialog. Just ensure that the 'Import local Queue Managers on startup' is unchecked.

You can re-drive this import check at any time from the menu File → Import Local Queue Managers …

You can also import location definition from....

- a Client Channel Definitions Table (CCDT) (see “From CCDT...“ on page 187)
- an MQ Explorer export file (see “From MQ Explorer File...“on page 188).
- another MO71 Configuration file (see “From MO71 Configuration...“ on page 188).

For further information on this topic, please see our blog for posts on the subject here.

3.2. Adding a Queue Manager Entry

To add an entry for a new Queue Manager in the Administrator use *'Add Location'* in the File menu. This will bring up a location dialog.

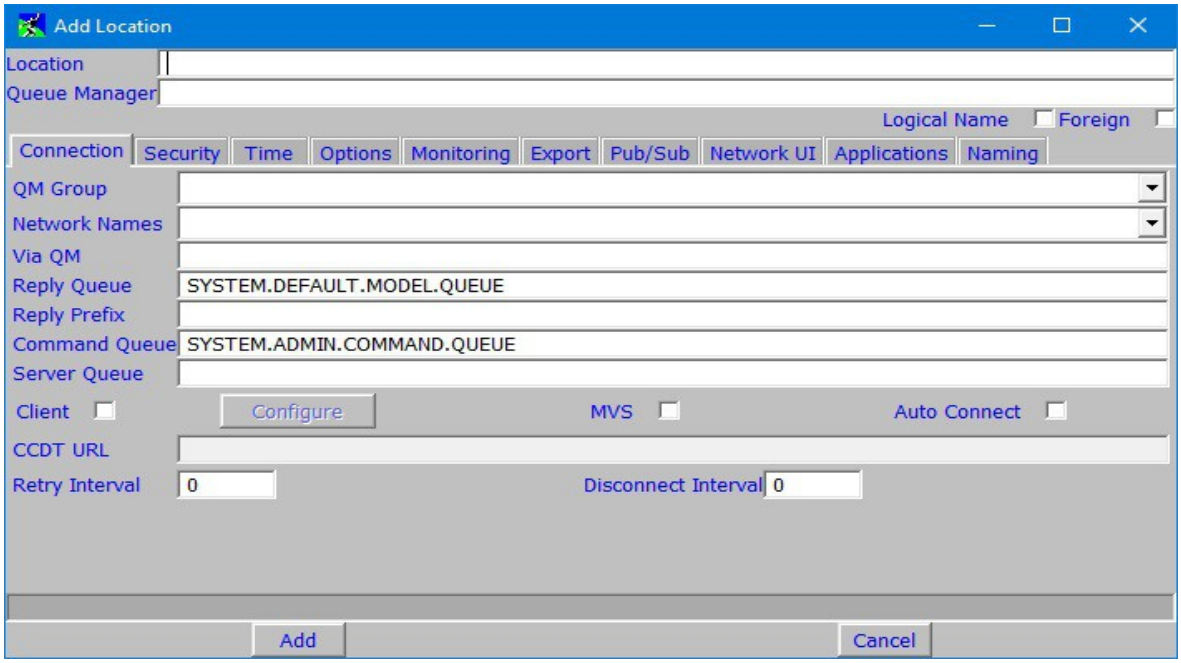


Figure 2: Add Location Dialog

You can administer a local Queue Manager by simply entering its name in the Queue Manager field and entering a text description of the location in the location field. To administer a remote Queue Manager you can either connect via a client or via another Queue Manager. By default the application receives replies from the Queue Manager via the standard model queue SYSTEM.DEFAULT.MODEL.QUEUE but it may be preferable to define an explicit queue for MO71 to use. See *"Reply Queue Details*" on page 229 for further details.

The following examples will show the additional configuration needed to use either of these methods.

3.2.1. Connecting via a Client

To directly connect to the Queue Manager via a client, select the Client check box and click on the Configure button to bring up a dialog to define the client connection channel. In this dialog fill in at least the Channel Name and the Connection Name. If you need to make a TLS protected client connection to your queue manager, see our blog post "MO71 TLS Configuration" for full configuration details.

If the Queue Manager can be hosted on multiple servers then you can specify the list of address in a comma separated list. What's more you can optionally follow each host address with a text string which will be displayed against that location whenever MO71 is connected to it. This means that you can easily tell which server is currently hosting the Queue Manager.

For example you could configure the following:

```
Connection Name    : host1(1414) "First", host2(1416) "Second", host3(1546) "Third"
```

The contents of the text string can be whatever you choose. For example, you may prefer to include brackets or only annotate the back-up server. For example:

```
Connection Name    : host1(1414), host2(1416) "Backup"
```

Note that you must configure the client channel directly for host list processing to work. You cannot, for example, use a channel defined in the Client Channel Definition Table (CCDT).

3.2.2. Connecting via a different Queue Manager

Connecting via a different Queue Manager can be useful in certain circumstances when a client connection to the required Queue Manager is either not feasible or possible. For example suppose we wish to remotely administer Queue Manager QM2 but wish all our messages to go to QM2 via QM1. To configure this set up you must perform two steps.

1. If not already defined we must first define a location for the 'via' Queue Manager QM1. We can either define it as a local Queue Manager (if it's on windows) or as a client.

2. Next we define the location we actually want to administer, QM2. In the 'Via QM' field of the location dialog we put the name of the Queue Manager where we are routing messages via. In this case it is QM1. We must also remove the Reply Queue since it is not needed for connection by this method.
   Note that in addition to defining these locations you must have IBM MQ communications (for example Sender/Receiver channels) in both directions between these Queue Managers. If the transmission queues are not called the same as the queue manager they are going to then it will also be necessary to define Queue Manager Alias definitions in accordance with IBM MQ remote configuration. Please refer to the *IBM MQ Intercommunication* manual for details in configuring remote messaging between two Queue Managers.

For details of all the fields in the location dialog see “Location Dialog” on page 214.

3.2.3. Creating two locations for the same Queue Manager

It should only be necessary to do this in very rare circumstance however it is possible. The reasons you might wish to do this are:

1. **In your MQ estate you have multiple Queue Managers with the same name**
   Of course every piece of MQ literature will strongly advise against having two different Queue Managers with the same name. Sometimes the reasons are understandable – such as we wish to have a test system that is identical to our production system. Sometimes the reasons are even humorous – the manual said we should issue CTRMQM QM1 – so we did! Either way, if you are in this unfortunate position rest assured, you are not alone.

2. **You wish to have multiple connections to the same physical Queue Manager**
   This can be useful if, for example, you wish to copy or move large numbers of messages via MO71 and still be able to Administer the Queue Manager.

Of course the simplest solution is just to run multiple instances of MO71. This is the recommended approach if you are trying to deal with the Test systems and Production systems having the same names problem. If this is the case you don't want anything connecting to both a production system **and** test systems; the possibility of cross-talk is just too high.

However, let's assume that you need to create multiple locations to the same Queue Manager – how do you do it? Well, the basic rule is that the combination of the Queue Manager name and Reply Queue must be unique. This means we have two options – we either change the Queue Manager name or we change the Reply Queue name.

- **Changing the reply queue**
  This is pretty straight-forward. The simplest approach is just to define a second model queue which can be used by these 'other' connections.

- **Changing the Queue Manager name**
  'How?' I hear you ask “can you possibly change the Queue Manager name and still connect to the right one?”". Well, you can use a feature of the MQ Client that any Queue Manager name starting with an asterisk(*) will be allowed to connect to a Queue Manager of any name. In other words if the Queue Manager starts with '*' it will allow the Queue Manager name to be different than the actual Queue Manager name. So, suppose you have a Queue Manager called 'QM1' you could define a location and specify the Queue Manager as '*QM1_SECONDARY'. Exactly what you call it does not matter, as long as it starts with an asterisk (*). Note though that this method only works if you are connecting via a client. You could still use it for a local Queue Manager it's just that you have to connect via the client library.

Chapter 4. Issuing Commands

MO71 provides a graphical interface to the most common commands one can issue to a Queue Manager. This interface has the advantage that the user does not need to remember the command syntax and generally reduces typing by keeping relevant fields on the dialog. To issue a command against one of the Queue Manager's via the dialogs the user must select the Queue Manager from the list and then select the item from the Commands menu that most closely approximates the command required[4]. To successfully issue commands then the Queue Manager and its respective command server (e.g. strmqcsv <QMNAME>) must be running.

The commands presented depend on the version of the Queue Manager at the remote location. For example clustering commands are only presented if the remote Queue Manager supports clustering.

The 'Commands' menu can be accessed either from the menu bar, or by pressing mouse button 2, after selecting the appropriate Queue Manager. Pop-up menus are available in most dialogs by pressing mouse button 2. The format of the Commands menu can be changed significantly depending on the users preference. For example, you can display the menus sorted by the most frequently used, or the most recently used. Or perhaps you prefer the commands grouped by category. To change the setting go to the "Preferences Dialog", see page 193 for more details.

The command menu contains commands such as :-

- a single object dialog for any Queue Manager object type
- a list dialog for any Queue Manager object type
- a single dialog and list dialog for Channel status
- queue statistics
- queue manipulation (See the section below on Queue Manipulation)
- cluster commands (where appropriate)

There are two styles of dialog for each of the main Queue Manager object types, for example, Queue and Queue List. The former style is designed to allow operations on a single object, on input of the object name. The list dialogs allow the display of, and operation on, multiple objects of that type.

4.1. The Command window

There are two types of command window, the list and single object display. Both window types allow the user to issue a command by pressing the command button if displayed at the bottom of the screen or by selecting the appropriate menu option from the pop-up menu that is displayed when you press mouse button 2. Note that only the common operations are presented as buttons, there are often more options available in the pop-up menu. The key point about command windows is that the user can start as many or few of them as required. So, it is perfectly reasonable to show a list of queues, a list of channels, the current set of channel status, etc all at the same time. What's more each window can remember their size and position so that when they are restarted they will come back to exactly the same position as the previous time they were shown.

The command windows are split into the following sections :-

4.1.1. Fields

These fields display the current value of the objects attribute. The user can move the entry field to any attribute (except certain read-only attributes) and change its value. This can be done by clicking anywhere on the field or its description with the mouse, using the up/down buttons or using the TAB keys. For fields that have a choice of values the drop down list must be displayed (PF4) and then the value selected using up/down keys.

- If the field is a name of another object, for example a transmission queue in a sender channel dialog then double clicking on the field will cause a dialog of that object to be displayed. This allows a quick method of displaying all the objects associated with a primary object.
- It is possible for an object dialog to contain more than one key field value. This is done by selecting more than one entry of a list dialog and then opening the definition or by ending the key field name with a '+' sign to indicate that you want to type further key values. The objects can be qualified with the name of the Queue Manager they belong to if necessary. This is done using the format: <Queue Manager Name> ':' <Object Name>.

[4]An alternative way of entering commands to a QMgr is to use the MQSC window. This is available from the Action menu and the Qmgr context menu and is described later. The MQSC window has the advantage that **any** command supported by the Queue Manager may be issued but it has the disadvantage that the output of the command is not formatted or post-processed and the user must know the exact syntax of the command.

When a dialog does have more than one key field specified then any command issued will be applied to all the objects in the list. For example, it is possible to change the maximum depth of queue 'Q1' on multiple Queue Managers by a single click. Note that some actions are only appropriate if all the objects are the same type. For example, trying to update Alias, Remote and Local queue definitions will not work since the object type is intrinsic to the command being issued. In these cases ensure that only the same type of object is in the dialog at any one time.

Note that wherever possible the commands available will reflect the objects selected or displayed but it may be that the command selected is not applicable to all the objects. In this case an error message will be displayed. Because a single command could now generate many responses, one for each object, the response window at the bottom of the dialog is a drop down list which allows all the responses to be checked.

For example to start (or stop) many channels at once. First display a list of channels. Select the required channels using standard controls e.g. <CTRL> + Mouse Click for individual selections, <SHIFT> + Mouse Click for selecting a range. Click Mouse Button 2 then select the command required. The command will be issued against all the selected objects in sequence and the command response is given in the drop down list at the bottom of the dialog.

- For list windows the fields are used to limit the amount of data requested from the command server. For example, on the Queue List specifying a Queue Name of "SYS*" will only request queues starting with the string "SYS".
  If space is limited the field area can be reduced by dragging the bar at the base of the fields or they can be hidden completely by selecting the *'Hide Fields'* pop-up menu option or clicking the 'Show Fields' tool in the toolbar.

4.1.2. Status window

The status window is used to relay information about the current status of the dialog, including any error messages. This is a drop down list so a history of responses can be seen. Note that the history will be abbreviated. Not all messages are written to the history and the same entry will not be entered twice in a row.

4.1.3. Action buttons

Some of the commonly used options are displayed as action buttons to the user. These are equivalent to the options on the pop-up menu. If space is limited the buttons can be hidden by selecting the *'Hide Buttons'* pop-up menu option or the 'Hide Buttons' tool in the toolbar.

4.1.4. Command Keys

Most of the commands can be issued using a keystroke such as *Alt*-r. If a command has a key shortcut the letter will be underlined in the context menu.

4.1.5. List Windows

- **Object List**
  A list box containing the returned objects. Double clicking on an entry will cause a dialog for that object to be shown. By default the field which is double clicked on with be the focus field of the new object dialog. This behaviour can be switched off in the Lists tab of the Preferences Dialog on page 198.

- **Queue Manager List**
  By default a list dialog is associated with just a single Queue Manager. However, most of the lists can be associated with more than one Queue Manager. This allows, for example, a single dialog to show the queues for multiple queue managers. Or perhaps to monitor the depth of the dead letter queue on a group of Queue Managers. There are two ways to associate multiple Queue Managers with a list.

  1. **Select 'Show Queue Manager List' or click the 'Show Queue Manager List' tool from the toolbar**
     A pane showing the available Queue Managers will appear on the left side of the list dialog. Each Queue Manager can be (de)selected by clicking on the selection button. Immediately to the right of this button is a database symbol which indicates whether data is available for this queue manager or whether the data is refreshing. By clicking on this button a request is sent to just this single queue manager for a refresh of the data.
  2. **Using the qm(), qmloc() and qmnet() filter functions**
     For example, a filter of *qm("NT*")* would associate all queue managers starting with the characters **NT** with the dialog. Care should be used with these functions. A filter of *qm("*")* will associate all Queue Managers with the dialog. This can cause a large number of requests for data to be sent when the dialog is refreshed and the dialog can also use a significant amount of storage if there are a large number of Queue Managers and Queues.

- **Split List**
  If required the list window can be split into two separate panes by selecting the 'split list' option from the context menu or clicking the 'Split List' tool in the toolbar. For example, the object name such as Channel Name can be kept in view in the left hand pane while the channel attributes are scrolled in the right hand pane.

- **Filter button and filter entry field**
  This entry field allows the user to perform filtering on the data returned by the command server before displaying. A filter can be added by typing a filter expression in the filter window and pressing the '*' filter button. The filters are stored in the list portion of the filter field, this allows previous filter values to be selected from the drop down list. Please see "Filtering" on page 28
  If space is limited the filter can be hidden by selecting the *'Hide Filter''* pop-up menu option or pressing the 'Show Filter' tool in the toolbar.

- **List Titles**
  When a list is refreshed the titles of the various fields in the list are displayed in the list titles window. The list of attributes displayed can be chosen from the '*Alter List'* pop-up menu option. If a particular attribute is not returned for any object in the returned list then that column is not displayed, thereby allowing more space to be used for other fields.

  The edges of each column can be dragged via the mouse to re-size the columns. The user specified widths can be cancelled by selecting the *'Reset Column Widths'* pop-up menu option or by pressing the reset width toolbar icon. If a list dialog column is too narrow to display the entire column data then the data will end with an ellipsis (…) indicating that there is more data to display. If the user hovers the mouse over the '…' characters then a side window will be displayed showing the full text.

  The text displayed in a list title will either be a long description, a shorter more compact description, or the MQSC keyword. You may choose one of these explicitly or allow MO71 to automatically choose between the long and short descriptions depending on the width needed for the contents of each column. There are two Auto options, one which favours the longer description and one that favours the shorter description. In the pop-up menu, choose *'List Titles'* and then one of *Auto – Short*; *Auto – Full*; *Short*; *Full*; *MQSC*. When using either of the Auto options, if you manually resize a column, or the entire dialog window, the longest description that fits into the column width will be used.

- **Sorting the list**
  Each list set will have a default sort fields. This can be changed using the '*Alter List'* pop-up menu option. A list can actually have two sort fields, a major and minor sort field. These can both be specified in the '*Alter List'* pop-up menu if required.

  To make a temporary change to the sort field for the list you can click on the list titles. This will sort the list by the column clicked. The sort will alternate between ascending and descending order. An arrow pointing either upwards or downwards in a column title shows which field is being used to sort by and whether it's ascending or descending respectively.

  Similarly the minor sort field can be set by clicking on a field title while holding the 'Alt' key. A slightly smaller arrow will show in the list title showing that it is the minor sort field. Again, the direction of the sort arrow shows whether the sort is ascending or descending. Bear in mind that, depending on what the major sort field is, the order of the minor sort field may have no effect whatsoever. For example, if a list's major sort is the object name then selecting any other field for the minor sort will have no effect unless there are multiple entries with the same object name.

  You can remove the minor sort field the same way as it was set by holding the 'Alt' key as you click on the minor sort field title.

- **Pie Charts**
  You can display a pie-chart of any field just by right clicking on the field column and selecting the 'Pie Chart Set Field....' menu. There are two types of Pie Charts depending on the type of field. For enumeration data types, such as Queue Types, where there are a clear, predefined set of values, then the pie chart will show you how many of each type you have. The other type of Pie Chart is concerned with fields like Queue Depth or Bytes Sent etc, ie values which are counting things. For these types of fields the pie-chart will reflect the actual value of the field compared to the total for all the fields.

  You can use the piechart() function to pie-chart a field from the filter line. This also has the effect of adding the field to list of displayed fields.

4.2. Dialog Examples

As has been mentioned before MO71 provides the user with many different dialogs which give the user different views into the MQ definitions. For the majority of MQ objects there is a dialog which shows the list of the MQ objects and another one which shows just a single definition of that MQ object. You can start as many dialogs as you wish and even have multiple versions of the same dialog. So, for example, you could have a list of queues showing the transmission queues, another queue list showing the queues with messages on them and yet another queue list showing just your cluster queues. MO71 is totally flexible and you can choose how you want to display or interact with the MQ definitions. As you might expect the context menu (right button click) shows various commands and options you have with each dialog.

4.2.1. Displaying a list of Queues

- Select the appropriate Queue Manager
- Choose *'Queue List...'* from Commands menu
- Press *Refresh* to cause the list to be shown

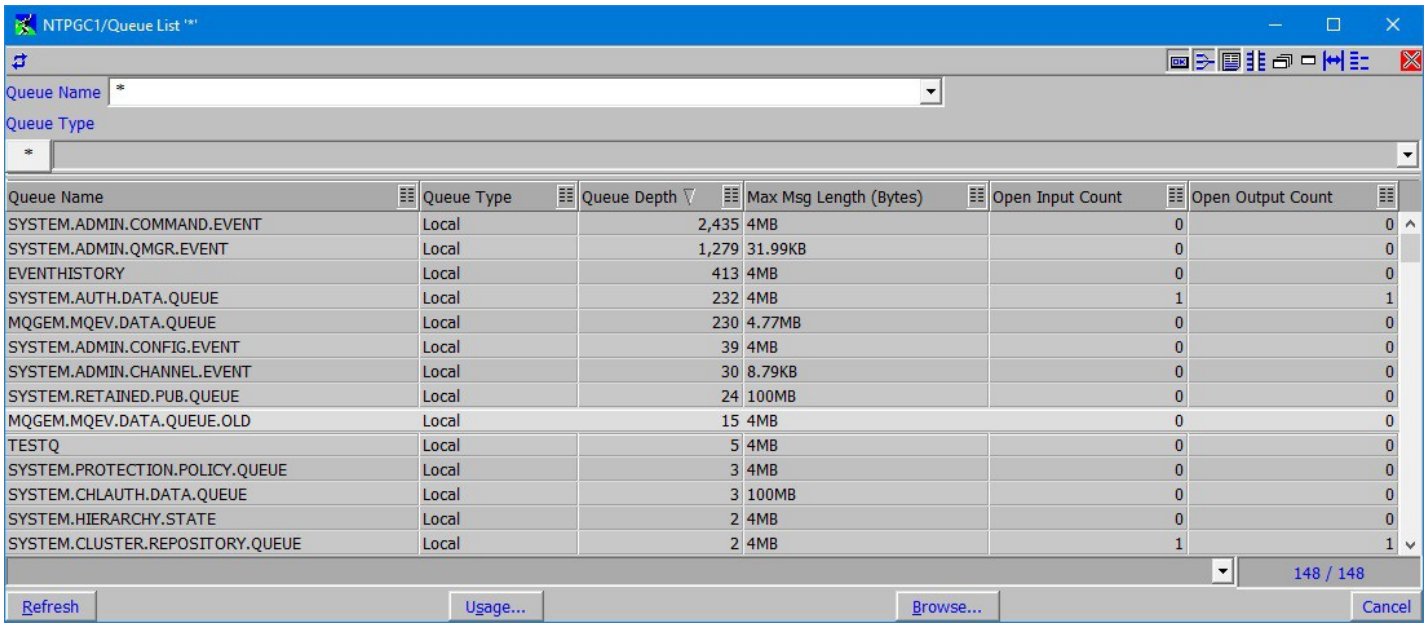


Figure 3: Queue List Dialog

A list dialog shows the list of objects of that particular type. We'll see later that you can change the view in many different ways. For example you can change which objects are shown using the filter field. You can change which objects fields are shown using the 'Alter list dialog'. The colours of the lines or even of individual cells can be changed using the filter.

The title of the dialog will show you the type of dialog and also the main key, if any, of the dialog. In this case just an '*' which indicates all queues. However, if you were to change the value of 'Queue Name' you would see the title change as you type.

4.2.2. Displaying Queues for multiple Queue Managers

A very powerful feature of **MO71** is the ability to show objects from different Queue Managers in the same dialog. You could, for example, monitor just the Dead Letter Queues on all your Queue Managers or you could look at all of your transmission queues. You could look at the status of all your channels across all your Queue Managers. There are clearly many potential uses.

In this example we are showing the list of Queues on two different Queue Managers.

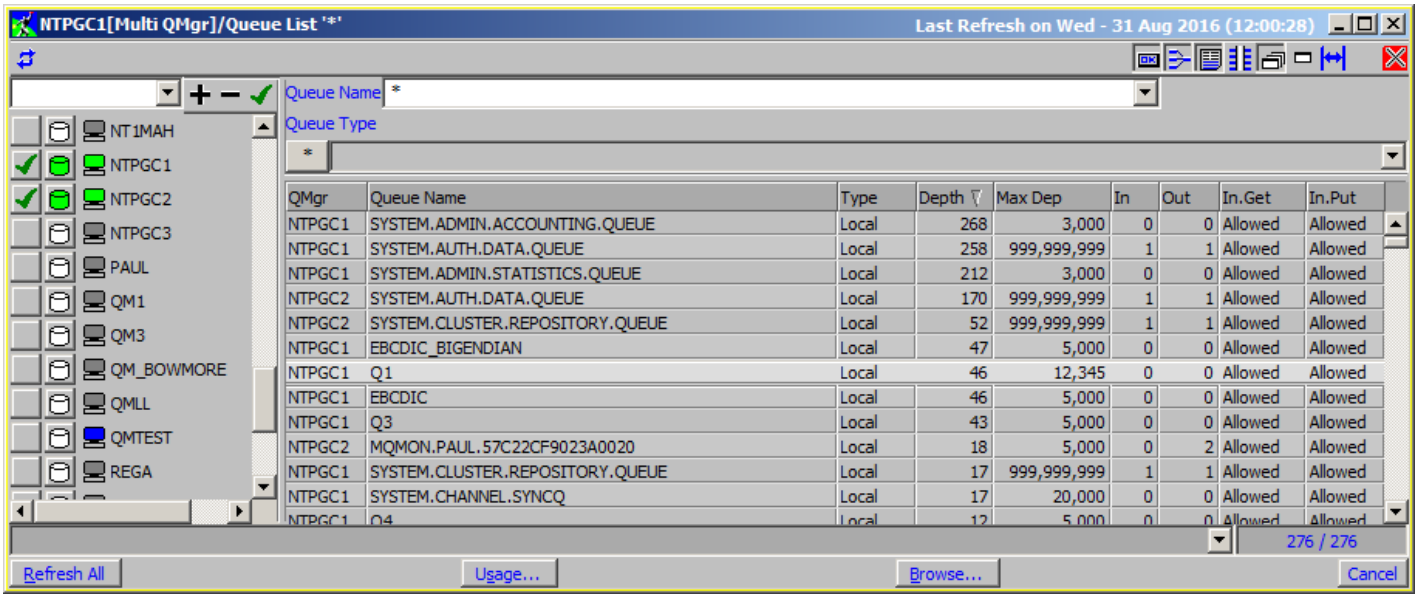


Figure 4: Multi Queue Manager List

To show multiple Queue Manager objects do the following:

- **Display the multi Queue Manager pane**
  This is done by clicking the 'Show Queue Manager list' tool in the toolbar or selecting the menu from the right-click pop-up menu Option. By default the 'current' queue manager will already be selected in the list.

- **Select the additional Queue Managers(s) you want to display**
  The search field at the top of the Queue Manager list allows you to filter the Queue Managers to show only the Queue Managers you are interested in. Clearly this is most useful if you have a large number of configured Queue Managers.

  By default the search field will be initialised to the value of the search field in the main window. This means that if you are only interested in a sub-set of Queue Managers you can change the main window and then start whatever lists you are interested in and, by default, their Queue Manager lists will also just show the same subset. This behaviour can be switched off in the Lists tab on the Preferences Dialog.

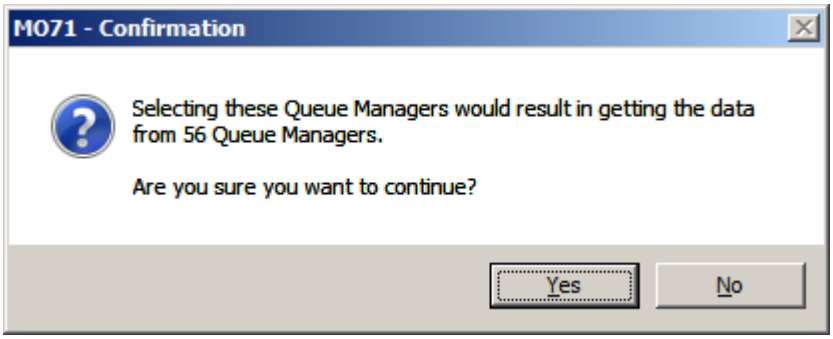
  As you type a value into the search field the list of Queue Managers is automatically filtered after a short delay. The field is a drop-down list which contains the last few search strings used, either in the main window or in any list dialog. The most recently used searches are shown at the top of the list. The syntax is the same as used in the main window which is described on page 191

To the right of the search field there are also three tools which can be used to make selecting groups of Queue Mangers easier. In order these are:

- **Add all shown**
  Selecting this tool will cause all of those Queue Managers displayed to be added to the currently selected list of Queue Managers.
- **Remove all shown**
  Selecting this tool will remove all of those Queue Managers displayed from the currently selected list of Queue Managers.
- **Select only shown**

  Selecting this tool will replace the current selection of Queue Managers with the ones shown.

Selecting a Queue Manager will show the currently held data if possible. If no data is currently held for a selected Queue Manager then **MO71** will connect to the Queue Manager and retrieve the information. If selecting one of the tools would cause **MO71** to retrieve data from a large number of Queue Managers, then a confirmation dialog such as the one below may be shown to verify the action, as it may cause significant message traffic to retrieve the data.



The number of Queue Managers at which point MO71 will ask for confirmation can be set in the Lists tab in the Preferences Dialog.

- **If necessary click the database icon to refresh the information**

The user should be sensible about how many Queue Managers are selected in the left-hand pane. If a very large number of Queue Managers are selected and each Queue Manager has a large number of queues then the performance of the dialog could be affected. In addition care should be taken with the 'REFESH' button since this will send a REFESH command **to each** of the Queue Managers potentially causing a lot of unnecessary message traffic. If you want to refresh just one of the Queue Managers then you should use the database icon next to each Queue Manager.

4.2.3. Defining a Queue

- Select the appropriate Queue Manager
- Choose *'Queue...'* from Commands menu
- Fill in the queue attributes required, for example *Queue Name*
  Any blank fields will inherit the default values
- Press *Create* to cause the queue to be defined
- Pressing *Refresh* will show the complete queue definition

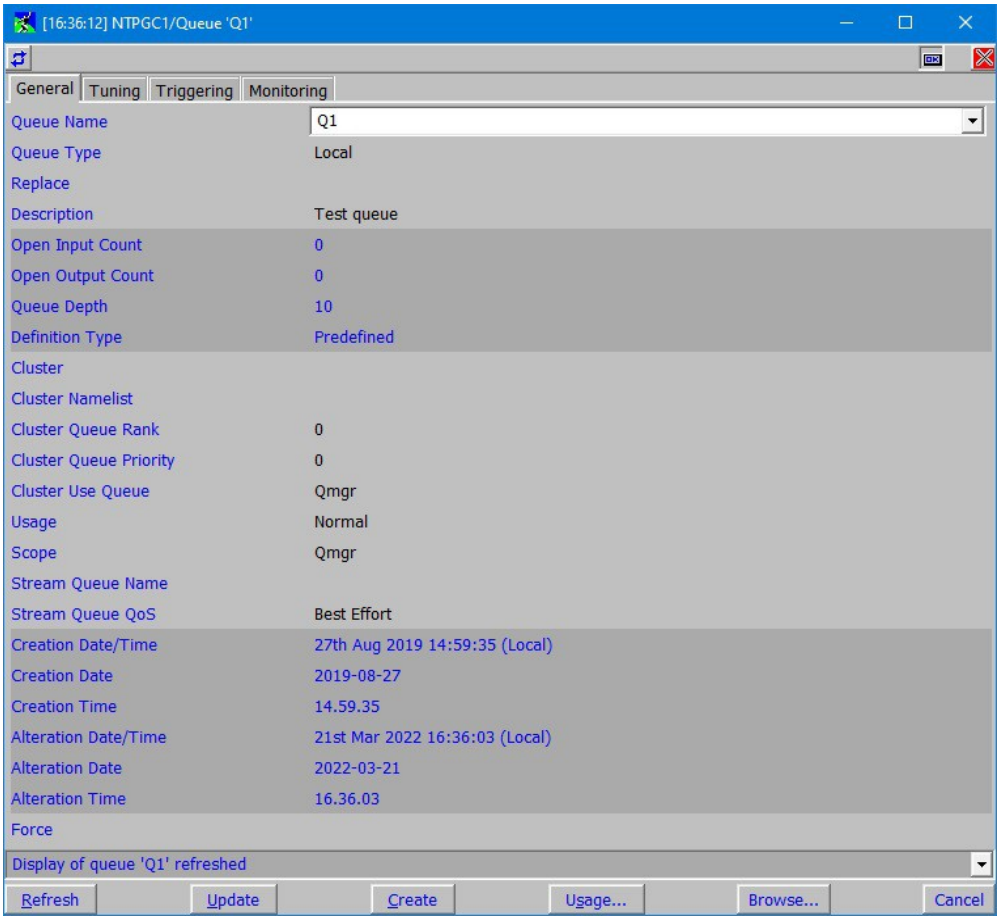


Figure 5: Queue Dialog

In the Queue dialog screen-shot you can see that the dialog has a number of Tabs at the top of the screen. The Tabs are on several of the object dialogs, the ones with large numbers of fields, to make it easier to find the required field. If you prefer you can change the look of the Tabs or even revert to just a list of fields by changing a preference option in the "Preferences Dialog", see page 193 for more details.

4.2.4. Defining a Queue like another Queue

- Select the appropriate Queue Manager
- Choose *'Queue...'* from Commands menu
- Enter queue name to copy
- Press *Refresh* to show complete queue definition
- Change any queue attributes required, which must include the *Queue Name*
- Press *Create* to cause the queue to be defined

4.2.5. Starting a Channel

The following examples illustrate the two different ways you could start a channel.

First Method

- Select the appropriate Queue Manager
- Choose *'Channel List...'* from Commands menu
- Press *Refresh* to cause the list to be shown

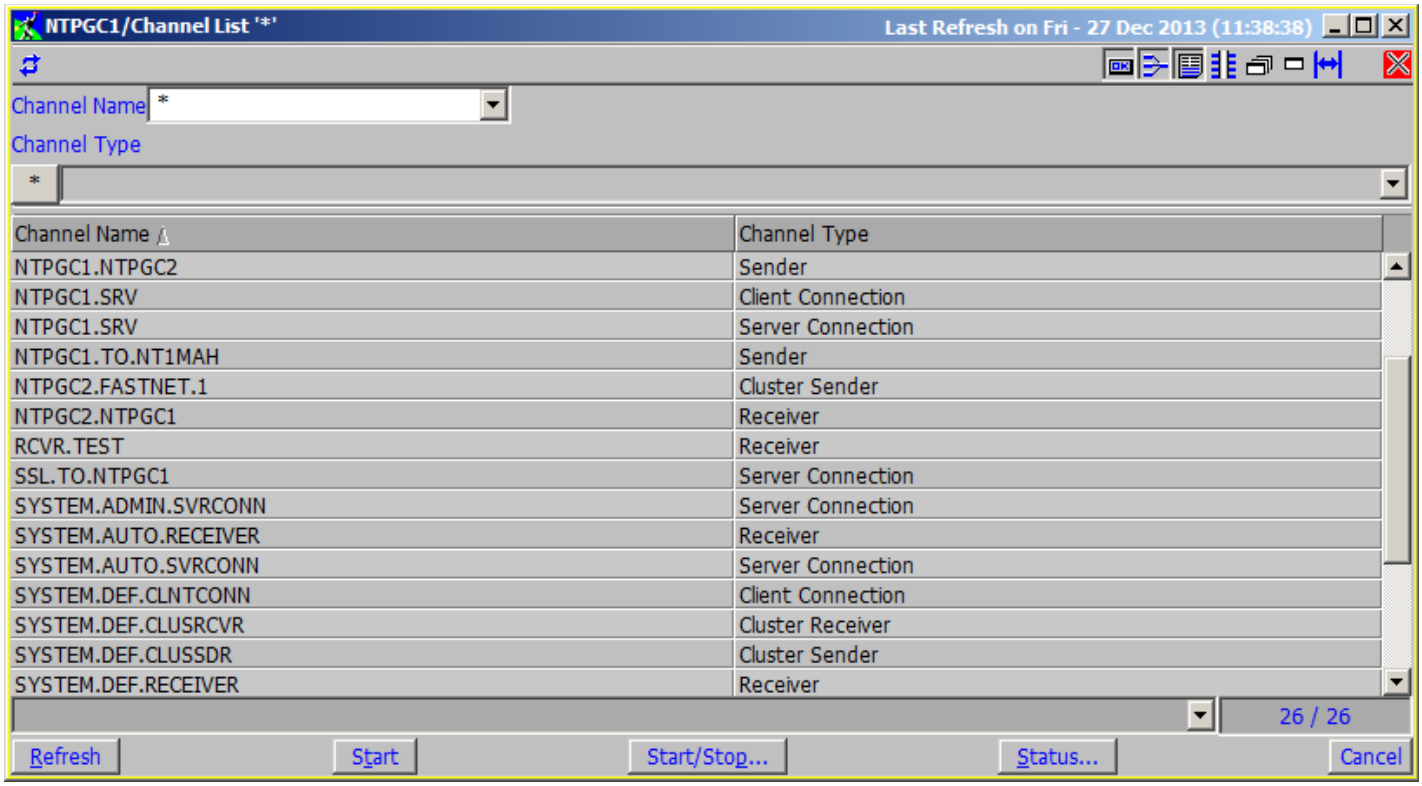


Figure 6: Channel List Dialog

- Find the Channel in question and select it
- Select *'Start/Stop...'* from the pop-up menu or button
- Press *'Start'* in the Start/Stop Channel dialog

Second Method

- Select the appropriate Queue Manager
- Choose *'Channel...'* from Commands menu
- Type the name of the Channel to start
- Press *Refresh* to cause the channel definition to be shown
- Select *Start/Stop...* from the pop-up menu or button
- Press *'Start'* in the Start/Stop Channel dialog

Clearly if there are a number of operations to be performed against objects of the same type then the list option involves less typing for the user.

4.3. The Pop-up menu

All the commands available are contained in a pop-up menu that is displayed when mouse button 2 is pressed. Care should be taken with destructive commands such as *'Delete'* and *'Clear Queue'* if confirmation dialogs are not active (see "Preferences Dialog" on page 193 to control confirmation dialogs). This pop-up menu may also contain a *'Print...'* option which is described in "Printing" on page 107, and an *'Export...'* option which is described in "Exporting Definitions" on page 101.

The pop-up menu also contains an *'Options'* sub menu. By selecting this the user is shown a number of options which change the look and behaviour of the dialog. Note that the list displayed will depend on the dialog. Pop-up menus for the message dialogs have other sub-menus in addition to this one. For details see "Queue Manipulation" on page 60.

- **Alter list**
  Each list has a predetermined set of attributes that are displayed for each object. For example the Queue list displays the Queue name, Queue type and the current depth. If required, the user can change this to a list of his/her choice in two ways. Firstly the lists can be changed globally for all locations by using the option under the *'View/Set Default Lists'* menu on the main window. Secondly the list of attributes can be changed only for this location by selecting the *'Alter list'* option from the pop-up menu. See "Alter list dialog" on page 23 for how to operate the dialog.
- **Split List**
  By selecting this option the object list can be split into two separate panes, each capable of displaying the entire data. This is useful when displaying a large number of attributes and the user wishes to keep one set of attributes on the screen, such as the object name, while the other attributes are scrolled from left to right.
- **Reset Column Widths**
  Selecting this options will cancel any user defined column width and the system will choose what it considers to be the best width for the column, The user can set the required column width by dragging the edge of the column title.
- **Make Filter Default**
  Objects lists can be filtered via a Boolean expression (see "Filtering" on page 28) to restrict the number of objects displayed in the list. Selecting this menu item will associate the current filter with this list type. Each time the dialog is initially displayed it will have this filter. Selecting this menu with the filter window empty will disassociate any previous default filter.
- **Initial Refresh**
  Selecting this menu item will cause the list to be automatically retrieved from the command server whenever a dialog of this type is opened. This should be used if the list filtering options are rarely used.
- **Hide System Objects**
  When selected all objects starting with "SYSTEM." are removed from the displayed list. The object count at the bottom of the window will reflect the fact that not all objects are being displayed.
- **Update Predefined Definition**
  By selecting this option a predefined dialog definition is either created or updated depending on whether the dialog was started from a predefined dialog definition. Note that if this option is used to create a predefined definition the 'description' field will be blank. In this case the *Predefined Dialog List* dialog should be used to select the newly created definition and add a description.
- **Auto Refresh**
  This displays a dialog which allows you to set an auto-refresh time interval and optionally ask for the returned information to be exported to a file. For more information please see "Auto Refresh Dialog" on page 21.
- **Highlight difference from default**
  This menu option is only available on object types which have the notion of a default object. For example, queues have a default definition for each type of queue. When this option is selected attribute values which are the same as the value in this objects corresponding default object will be displayed in a subdued colour[5]. In other words, the differences will be highlighted.

  This menu will be greyed out if the default object definitions have not yet been retrieved.
- **Display difference from default**
  As above, this menu is only available on object types which have the notion of a default object. When this option is selected then only those columns which contain a difference from the default object definitions will be displayed. Note that which columns are displayed is therefore heavily dependant on which objects are displayed.

  If this option is chosen the window title will show "(Default Differences)" to remind you that you have this option selected and thus may not be seeing all the columns you have requested, such as "Queue Depth" on a queue list.

[5] The colour with which 'lowlight' items are displayed in a list can be set in the colour dialog by adjusting the value for 'List lowlight Foreground'.

- **Highlight difference fields**
  This menu is only available on 'compare' type dialogs. When selected the fields (attributes) which are the same on both Queue Managers are displayed in the list lowlight.
- **Display only different fields**
  This menu is only available on 'compare' type dialogs. When selected only field columns which differ between the two Queue Managers are displayed. The columns displayed are therefore heavily dependant on which objects are displayed.
- **Display objects**
  This menu is only available on 'compare' type dialogs. It allows a quick way of selecting a subset of the objects based on the categories :-

Menu Option	Meaning
Qmgr A defined only	Display only objects defined on Queue Manager A
Qmgr B defined only	Display only objects defined on Queue Manager B
Matching entries	Display only objects which are defined the same on both Queue Manager A and Queue Manager B
Non-matching entries	Display only objects which defined differently Queue Manager A and Queue Manager B

- **Copy to clipboard**
  This option provides a quick way to copy some of the list data to the clipboard. The data which is copied depends on the state of the control keys at the time the option is selected.

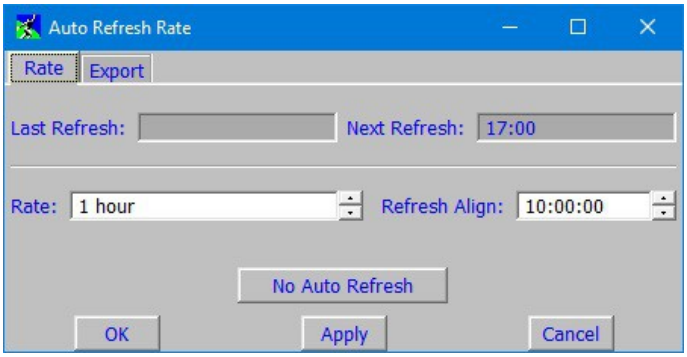
SHIFT key pressed ?	CTRL key pressed ?	Action
No	No	Copy the object name of selected entries
Yes	No	Copy the object name of all entries
No	Yes	Copy all fields of selected entries
Yes	Yes	Copy all fields of all entries

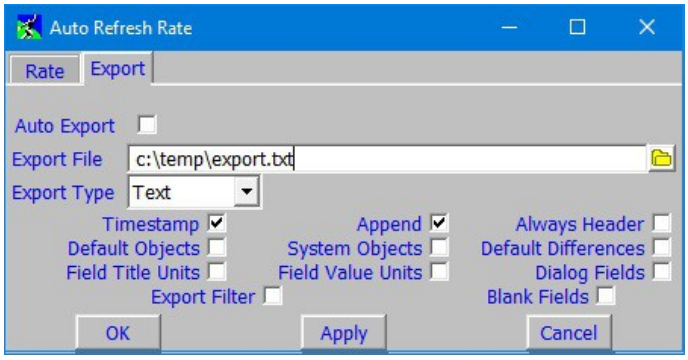
- **Hide/Show Buttons**
  This option allows the user to Hide/Show the buttons of the dialog to make best use of the available screen area.
- **Hide/Show Toolbar**
  This option allows the user to Hide/Show the toolbar. The toolbar allows quick selection of common operations.
- **Hide/Show Fields**
  This option allows the user to Hide/Show the fields of the dialog to make best use of the available screen area.
- **Hide/Show Filter**
  This option allows the user to Hide/Show the filter windows of the dialog to make best use of the available screen area.
- **Hide/Show Column Filters**
  This option allows the user to Hide/Show the column filter windows displayed at the top of each column.
- **Hide/Show Queue Manager List**
  This option allows the user to Hide/Show a pane of Queue Managers to the left of many of the list dialogs. This pane allows the dialog to be associated with more than just a single Queue Manager and therefore display command results from multiple locations.
- **Reset Queue Manager List**
  Selecting this option will reset the dialog back to being associate with the single Queue Manager that was selected when the dialog was started.

4.4. Auto Refresh Dialog

This dialog allows you to set an auto-refresh time for a command dialog. This allows an administrator to constantly monitor, for example, the depth of the Queues or the status of Channels.

It is displayed when you select *'Options-Auto Refresh'* from any object or list dialog. You can set a different refresh time interval per command dialog type. Auto Refresh is also available on the MQSC window. The window title will display the refresh rate if auto refresh is active. Different refresh intervals can be specified for each type of dialog.





The dialog allows you to specify a refresh rate and, if required, a refresh align time. The rate is how often you would like the dialog to refresh. In this example we are saying every hour.

The refresh align time field allows you to specify an alignment time. By default it has the value of 'None' indicating that there is no alignment. However, you can specify a time as in the example above. The align time is one of the times you would like a refresh to occur. So, in this example we have specified a time which is 'on the hour' so this dialog will refresh every hour on the hour. We could have specified, for example, that we wanted to refresh daily at 9am. Aligning the refresh intervals to specific times of the day is most useful when combined with auto refresh exporting (which is explained in the next section). By aligning the export times you can more easily compare results of successive days and look for differences or trends.

You may only use refresh alignment if the refresh rate is exactly divisible into 24 hours. If you specify a rate which is not divisible then the refresh align field will be disabled and the value of 'Unavailable' will be shown.

It is possible to set a very small refresh interval, for example 1 second, which is fine for particular tests on a development machine but probably not a good idea on a production system. Remember that refreshing a dialog involves message exchanges with the command server and is therefore not an insignificant cost particularly when there are a lot of objects involved in the request.

The 'No Auto Refresh' button is a quick and simple way of switching off auto refresh rather than having to type in a zero rate.

The Auto Refresh dialog also gives you the ability to have an export file generated each time the data is refreshed. This is a rather specialised use of export but can be useful when you wish to capture a series of object status history. The resultant file can then be post-processed by another program to look at the trends. As usual with export you can choose from several file formats, for post processing the formats of CSV, XML or JSON are likely to be the most useful.

4.4.1. Auto Refresh Export Fields

The fields related to auto refresh export and their meanings are as follows

- **Auto Export**
  Controls whether the dialog will write to the auto export file or not.
- **Export File**
  The export file format can contain character inserts which are described in chapter 24.8 File Name Inserts on page 228. The button immediately to the right of this field will show a file selection dialog.
- **Export Type**
  Determines what type of export file will be generated.

- **Timestamp**
  Controls whether date and time columns are added to the exported data.
  If this option is used with the MQSC dialog then the exports times are the times of the export not of the time that the data was received from the Queue Manager.
- **Append**
  When selected any data will be appended to the end of the file if it exists. If not selected then any existing file will be over-written.
- **Always Header**
  Controls whether the column headings are always exported or not. If not set then the header is only written to the first line in the file.
- **Default Objects**
  Controls whether the default object definitions are also exported.
- **System Objects**
  Controls whether the system objects are also exported.
- **Default Differences**
  Selecting this option will cause only the attributes which are different from the default object to be written. Note that this option only applies to objects which have the notion of a default object, other objects will have their complete definition written even if this option is selected.
- **Field Title Units**
  Whether the field titles should contain the Units names such as Seconds, Bytes, Milliseconds etc
- **Field Value Units**
  Whether any Unit fields should be expressed in terms of their Units.
- **Dialog Fields (Lists only)**
  Whether the dialog fields, as well as the list data, should be exported.
- **Export Filter (Lists only)**
  Whether the current filter string should be exported along with the list data.
- **Blank Fields**
  This options means slightly different things depending on when it is being used.
  - **Export of an object (such as a single object or an 'All Fields' export from a list)**
    This option requests that fields that would normally not be shown, because their value is empty, be also included in the export.
  - **Export from a list (only CSV)**
    It specifies that columns from the list that are completely blank and would normally be excluded both from the display and from the export to the file, are to be included in the CSV file export. This makes it much easier to append or merge multiple export files since the number of output columns remains constant.
- **One Line (MQSC only)**
  For an MQSC export this option controls whether the MQSC should be output in a single line. A single line allows multiple objects to be compared easily since each attribute is given it's own column.

4.5. Alter list dialog

This dialog allows the user to change the list of attributes displayed in a list window. It is invoked either by selecting the *'View/Set Default Lists'* menu on the main window or by selecting *'Alter list'* from the pop-up menu on a list window. The first of these sets the fields 'globally' for all locations. The second invocation is only 'Local' to this location and will not affect the lists displayed by other locations.

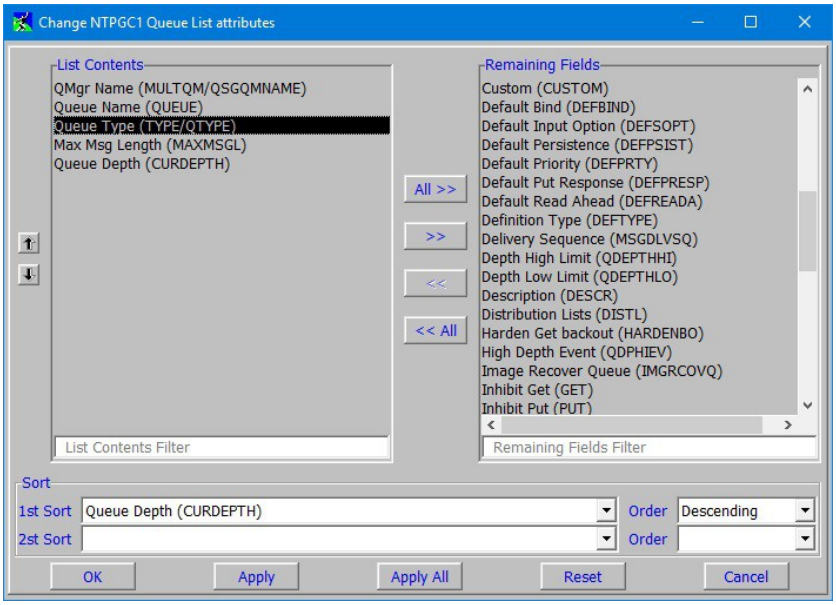


Figure 7: Alter List Dialog

The dialog displayed shows two sets of the attributes, the **List Contents** in the left hand column and the **Remaining fields** in the right hand column. Attributes are moved either by selecting them and pressing the arrow buttons or double clicking on items. To move all the attributes use the *'All >>'* and *'<< All'* buttons. The attributes in these lists have a text description followed by an identifier in brackets. This identifier is the name, usually the MQSC name, which should be used in filter expressions (see "Filtering" on page 28 for more information). A value entered into either filter field will limit the fields in the list to those containing that string. This allows you to more easily find the fields you are looking for on object with lots of fields.

When a field is added to the List Contents it will be placed before the first selected item if any. If none of the fields are selected then it will be put at the end of the list. At any time fields in the List Contents can be reordered by selecting a single entry and pressing the Up or Down arrows appropriately. The default sort fields can also be specified using this dialog. Select the fields required for the major and minor sort fields and the direction of the sort. A list must have a major sort field but it does not have to have a specified minor sort field.

Pressing the *'OK'* or *'Apply'* buttons will cause the new attributes to be adopted by the dialog list. By default the first attribute will be used as the sort field for the list. However, if an item is selected in the List Contents column when the *'OK'* or *'Apply'* button is pressed then that field will be used as the sort field. This simple technique is overridden if the user uses SORT functions in the filter window or clicks on the list titles. The *'Apply All'* button will update all instances of this type of list. For a global change this means that all locations will have their current list replaced by the new list in the dialog. For a local change the change will be 'saved' for this location. Subsequent invocations of MO71 will display this new list for this location. This method replaces the *'Make List Default'* option available in previous releases.

As the number of columns is increased you may find that the complete field title is not displayed in order to safe space on the dialog. By hovering the mouse over the column title a tooltip will be displayed which will give the complete name of the field and the MQSC attribute name which can be used in filter expressions.

4.5.1. MULTIQM or QSGQMNAME field

This field is special in that it only appears if it is appropriate to do so. The dialog allows the user to position the field within the list, or remove it completely, but to save space in the list dialog it won't always be displayed. For example, if the dialog is associated with more than one Queue Manager then this column will be visible to allow the user to see which Queue Manager an object belongs to. If the dialog is only associated with the dialogs main Queue Manager then the column is unnecessary and is not displayed. It is recommended that this field always be included and be the first element in the list.

4.6. Hostname Support

In a channel status list and object dialog MO71 will faithfully display the values returned by the Queue Manager. To identify the machine that is at the other end of the channel MQ returns the 'connection name' field, which almost invariably contains the IP address. The problem with IP addresses though, of course, is that they are not very user friendly. To alleviate this problem MO71 can also display the host name. This is a more memorable name which helps in identifying the machine in question. The host name field is always displayed in a channel status object dialog but must be selected using the “Alter list dialog“ for the list dialog.

The value shown in this field can either be a system or user value. By default it will be a system value. A system value is one that is returned by the standard TCP/IP function gethostbyaddr(). For example *paul.mqgem.com*.

4.6.1. DNS System Values

It is well known that calling gethostbyaddr() can take a considerable amount of time. Any API which invokes the DNS server can be costly. Because of this DNS System Values are only queried if necessary and, once queried, they are then cached. If you wish they can even be cached across restarts of MO71.

By default host names are not displayed in the channel status list dialog. This is to lessen the potential impact of calling gethostbyaddr() for hundreds of different IP addresses. If you choose to display IP addresses then you should be aware of the following:

1. **IP addresses are queried by a separate thread**
   Because of the potential for a request to the DNS server taking a long time MO71 delegates this call to a separate thread. This means that the host name value displayed in the dialog is completed asynchronously. During the period before the real value is returned MO71 will display '…'. The separate thread will issue the DNS requests serially. In other words, only one outstanding gethostbyaddr() call will be issued at any one time.
2. **Host name caching**
   When a value is returned from the server, even if it is 'No host found', this value is cached by MO71 for a certain length of time. By default this time is 1 week but this can be changed in the Connection tab of the Preferences Dialog. By default MO71 will forget about all the system host names when the program is ended. However, by checking the 'Save resolved hostnames' setting in the Connection tab of the Preferences Dialog the values will be saved to a file called **mqmon.dnc.** This file is not intended to be edited by users. You can, if you wish, delete the file which means that MO71 will restart with an empty system host cache.
3. **DNS Suffix removal**
   In the Connection tab of the Preferences Dialog you can specify a comma separated list of DNS suffixes which should be removed from any system host name. For example a value of *mqgem.com* would remove this suffix from any returned host names.
4. **Clearing the cache**
   The entire system host name cache can be cleared by pressing the 'Clear DNS Cache' button in the Connection tab of the Preferences Dialog. This will only clear system entries, user entries will remain unaffected.

5. **127.0.0.1**
   IP Address 127.0.0.1 is a special IP address which refers to the local machine itself. MO71 cannot resolve this IP address to an actual host name since the 'local' machine refers to the machine where the Queue is running; not the machine where MO71 is running. For this reason IP address 127.0.0.1 is always displayed as hostname 'localhost'.

6. **Availability**
   In order to do the reverse look-up MO71 needs to call the functions gethostbyaddr() and inet_pton(). In some environments these functions may not be available. In such cases the host name is given as '<Not Aavailable>'.

System values are a big usability improvement over an IP address such as 198.51.100.34. However, you can go further if you wish by using a DNS user value.

4.6.2. DNS User Values

Sometimes it can be useful to define your own text for the host names. This allows you to associate any information you might thinks is relevant. For example, you could include a persons contact name, a department name or phone number. A DNS User value will always be displayed in preference to a System value. DNS User Values are read from a file called **mqmon.dns**. This file is similar to **mqmon.dnc** above but this one is intended to be edited by users. This file has a very simple format such as the following:

```
* MQGem MO71 DNS User Values

198.51.100.34 Paul's main server x1435
198.51.100.27 Morag's laptop x2056

* IP v6
2001:0DB8:0000:0000:0000:0000:0046:0124 Linux test machine
```

Each line of the file should be either blank, a comment line starting with * or a DNS User Value line. The DNS User Value line consists of an IP address, either IPv4 or IPv6, followed by a space. The rest of the line is the value which is associated with the IP address. The file is read by MO71 upon start-up. If you change the file while MO71 is running you can tell MO71 to re-parse the file by pressing the 'Reload DNS File' button in the Connection tab of the Preferences Dialog.

4.7. MQSC Window

The MQSC window is provided to allow the user to issue commands not provided by MO71, for example, commands like **DISPLAY DQM** on z/OS. The screen is split into three main areas, the entry field at the top where you enter an MQSC command; the central area where the command server responses are displayed; and status messages, as usual, are shown at the bottom of the screen.

To make life a little easier the entry field is a drop down list box. The last 20 commands entered are kept as a history and can be recalled either by dropping down the list box or by using the up and down arrow keys. The command history will also be saved across invocations of MO71.

As with nearly all the dialogs the *'Auto Refresh'* option is available. This allows the user to type a command on the command line and have the program execute the command every few seconds. Care should be taken when doing this, however, since the response window maintains a list of **all** the responses from the command server since the start of the dialog, therefore this should be used in conjunction with the **cls** command below.

The command entered in the command field is interpreted by the program before being passed to the command server.

The following additional capabilities are allowed :-

Command	Meaning
cls	**Clear screen command**. This command will delete all data in the main window
=mqev	**Direct commands to MQEV processor**
=mqsc	**Direct commands to IBM MQ command server**
;	**Command separator**. Allow multiple commands to be entered on a single line Note that commands themselves should not contain this character
/	**Remove search string**. A find command with no following text will remove the highlighting of the search text.
/text	**Find text**. Issuing this command will find the next occurrence of 'text'. All instances of 'text' will be highlighted. This command stays in effect until the search string is removed by the command below
file(?)	**Bring up file selection dialog.** When a file is selected it will either be run against the command server or the file name copied to the input field depending on your preference setting.
file(infile)	**Execute MQSC script file**. Any '?' at the end of the *infile* will bring up a file selection dialog for the path, if any, specified in *infile*.
file(infile,outfile)	**Execute MQSC script file and write the result to the output file**. Any '?' at the end of either the *infile* or *outfile* will bring up a file selection dialog for the path, if any, specified in the filename.
>	**Execute command and write the result to the output file**. For example *dis q(*) > myfile.txt*
>>	**Execute command and append the result to the output file**. For example *dis q(*) >> myfile.txt*

4.7.1. MQSC Examples

To clear the screen, display the queue and channel called ‘FRED’ and then highlight all instances of the word ‘FRED’, use the following command :-

```
cls;dis q(FRED) usage;dis chl(FRED) chltype;/FRED
```

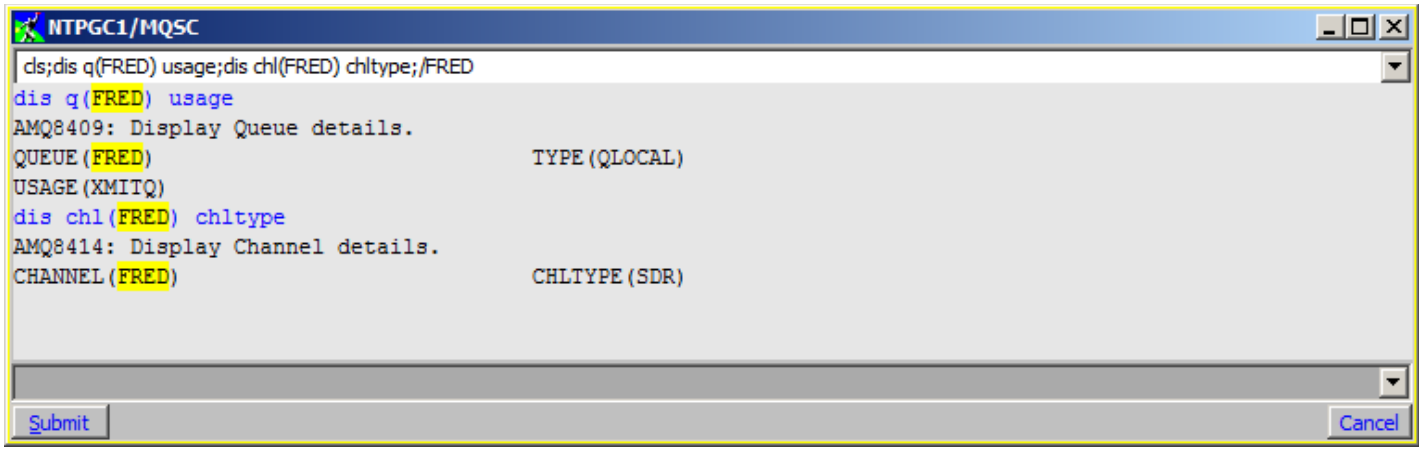


Figure 8: MQSC Window

The colour of the search text is settable in the 'Colour Dialog'.

With auto refresh enabled the following could be useful to constantly display the state of Distributed Queuing on a z/OS Queue Manager :-

```
cls; dis dqm
```

4.7.2. MQSC Clipboard

Three menu items are provided for copying the displayed data to the clipboard.

1. **Copy to clipboard**
   This option will the selected text to the clipboard

2. **Copy show to clipboard**
   This option will copy just what is displayed on the window to the clipboard

3. **Copy all to clipboard**
   This option will copy all of the data currently in the MQSC window history to the clipboard

In addition Ctrl-C can be used to copy the selected text to the clipboard. Ctrl-A can be used to select all the text.

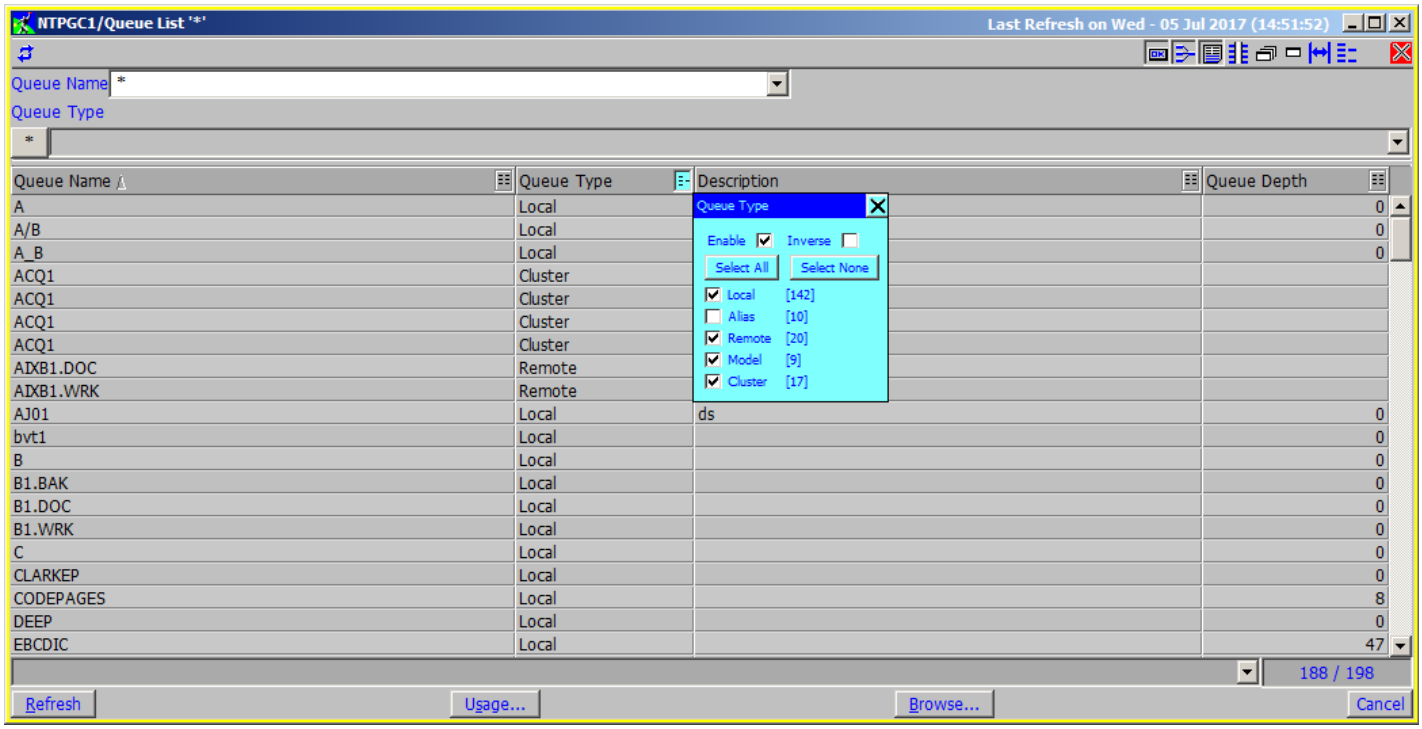
Chapter 5. Filtering

On Queue Managers where there are many definitions of the same object type it can be difficult to 'see the wood from the trees'. When you display a list of objects, say queues, we need is a way of limiting which objects are displayed. This is known as filtering. MO71 provides two ways of doing filtering, known as Column Filters and Filter expressions. There are advantages in each approach and which you use is entirely up to you. The advantages of each are:

- **Column Filters**
  Very simple click&display interface
  Show counts of enumeration fields. For example, you can easily see how many Channels are 'Stopped' or 'Retrying'.
- **Filter expressions**
  Very flexible, can be use to modify the display such as changing colours, adding and removing columns
  History of previous filters stored for later selection
  You can associate a default filter expression with a dialog

You can, of course, use both mechanisms at the same time. If you do then the filter expression will be applied first, followed by any column filters. The numbers at the bottom-right hand of the dialog will show the results after both types of filtering have been applied.

5.1. Column Filters



On the right-hand side of each column heading is a button which signifies that this column can be filtered. The icon on the button can be displayed in two ways. If both sides are the same then the filter is inactive. If there are less lines on the right-hand of the icon then the filter is active. In this case we have clicked on the 'Queue Type' filter and de-selected Alias queues from the list. This will automatically enable the filter. All filtering is done locally, MO71 will not go back to the Command Server for new data. Press Refresh on the dialog if you want a new set of data.

You can have as many column filters active as you like although only one filter window can be visible at any time. Selecting another column filter will close any previous column filter window for this dialog.

You can set the colour of the filter window by selecting *'View-Set Colours...'* from the main window. The filter windows colours are set by changing the 'Menu background' and 'Menu Foreground' colours. For information about setting colours please see Chapter 24.10.Colour Dialog on page 229. You can choose not to display column filters on a dialog if you wish by selecting *Options-Hide Column Filters* from the context menu.

5.1.1. Column Filter Types

As we know the different columns show different types of data. Each type of data will show a slightly different column filter window although the principals are the same. The following types are supported:

- **Character String such as Queue Name, Connection Name etc**
  You can type in a single string of characters. You can choose whether MO71 should merely search for these characters within the column or whether it should match the characters the with column entry. So, for example, suppose we entered the string **Q1** in the Queue Name column filter. It would display queues **Q1**, **FQ1**, **Q1TEST** if we had selected contains. But it would only display **Q1** or **q1** is we said matches. If we select 'Match Case' then it would only show Q1.

  We can go further with a *match* type filter and add wildcards. So, for example, we could enter the characters **Q*1** and this would show all queues that started with **Q** and ended with **1**. We can also use **?** Indicating a single character. So, for example, **????**, would match with any four character queue name.

- **Hex fields such as Message Id, Correlation Id**
  Hex strings are very similar to normal string except that the concept of 'Match Case' does not apply. You must enter a hex string and possibly wildcards for the match.

- **Numbers such as Heartbeat Interval**
  To filter by a number you are given a minimum and maximum value and for most numbers a list of the used values. If you enter a number in the minimum field then only entries with a value greater than or equal to this value will be shown. Similarly a number entered in the maximum field will cause on entries with a value less than or equal to this number to be shown.
  Most numeric fields will also display a list of the used values and how many of each are in use. Whether this is available depends on the type of field, whether it is status or whether it is definitional. For example, it is not available on 'Queue Depth' because the value keeps changing and filtering on the actual number would probably lead to unexpected results. If it is applicable though this feature allows you to easily select the values that interest you. Alternatively selecting 'Any value' will remove any selection based on a particular value and show entries of any value. By default only the 'in use' values are shown in the list of values but selecting 'Show All' will show any values that MO71 has seen but that are not in the currently displayed list. Pressing 'Refresh Values' will cause MO71 to forget any value that is not currently in use by this list.

- **Enumerations, such as Queue Type**
  There are essentially two types of enumeration fields in MQ. Those that have a small number of choices, such as queue type and those that have a large number of choices, such as feedback code.

  - **Small numbers of choices (15 or less)**
    A simple list of checkboxes is displayed. Each checkbox is followed by a number which is the count of items in that list. All possible choices are shown regardless of whether the list itself actually contains entries of that type.

  - **Large numbers of choices**
    If there are a large number of possible choices for this field then MO71 will, by default, only show the choices that currently exist in the displayed list. You can select the 'Show All' checkbox if you really want to see all possible values. The values are shown in a scrollable list.

- **Combined Date&Time fields**
  Traditional MQ Date and Time fields are just a set of characters and are treated as such. This can be useful if you want to do wildcard processing etc. However, since the combined Date&Time fields are an actual time their column filters can treat them as such. For more explanation please refer to 'Date&Time Column Filters' and page 30.

- **Flags, such as Connection Options**
  Flag fields are very similar to normal enumeration fields except that the field counts are handled differently. A single list entry could have more than one flag active. It follows therefore that the sum total of the enumeration counts will not sum to the number of items in the list.

The dialog list display will be updated immediately when most of the filter fields are changed. For example when an enumeration choice is selected or deselected. However, the application of the data entry fields, such as the search field and the minimum and maximum values is not immediate. For these fields MO71 will only update the list when no data entry has occurred for approximately two seconds. Alternatively you can signal that editing has completed by pressing *<ENTER>*. In which case the dialog list will be re-filtered immediately.

As mentioned before you can have as many Column Filters enabled as you wish. You can select '*Options-Remove Column Filters'* from the context menu (or dialog toolbar) to quickly remove all column filter filtering. MO71 will still remember the last filter values used it is just that the column filter itself will not be enabled.

5.1.2. Column Filtering Positioning

MO71 will attempt to put the filter window in a suitable place next to the column being filtered. If you move the dialog around or resize it then the filter window will move accordingly. If you wish to have the filter in a particular place though then you can click on the titlebar of the window and move the filter window to your preferred location. MO71 will remember the relative position of the window and the filter window will continue to be moved in accordance with size and position changes to the dialog.

5.1.3. Date&Time Column Filters

Because a Date&Time field represents an actual time we can filter by physical timestamps. Consider the following:

This filter will show only those items in the list which were altered on or after 9am April 1st 2019 but before 9am April 4th 2019. Note that the 'From' date is inclusive, but the 'Until' date is not.

The Date&Time Column Filter can be used in a number of different modes. The most obvious difference depending on whether the 'From', 'For' or 'Until' fields are selected. These modes are described in the following table:

From Selected	For Selected	Until Selected	Effect
X			Shows items from the given date until 'now'
X	X		Shows items from the given date for an elapsed time period.
	X		Has no practical meaning. All list items will be shown
		X	Shows items until the given date.
	X	X	Shows items for a time period leading to an explicit date.
X		X	Shows items from the given date up to, but not including, the Until date.

So, depending on which fields are 'active' the dialog can be used to select items on the list conforming to a variety of selections.

The fields themselves can be changed in a variety of ways. However they can only be changed while they are selected. Clicking on an individual element will place focus bars above and below the element. This allows you to know which field will be changed. You can change which element has focus by either clicking on a different element with the mouse or using the left and right keys. The value itself can be changed in any of the following ways:

- **Using the up and down keyboard keys**
  Holding down a key will continuously change the value
- **Using the mouse wheel**
- **Clicking on the focus bars.**
  Holding down the mouse will continuously change the value.

Note that you cannot have all three fields selected. However, if you have two fields selected when you are changing a value then the 3rd value will be changed accordingly. If an invalid value is entered, for example 30th Feb, then the field will be shown in red and the field will not be active in the selection.

5.1.3.1. Now

The Until time has a 'Now' button following it's value. This allows you to quickly revert the value of Until to the current time. In fact, while ever the Now button is pressed the 'Until' time will constantly be updated with the current time. This allows you to have an ongoing filter which shows you, for example, just what happened in the last 4 hours. Note that column filters only filter what is currently in the list, they do not cause data to be refreshed from the Queue Manager. Therefore, in order to do this, you must use it in conjunction with Auto-Refresh on the dialog. Auto Refresh allows you to specify the interval you wish to refresh, for example, every 5 minutes. Choose a sensible refresh interval, unless you are testing, do not refresh your dialog every second!

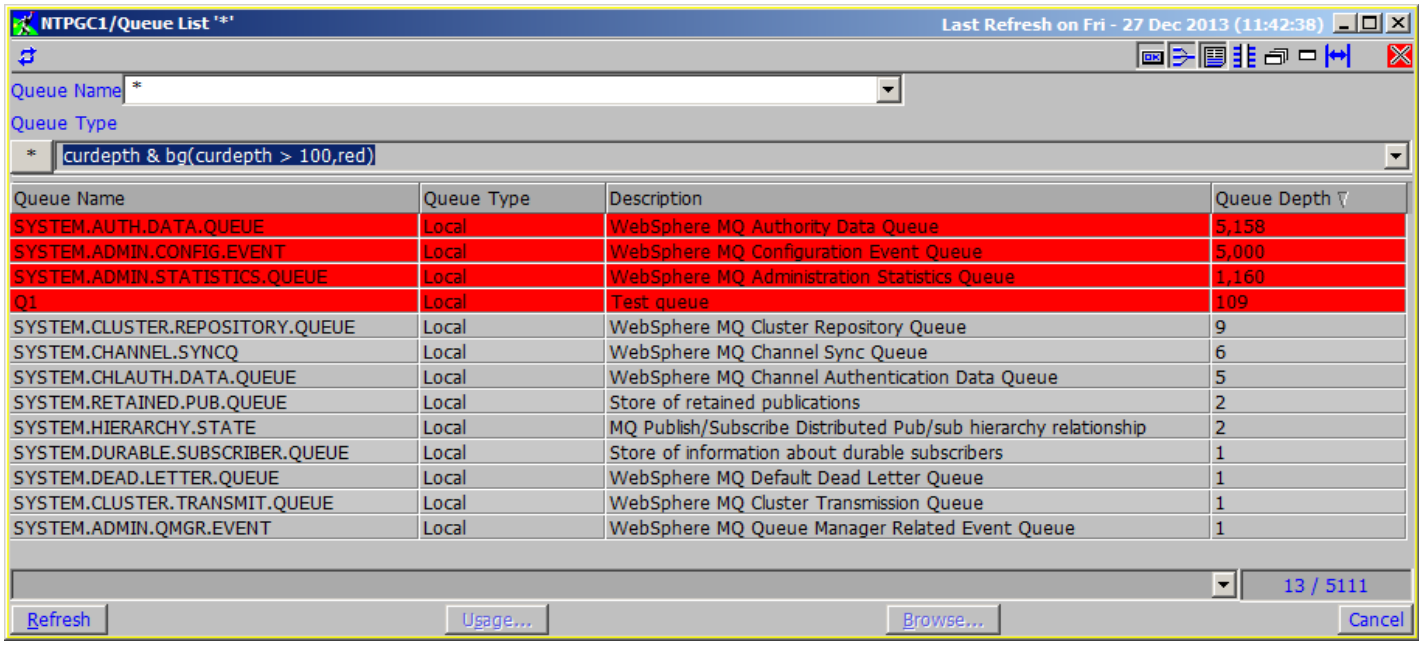
5.1.3.2. Local Time

You can choose whether the entered times should be treated as 'local' times or a time from the Queue Managers reference point. For example, something which happened at 9am my time or something which happened at 9am at the Queue Manager. Clearly this is only useful if your Queue Managers are on different time-zones than yourself.

5.2. Filter Expressions

Nearly every list dialog in MO71 contains a filter field in which the user can put a filter expression. At its heart a filter expression is merely a boolean expression which returns TRUE or FALSE for each object in the list which determines whether the object is displayed or not. However, filters can also be used to change the way the object is displayed.

Let's take as an example a queue list display. Suppose we just want to see the queues with messages on them and further we want queues with more than 100 messages on them to be highlighted in red. We can use the following simple filter.



Those of you with a mathematical leaning will understand that filter immediately but for others it may take a little playing with. All you need to remember that at it's heart a filter is merely a boolean expression which evaluates to TRUE or FALSE for each object in the list. So, this filter is saying the expression is TRUE if 'curdepth' is non-zero AND the result of function *bg()* is TRUE. Well, *bg()* always returns TRUE as we'll see later. So, this expression will be TRUE for any object with a non-zero current depth. Which is exactly what we wanted. All empty queues will evaluate to FALSE and therefore not be displayed. However, the calling of the *bg()* function clearly has another effect. b*g()* takes two parameters, the first is another boolean expression and the second is a colour, in this case the constant 'red'. I'm sure by now you can guess just how the *bg()* function works and could quite easily write your own filter which showed transmission queues in blue or retrying channels in yellow. Filters are very flexible and powerful as we will see in a minute.

Where possible, filtering is performed against the data cached from the command server. In other words, typing in an expression and pressing the '*' filter button will not necessarily cause a request to the command server. If all the information needed to satisfy the expression is already stored locally the filter will be performed on that information. Consequently the display of data is much faster. However, as a consequence the data displayed will not necessarily be the latest up-to-date data and users should periodically hit the *'Refresh'* button to download the latest information from the server. If the filter expression contains attributes that are not cached then a request for more information will be sent to the command server.

It is possible, in fact common, for a filter expression to contain field names that are not applicable to all entries in the list. If this is the case then the value of the field for the particular entry will be FALSE.

If the list is refreshed in some way with an active filter then the filter will be added to the filter history for this dialog so that is appears in the drop-down. It is also possible to have a global filter history so that it will appear in dialogs of the same type for other queue managers. If you wish to remove a filter from the filter history then press *Ctrl-Delete* while the text you wish to delete is in the entry field.

At the bottom right hand corner of the screen are two numbers. The first number gives the number of objects displayed in the list, and the second number gives the number of objects returned from the queue manager. If filtering is not active, these two numbers will be the same, however, with filtering there may be fewer objects in the list than were returned from the queue manager.

So, what are the rules for defining a filter?

The expression rules are as follows:-

- **The expression is free-format**
  e.g. "2+4" and "3 + 4" are both acceptable
- **Constants**
  Constants are integer values which have a recognised and predefined name. For example, the are a number of colours which are constants – **red**, **blue**, **green**, **yellow** and so on. The name is case insensitive eg. the value **RED**, **Red** or **red** can be used, even **rEd** if you are feeling obtuse.

  There are constants for boolean expressions such as **true** and **false**. A number of functions can take constants (or combinations thereof) as their parameters – constants like **left**, **right**, **col**, **bar** etc

  For a full list of constants please see Appendix B: Constants on page 270.
- **Numbers**
  Numbers can be entered as:
  - Integer 12,1234
  - Real 1.3,12345.5
  - Hex 0xFAB,0x10

- **Strings**
  Strings can be any sequence of characters between ‘ or “ characters. Special characters can be included in the string using the ‘\<value>’ convention.
  - \\ Backslash. This should be used, for example, when specifying file names with back slashes in them. For example use **c:\\temp\\file.dat** not **c:\temp\file.dat**
  - **\n** New line
  - **\”** Double quote
  - \’ Single quote
  - **\a** Alert
  - **\b** Back space
  - **\f** Form feed
  - **\r** Carriage return
  - **\t** Tab
  - **\v** Vertical tab

  For example :-

  - "SYS.*"
  - 'ABC'
  - "This string has a \" in it"

- **Statements**
  A filter can consist of multiple statements joined together using a ‘;’ character. The value of the statements is the last statement in the list. For example the filter **status(“Hello World”);0** has the value 0.

- **Functions**
  Various functions, such as the *bg()* function we've already seen, are provided which can be called to do variety of tasks.

5.2.1. List Variables

Any attribute name of the object being displayed may be used in the expression. The attribute name is normally the MQSC name of the attribute and only enough of the name to ensure uniqueness need be specified. To see the identifier that should be used look in the *'Alter List'* dialog where the complete list of attributes and their identifiers are given. Alternatively if it's a displayed field you can hover the mouse over the column title and a tooltip window will be displayed giving the full name of the field and the MQSC attribute name which should be used in filter expressions.

It is not necessary to only use fields from the currently displayed list. Note that the identifier name is not always the MQSC name since some fields don't have one. For example, Trigger Control doesn't have an actual MQSC attribute name it only has a value TRIGGER or NOTRIGGER. In this case you can use the value of TRIGCTL to identify this field. The same is true of fields like Harden Get Backout and Shareability.

It is often useful to be able to temporarily see a value in the list without going through the process of actually changing the list items using *'Alter List'* described earlier. In this case you can follow the variable name with a '#' (hash) character. This will cause that field to be displayed if it's not already in the list. The columns will be displayed after the defined columns for the list in the order they were written in the expression. As soon as a filter is entered without this field followed by a hash then the field will be removed from the display.

So, for example the following filter will add the usage field to a queue list but still display all queues.

```
usage#;true
```

The effect of adding the ';true' to the end of a filter is to ensure that it always returns TRUE. If we had just used a filter of

```
usage#
```

Then only queue with a non-zero value for their usage field would be displayed. This essentially means that only transmission queues would be shown.

Some other example of field names in a Queue List are as follows:

Variable	Effect
Queue	Queue Name
Qtype	Queue type
Type	Queue type[6]
Usage	Usage value
Usage#	Usage Value and display Usage in the list display
Curdepth	Current Queue depth
Cur	Current Queue depth
Cu	Current Queue depth
rname	A remote queues remote queue name. Note that an expression using this field will only show remote queues since this field only applied to remote queues.

[6]Note that Queue Type is known by both 'Qtype' and 'Type' in different MQSC commands so either value can be used to refer to this field.

5.2.2. Operators & Flow control

Normal operator operation and precedence apply :-

Operator	Meaning
+, -, *, /	Addition, Subtraction, Multiplication, Division The '+' operator is also used to concatenate strings.
mod	Modulus
=, >, >=, <, <=, <>, !=	Boolean comparisons
& , AND **| , OR** **! , NOT**	Boolean AND Boolean OR Boolean NOT
&& ||	Bitwise AND Bitwise OR[7]
==	String wild card matching The second operand is specified as a string containing the following wild card characters: - '*' Matches any number of characters (including 0) '?' Matches exactly 1 character.

- **Conditional branches**
  It is possible to take conditional branches using the following construct:-
  if (expression) {statements} [else {statements}]

- **Coercion**
  If two different operand types are involved in a sub expression the type of one or more of the operands will be changed. Most notably if strings are used in expressions other than comparisons then the string length is what is used.

  For example, the filter **"desc"** is TRUE only if the object has a non-blank description. So for example the filters **queue == "???"** and **queue = 3** will both display the list of queues with only three character names.

  As another example, **SORT(queue)** will sort the list in alphabetic queue name order but **SORT(queue+1)**, since we're forcing the queue to be treated as a number, will display the list in order of the length of their name. A similar effect is achieved by using **SORT(+queue)**.

- **Invoking other filters**
  If a filter has been given a name then it can be invoked from another filter using the expression **$<filter_name>**. Note that this means that filter names cannot contain blank characters.

[7] Please note that the bitwise and logical operators are the other way round from languages such as 'C'. This is due to the fact that most of the time the single operator, logical version, is required.

5.2.3. System Variables

System variables are predefined variables which are initialized to a value before being used in the filter.

- **_first** An integer which is set to 1 (or true) each time the filter is matched against a set of entries. This integer allows you to do some initial processing. Note that the filter should set the value of **_first** back to 0 once initial processing has completed with the statement.
  For example, **_first := 0;**
- **_last** An integer which is set to 1 (or true) only for the last entry in the list. The **_last** system variable is most useful for reporting results.
  For example, **if (_last) status("All done now");**
- **_time** An integer which contains the current time. This can be useful for calculating the different in time between successive invocations of the filter or for performing different processing in the filter depending on the day of the week or time of day.
- **_index** The index of the current item in the list starting at 0.
- **_items** The number of items in the list
- **_qmname** The name of the Queue Manager
- **_locname** The name for this location
- **_scan** The scan instance. This field can be used to discover how many times this current data has been scanned. This can be used in conjunction with **_rescan** to examine all data in scan 0 before deciding what to return in scan 1.
- **_rescan** Whether another scan should be performed. A filter can set this value to true to force a re-scan of the current entries. This can be useful to collate information about the data in the first scan, then during the re-scan the filter can use this collated data to make decisions about the data.

5.2.4. User Variables

User variables begin with the character '@'. They may be used to store information either between items in a list or between separate invocations of the list filter. Different data types of user variables are allowable by following the '@' by an optional character.

- **@x := 1** Declares x as an integer variable
- **@@x := 3.2** Declares x as a real or float variable
- **@$x := "Hello"** Declares x as a string variable
  String variables may only be a maximum of 200 characters.

As shown in the example above, assignments are made using the '**:=**' operator. Note the use of the colon character to distinguish the assignment from a comparison.

For example, **@x = 3** is comparing **x** with the value 3 whereas **@x := 3** is assigning **x** the value 3.

5.2.5. User Variable Inserts

Sometimes it may be useful to have a variable for each object in the list. There are three variants of these inserts. Essentially the object name will be included in the variable name but the three different styles allow for slightly different behaviour.

The insert characters are as follows:

- **%n** **Inserts just the object name and nothing else**
  This insert should be used with caution since it can clash with user defined variables.
  For example for a returned queue 'Q1' the variable @varQ1 and @var%n are the same.
  .
- **%o** **Inserts the object name qualified with the Queue Manager name**
  This insert is generally the one to use. It provides a unique variable for each unique object name across all the Queue managers.

- **%O** **Inserts the object name but qualifies it as a 'created' variable.**
  This insert does not qualify the object name with a Queue Manager name therefore Q1 from Queue Manager QM1 will generate the same variable at Q1 from Queue Manager QM2. However, the variable generated will always be distinct from user variables.
  For example for a returned queue 'Q1' the variable @varQ1 and @var%O are not the same.

String variables may only be a maximum of 50 characters.

For example, **@depth%o := cur** will assign the queue depth to a unique variable. Note that the uniqueness of the variable is only as unique as the object name itself. When used in a queue list it is possible for there to be more than one entry with the same queue name. A local queue and a cluster queue can appear in the same list. Similarly it is possible to have multiple entries in a channel status list each with the same channel name.

Normally insert %o should be used but if the same variable should be used, for the same object name, for all Queue Managers in a multi-Queue Manager dialog then the %O insert should be used.

5.2.6. Functions

- **alarm(Expression)**
  Causes the application to alarm, always returns TRUE. If 'Expression' evaluates to TRUE then the application will issue an alarm BEEP every few seconds provided alarms are switched on for the application.

- **all(Expression1, Expression2, Expression3,.....)**
  Takes 1 or more parameters. The **all** function is used for temporarily adding a new column to a list display. For example a filter of **'all(maxdepth#)'** will add max depth to the list of columns displayed but will still display all queue definitions regardless of whether maxdepth is applicable to that type of queue.

- **any(Expression1, Expression2, Expression3,.....)**
  Takes 1 or more parameters. The **any** function is useful for displaying fields which can contain any value. For example a filter of **'trigctl#'** will display a column for Trigger Control but will only display those items for which the value of Trigger Control is non-zero, it is equivalent to 'trigctl# <> 0'. There are times when you want to see the value displayed regardless of what it is. A filter of **'any(trigctl#)'** will display those queue definitions for which Trigger Control is a valid attribute regardless of what that value is.

- **beep(Expression)**
  Causes the application to beep, always returns TRUE. If 'Expression' evaluates to TRUE then the application will issue a BEEP.

- **bg(Expression, Colour)**
  Sets background colour, always returns TRUE
  If 'Expression' evaluates to TRUE then it sets the background colour of the object in the list
  The Colour parameter can be an RGB value or one of the many Colour constants such as blue, green, yellow etc. Please see Colours on page 272 for the full list.

- **bgcell(Expression, Colour, Cellname)**
  Sets background colour of a single cell, always returns TRUE
  If 'Exp' evaluates to TRUE then it sets the background colour of just the specified cell in the list.
  For example, **bgcell(cur, yellow, queue)** will set the background colour of just the Queue Name for any queues which have messages on them. The Colour parameter can be an RGB value or one of the many Colour constants such as blue, green, yellow etc. Please see Colours on page 272 for the full list.

- **csl(Expression, Priority, Error Identifier [, Category [, Text]])**
  This function allows the user to and and remove items to the console window. This function can be called with 3, 4 or 5 parameters but the parameters must always be in the order above. In addition to the above parameters the console entry will contain the name of the current object the filter is processing.

  The **Expression** parameter controls whether this entry is either added or removed from the console. If the expression evaluates to TRUE the entry will be added to the console, if it evaluates to FALSE it will be removed from the console. If the function is adding to the console then it will replace any previous entry which is the same. If the function is removing from the console then it will remove and and all entries which are considered the same. In order for an entry to be considered the same then the values of **Object Name**, **Error Identifier** and **Category**, and possibly **Text** (see below) must all be the same.

  The **Priority** parameter sets the priority value to be used for the entry in the console. Constant values such as **info, low, medium** and **high** may be used for this parameter.

  The **Error Identifier** parameter can either be a IBM MQ error code or a string error message. This error identifier, or it's string equivalent, will be shown in the console window. Note that although the parameter is called “Error Identifier” it need not necessarily signify an error. It could be any situation that you wish to bring the operator's attention to.

  The **Category** parameter is a free format, optional, string of your choosing that can be useful to group similar entries together. It can help with filtering the output.

  The **Text** parameter is a free format, optional, string which allows you to give a more detailed description of the 'problem' should you wish. Ordinarily the value of 'Text' must match an existing entry to be considered the same. However, if the text starts with the exclamation mark character (!) then the text will match any value. Note that in this circumstance the exclamation mark itself is not included in the entry. Starting your text with an exclamation mark is a useful way, therefore, to deal with entries which a logically the same but have different text strings (see the worked example below).

  So, as an example of the use of **csl()**, suppose we have a queue list and we wish to have the queues which have more than 100 messages on them displayed on the console. We could use the following filter :-

  csl(curdepth > 100,medium,"Loaded queues")

  Note that the console entry will automatically have the name of the current object (a queue in this case) included in the entry. So, the above will work but we could be more explicit with something like:-

  csl(curdepth > 100,medium,"Loaded queues","DEPTH","Depth is greater than 100")

  So, now the operator knows a little more about why this console entry is there but it doesn't really tell them how bad the issue is. The obvious thing the operator might want to know is just how deep is the queue. We can add this to the console entry very easily like this :-

  csl(curdepth > 100,medium,"Loaded queues","DEPTH","Depth is greater than 100 (" + str$(curdepth) + ")")

  However, this will not quite work correctly since every time the depth of a queue changes we will get a new console entry for the same queue and previous console entries will not be deleted when the **Text** is different. So, over time we will get many entries for the same queue all with different current depths. The solution to this is very simple, it merely requires us to start the text with the exclamation mark (!) character. This tells the csl() function that this definition replaces all matching entries even if the text itself doesn't quite match. The correct filter to use is therefore:-

  csl(curdepth > 100,medium,"Loaded queues","DEPTH","!Depth is greater than 100 (" + str$(curdepth) + ")")

- **day(time)**
  Returns day of the month represented by the time parameter. Returned value from 1->31.

- **date$([time [,format]])**
  Returns the date as a formatted string.
  This function takes 0, 1 or 2 parameters. If no parameters are given the current time is returned in the default format. If just the time parameter is passed then that time is returned in the default format. If both a time and format is given then the returned value is the time in a formatted string according to the following values for the format string.

Insert	Meaning
H	Two digit hour (24 hour clock)
HH	Hour (24 hour clock)
h	Two digit hour (12 hour clock)
hh	Hour (12 hour clock)
M	Two digit minutes
S	Two digit seconds
d	Two digit day of month
dd	Day of month including suffix eg.1st, 2nd, 3rd …
j	Julian day of year (zero based)
J	Julian day of year (one based)
m	Three character month name eg.Jan,Feb,Mar….
mm	Two digit month
mmm	Full month name eg. January, February,March…
P	AM/PM
p	am/pm
y	Four digit year
yy	Two digit year
D	Three character day of week eg.Mon,Tue,Wed…
DD	Full character day of week eg. Monday, Tuesday, Wednesday…
t	Simple time format eg. 18:14:03
<char>	Escape character sequence. eg. \\m will print 'm'

  The default format is : **H:M d/mm/y** eg. 22:10 05/10/2006

- **day$(time, type)**
  Returns day, as a string, represented by the time parameter. The type parameter controls the format of the returned string.
  - **Type = 1**
    Sun,Mon,Tue,Wed,Thu,Fri,Sat
  - **Type = 2**
    Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday

- **exists(variable)**
  Returns TRUE if the variable given exists in the list. For example the filter **exists(curdepth)** would return TRUE for all local queues but return false for other queue types such as Alias and Remote.

- **fclose(fd)**
  Closes the file descriptor.

- **fg(Expression, Colour)**
  Sets foreground colour, always returns TRUE
  If 'Expression' evaluates to TRUE then it sets the foreground colour of the object in the list.
  The Colour parameter can be an RGB value or one of the many Colour constants such as blue, green, yellow etc. Please see Colours on page 272 for the full list.

- **fgcell(Expression, Colour, Cellname)**
  Sets foreground colour of a single cell, always returns TRUE
  If 'Expression' evaluates to TRUE then it sets the foreground colour of just the specified cell in the list. For example, fgcell(cur, red, queue) will set the foreground colour of just the Queue Name for any queues which have messages on them.
  The Colour parameter can be an RGB value or one of the many Colour constants such as blue, green, yellow etc. Please see Colours on page 272 for the full list.
- **findstr(String, SubString)**
  Searches for SubString in the String. If it is found the function returns a 1 based offset of the position. If the string is not found the function returns zero (FALSE). The search is case sensitive.
- **findstri(String, SubString)**
  Searches for SubString in the String. If it is found the function returns a 1 based offset of the position. If the string is not found the function returns zero (FALSE). The search is case insensitive.
- **flash(Expression)**
  If 'Expression' evaluates to TRUE then it will periodically flash the line, always returns TRUE
- **flashcell(Exp, Cellname)**
  If 'Expression' evaluates to TRUE then it will periodically flash specified cell, always returns TRUE
- **fopen(FileName [, filemode])**
  Returns file descriptor which can be passed into file routines, or 0 if open failed.
  If the file name contains the '\' character remember to use '\\' instead.

  Optional file mode parameter controls how the file is opened. Valid values are :-
  - **"r"** Open for reading
  - **"w"** Open for writing (Default if filemode not specified)
  - **"a"** Open for append
  - **"r+"** Open for reading and writing but file must exist
  - **"w+"** Open for reading and writing. Any current file will be overwritten**.**
  - **"a+"** Open for reading and appending
  - **"b"** Open in binary mode
  - **"t"** Open in text mode [Default if neither "b" or "t" specified]

- **fprint(File, Expression1, Expression2, Expression3,…….)**
  Writes the expressions to a file.
  The file parameter can either be a file name or a file descriptor returned from **fopen()**
  If the file name contains the '\' character remember to use '\\' instead.
- **fprintf(File, Fmt, Expression1, Expression2, Expression3,…….)**
  Writes the expressions to a file a formatted string.
  The file parameter can either be a file name or a file descriptor returned from **fopen()**
  If the file name contains the '\' character remember to use '\\' instead.

  For example the filter **fprintf("c:\\temp\\mo71.out","%50s %d\n",queue,cur)** on a queue list dialog will write a file containing the queue name and current depth of all local queues.
- **hidecol(ColumnName)**[8]
  This function will hide the names column from a list display. For example, **hidecol(qtype)** will suppress the display of the Queue Type column in a Queue List display. A special value 'all' can be used to hide all columns.
- **hour(time)**
  Returns the hour portion of the time parameter
- **htmlfile(filename)**
  This function takes a single file-name, as a string parameter. The function only has an effect in filters run from a browser. The function allows the filter to override the default HTML file which would be used as the basis of the response. This can be useful if, for certain filters, you wish to override the HTML data returned or have a different look and feel to the web page. Another reason could be to add a meta tag to the HTML file. Consider the case where we would like the browser to refresh the list of queues every 30 seconds. We could just edit the queues.html file and add the meta tag. However, we would like to be able to control, via the URL, whether the refresh is active. So, what we do is create a new HTML file, let's

[8] This function is unusual in that it is not run against each entry in a returned list – instead it is run once, initially, when the filter is defined.

call it ***queues_refresh.html*** that looks something like this:

```
<html>
<head>
<body>
%[include header.html]
<meta http-equiv="refresh" content="30" />
%[content]
</body>
</head>
</html>
```

You can see that we have added meta tag information which tells the browser to refresh the page every 30 seconds. We could, of course, have altered the file in any other way altering it's appearance for example. Now, all we need to do to use this file rather than the default is to specify a URL like the following:
http://localhost/mq/admin/Queues/?filter="htmlfile("queues_refresh");cur"

This URL will display the queues which have messages on them for the default HTTP Queue Manager and the page will be refreshed every 30 seconds.

- **max(Expression1, Expression2)**
  Returns maximum value of the two expressions
- **min(Expression1, Expression2)**
  Returns minimum value of the two expressions
- **minute(time)**
  Returns the minute portion of the time parameter
- **month(time)**
  Returns month of the year represented by the time parameter. Returned value from 0->11.
- **mon$(monthindex, type)**
  Returns month of the year, as a string, represented by the month index (0->11) parameter. The type parameter controls the format of the returned string.
  - **Type = 1**
    Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
  - **Type = 2**
    January, February, March, April, May, June, July, August, September, October, November, December
- **mqtime(Date, Time)**
  Accepts date in YYYY-MM-DD format and time in HH.MM.SS format and returns number of seconds since 1970 as a long.
- **piechart(ColumnName)**
  Display a pie-chart of the given field. This function will also add the given field to the list of displayed fields in the list itself. This function can be useful in combination with predefined dialogs to allow a predefined display of pie-chart of a list field.
- **qm(Pattern1, Pattern2,….)**[9]
  Associates the list dialog with all queue managers names matching the patterns. Wildcards are supported. Passing no parameters will reset the Queue Manager selection back to the original single Queue Manager.
- **qmloc(Pattern1, Pattern2,…)**[9]
  Associates the list dialog with all queue manager location values matching the patterns. Wildcards are supported. Passing no parameters will reset the Queue Manager selection back to the original single Queue Manager.
- **qmnet(Pattern1, Pattern2,…)**[9]
  Associates the list dialog with all queue manager network name values matching the patterns. Wildcards are supported. Passing no parameters will reset the Queue Manager selection back to the original single Queue Manager.

[9] This function is unusual in that it is not run against each entry in a returned list – instead it is run once, initially, when the filter is defined.

- **second(time)**
  Returns the second portion of the time parameter
- **set(field, Expression)**[9]
  Sets the list field to a particular value. This can be useful to change the default values for a list dialog. For example, by default the connection list dialog shows ‘connection’ based values, but by using filter ‘set(type,handle)’ the handle based values will be displayed. By saving this filter as the default the defaults for the dialog can effectively be changed. This is also very useful to do in a Predefined Dialogswhere you might want to change the default settings of any dialog fields.

  The field name to use is the MQSC spelling of the keyword. Usually this will be obvious, or can be found by hovering the mouse pointer over the equivalent column in the list display. For those fields where this is not obvious, please see Appendix D: Filter function 'set' - Field names on page 275.
- **showcol(ColumnName, Column)**[9]
  This function will display the required column at the given Column number. For example, **showcol(maxmsgl,4)** will display the MAXMSGL values in the fourth column, provided that any of the displayed objects have a value for MAXMSGL. If the Column value is zero or less then the column position is not changed.
  The column position should be specified as though the dialog were not a multi-Queue Manager display. In other words the Queue Manager name column is ignored, unless, you are setting the column of the Queue Manager field itself.
- **sort(Expression)**
  Sorts the list in ascending order of the expression. Note that since this is an explicit sort it will effectively disable primary sorting by the user by clicking on the list titles.
- **sortd(Expression)**
  Sorts the list in descending order of the expression. Note that since this is an explicit sort it will effectively disable primary sorting by the user by clicking on the list titles.
- **sort2(Expression)**
  Specifies the secondary sorting expression in ascending order. Note that since this is an explicit sort it will effectively disable secondary sorting by the user by clicking on the list titles.
- **sortd2(Expression)**
  Specifies the secondary sorting expression ni descending order. Note that since this is an explicit sort it will effectively disable secondary sorting by the user by clicking on the list titles.
- **sound(Expression, Wave file)**
  Causes the application to play wave file, always returns TRUE. If 'Expression' evaluates to TRUE then the application will play the wave file.
- **status(Expression1, Expression2, Expression3,…….)**
  Writes the expressions to the status line of the dialog.
- **statusf(Fmt, Expression1, Expression2, Expression3,…….)**
  Writes the expressions to the status line in a formatted string.
- **str$(Expression)**
  Return a string representation of whatever parameter is passed in.
- **substr$(String, Offset, Length)**
  This function returns the substring of the given String at the given offset and length. The start of the string is at offset 1. An Offset value of -1 requests that the function returns the substring from the end. For example substr$(“ABCD”,-1,2) will return “CD”.
  A Length value of -1 requests that the function returns as much of the string as is left. For example substr$(“ABCD”,2,-1) will return “BCD”.
  The substr$ function will return a maximum of 200 characters.
- **system(Expression, Command String)**
  If the expression is TRUE then the Command is executed, always returns TRUE. This is useful if you want to check for out of line situations for example a Channel being in STOPPED state or a Queue Depth being larger than 100,000. The command is the name of an EXE program on the local machine and can be followed by parameters. The program will be started as a background process.
  e.g. command(cur>100000,"LOG.EXE -t \"Queue %o is in trouble!\" ")
  For more information refer to “Command Strings” on page 109.

- **usrcol(Name, Value, Options, MaxValue, Colour1, Colour2, FgColour, BgColour)**
  This flexible function allows the user to display chart elements such as a bar chart as well as a number of other capabilities. For a complete description please see Function - usrcol(Name, Value, Options, MaxValue, Colour1, Colour2, FgColour, BgColour) on page 44.
- **weekday(time)**
  Returns day of the week represented by the time parameter. Returned value from 0->6 where 0 represents Sunday.
- **where("*where expression*")**
  The where() function takes a single string as a parameter and is executed **before** the command is issued to the Queue Manager. It allows the user to include a WHERE clause expression in the Queue Manager request. This can allow the request to be more efficient than the Command Server sending every object and performing all the filtering locally. For example, suppose in a Queue List you wanted just the queues that contain messages you could use a filter of **where("curdepth gt 0")**. MO71 also allows you to use the simpler maths operators such as >, >=, <, <=, =., <>, !=. It also allows only as much of the attribute as makes it unique to be specified. So, for the same result, you could also say **where("cur>0")**. For simplicity MO71 also supports the shorthand of just the attribute name. So, for example a filter of **where("cur")** is the equivalent of **where("cur ne 0")** and **where("descr")** is the equivalent of **where("descr ne ' '").**
- **yearday(time)**
  Returns day of the year represented by the time parameter. Returned value from 0->365.

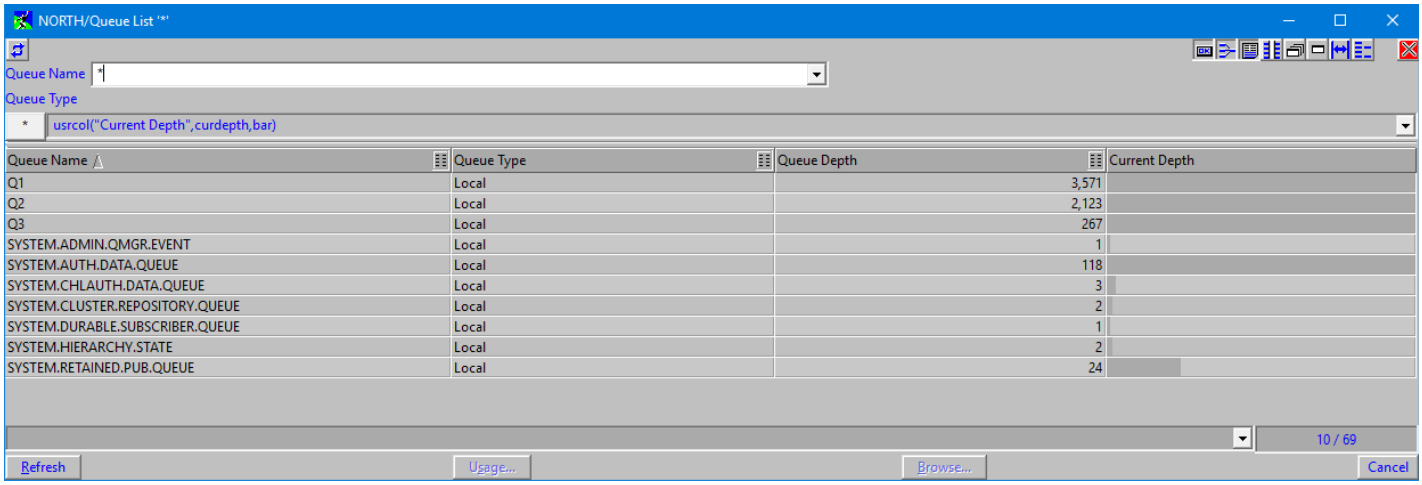
5.2.6.1. Function - usrcol(Name, Value, Options, MaxValue, Colour1, Colour2, FgColour, BgColour)

This flexible function, meaning 'User Column', is primarily designed to allow the user to add a new column, containing either text or a chart element such as a bar chart, to a list dialog. It can take up to 8 parameters, however, only the first 3 are mandatory. The user may provide anywhere from 3 to 8 parameters but any given parameters must be in the given order. To begin with let's just consider these first 3 mandatory parameters.

The first parameter is just a name that the column is known by. This can be any string but must be unique for each list. The second parameter is the value that the column should take. The third parameter are a set of flags which control how the column is displayed. Let's suppose we want to add a column to our Queue List display. We could have the following filter

```
usrcol("Current Depth", curdepth, bar)
```

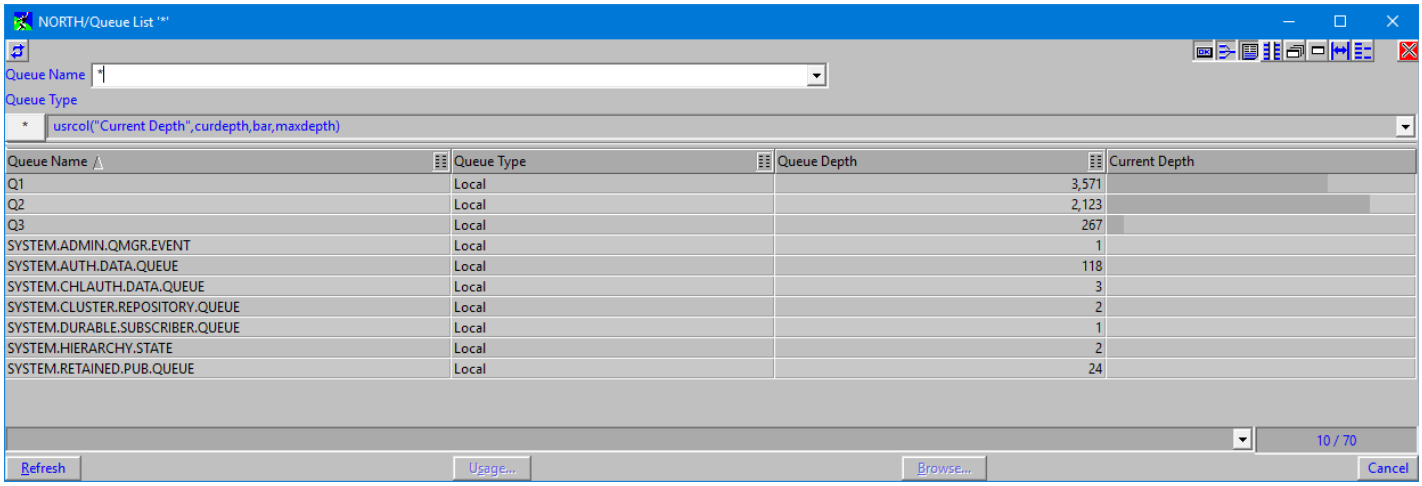
This very simple call to the function just says you want to have a new column, called “Current Depth” which displays the current depth as a bar chart. We might get an output such as the following:



We can see that we now have a new column. However, the bar chart isn't quite how we wanted it. What we wanted (probably) is that the bar chart shows how full the queue is. So what has gone wrong ? Well, we haven't provided a maximum value so the usrcol() function assumes a default of 100 for the maximum value. What we really need to do is to add another parameter to the usrcol() call. Let's tell the function what the maximum value is.

```
usrcol("Current Depth", curdepth, bar, maxdepth)
```

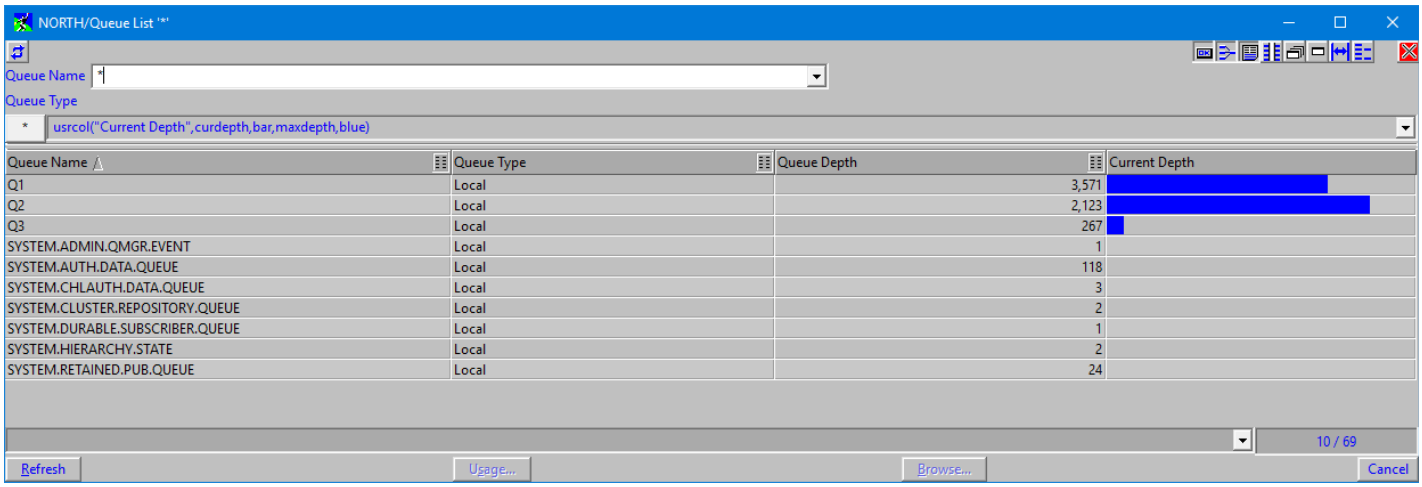
This filter will give us the following, much better, display:



The astute amongst you will immediately see a slightly surprising result. Although Q2 has a lower queue depth than Q1 it is displayed in the chart as a longer line. So, what is going on here. Well, as you might imagine the definition of Q2 has a much lower maximum depth – a value of 2500 as opposed to 5000. This is the power of a chart type display you can very quickly see the relationship between two values. So, let's continue to expand our call to usrcol(). Some people might find the default colour of the chart a little too plain – it is the list dialog title colour from your colour scheme. So, let's make it stand out a little more. Suppose we have the following:

```
usrcol("Current Depth", curdepth, bar, maxdepth, blue)
```

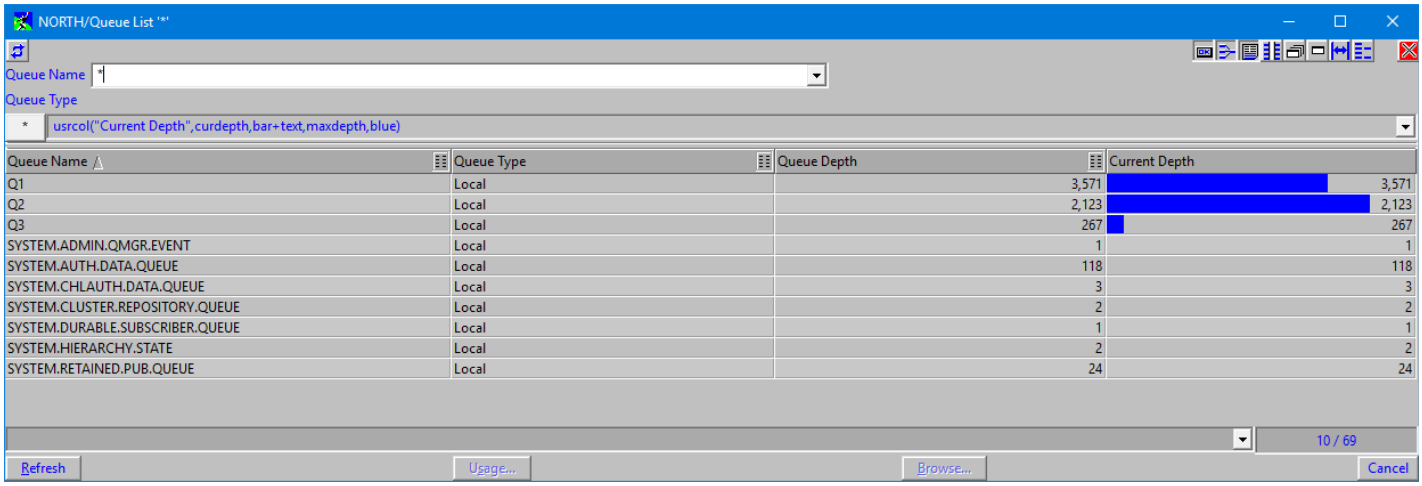
Now our chart shows up very clearly. Of course the colour you might want to choose will depend on your colour scheme.



Suppose we also want to display the value as a text item in the column, can we do this? Well, we can, all we need to do is to add another option '**text**' to the call. Each of the usrcol() options are essentially bit fields. You pass multiple options by OR-ing them together. The bitwise OR instruction is || (double vertical bar). However, adding two options together (as long as you only do it once) has the same effect and many people find that easier to understand. So, it is entirely up to you. You can either specify an option of **bar** || **text** or **bar** + **text**. We will use the latter format because it is a little easier to read. Just remember that what you are really doing it OR-ing options together and each option should only be specified once[10].

So, let's request the text is also displayed:

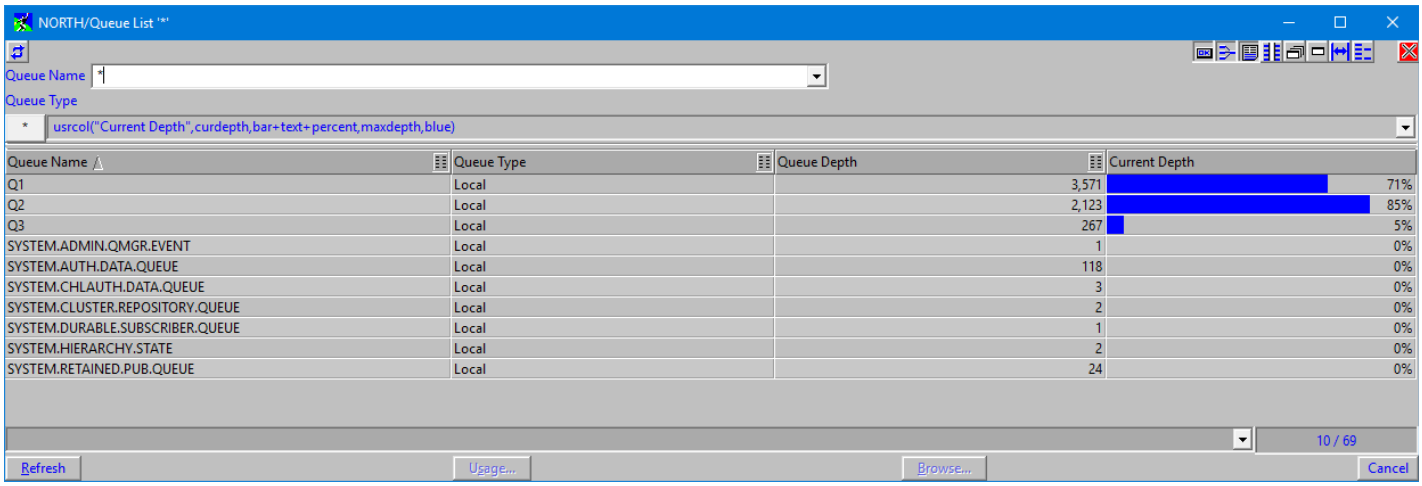
```
usrcol("Current Depth", curdepth, bar + text, maxdepth, blue)
```



[10]In other words **bar||text||text** will still work but **bar+text+text** will not because **bar+text** does not have the same value as **bar+text+text**.

So now the user column contains the 'value' of the column. Not surprisingly this is current depth of the queue for that line. Not a surprise because, after all, that is what we passed in. This could be useful if you wanted to replace the Queue Depth column with a display of this format. However, if you are adding the column in addition to the Queue Depth column then it would make more sense perhaps to display the text as a percentage. As you might have guessed this is simply a question of adding the percent option.

```
usrcol("Current Depth", curdepth, bar + text + percent, maxdepth, blue)
```

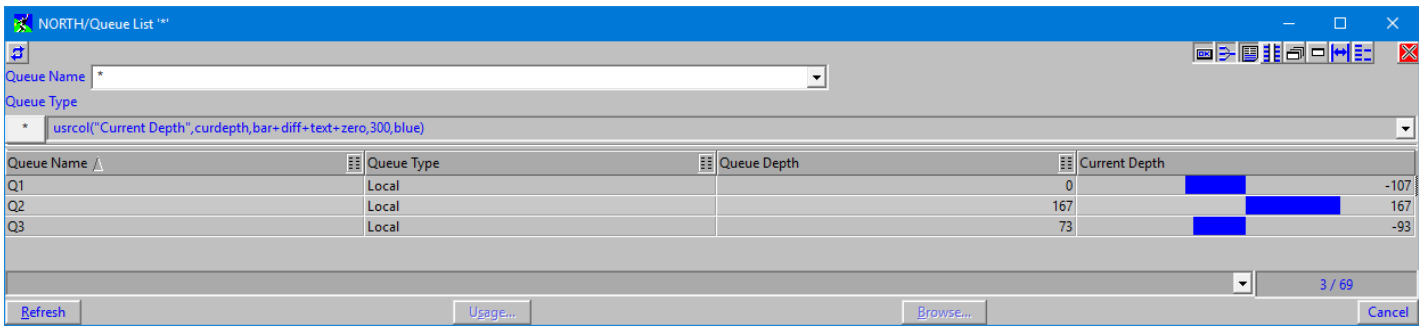


Now we get the percentage 'fullness' as we wanted. Note that the percentage is calculated to the nearest percent and many of our queues are not even 1% full. However, even though they have a value of '0' notice that they are still displayed in the list. This is a clue that even though we are displaying as a percentage the function itself is still returning the actual value – ie. The Current Queue Depth. So, even if a queue has only a single message on it it will still be in the list.

Of course we may not be interested in a value directly but may be more interested in the difference since last time we looked. For example, just graphing the 'Bytes Sent' value in Channel Status list would just show an ever increasing number. Not really very useful. However, graphing the difference in value since last time will tell you how many bytes have been sent in that interval. In our example here we might decide to graph the difference in depth of the queues. To do that we need to introduce a couple of new options. The option '**diff**' tells usrcol() to act upon the difference in value between the current value and the previous value. The option '**zero**' tells usrcol() to plot the bar chart around the zero point. Clearly if we are showing the change of depth of a queue this value could either be positive or negative so it makes sense to show the zero point in the centre of the bar. This would not have been the case in the Channel Byte Sent example since the number of bytes sent by a single channel can only go up. So, let's change our Queue List display to use the following filter:

```
usrcol("Current Depth", curdepth, bar + diff + text + zero, 300, blue)
```

Note that as well as adding a couple of new options we have changed the maximum value. The value now effectively controls the sensitivity of the display. If we left it at the MaxDepth value then only large changes in Queue Depth would be visible. The value you might want to use for any chart will depend on the maximum expected change within the time interval. With some applications processing our test queues we got the following display. Note that the very first time you use the '**diff**' option there will be no previous value so the change will not be reflected, you need to get at least two points on the graph.



So, here we get a good indication of whether messages are being added or removed from the each queue. Queues for which there is no activity are not shown. Of course, we could easily add those queues by ensuring that the filter returned a suitable values. For example, if we wanted only the queue that contain messages we could say:

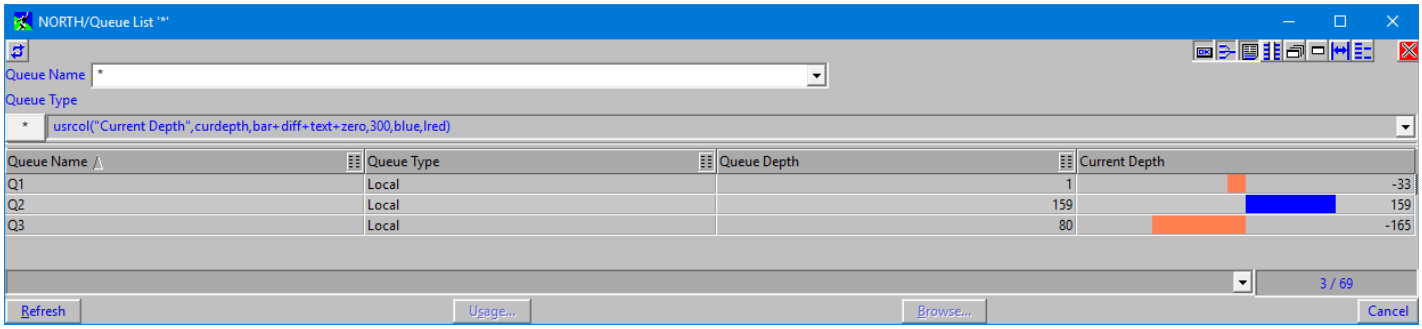
```
usrcol("Current Depth", curdepth, bar + diff + text + zero, 300, blue); curdepth
```

If we wanted all queues to be displayed, even Alias queues etc then we could could say :

```
usrcol("Current Depth", curdepth, bar + diff + text + zero, 300, blue); true
```

However, let's consider how else we could display the data. Well, first of all we can change the colour of the negative bar by adding another colour to the call.

```
usrcol("Current Depth", curdepth, bar + diff + text + zero, 300, blue, lred)
```

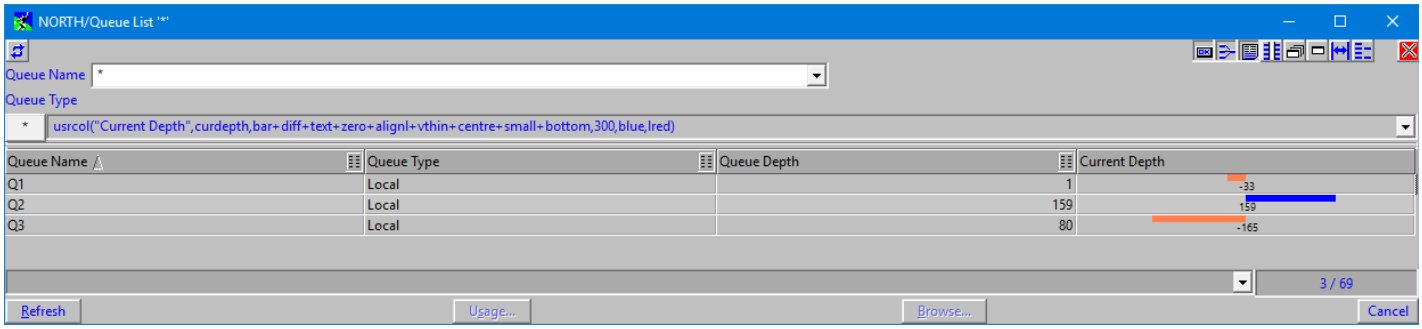


This clearly makes it much easier to see whether the number is positive or negative. Of course you might choose to put the colours the other way around. After all, a queue getting more empty is a good thing....and doesn't really warrant a red line!

There are a number of other formatting options we have. One of the problems with displaying the number in text is that the bars can obscure the number. Of course you could choose a maximum value which is unlikely to be reached but another method is to change the size of the line and text.

```
usrcol("Current Depth",curdepth,bar+diff+text+zero+alignl+vthin+centre+small+bottom,300,blue,lred)
```

At first glance this seems like a mouthful; we've added five more options! So, what does it do to our display ?



As you can see it moves things around a little bit. Let's explain each term in turn. **alignl** requests than the chart is drawn to the top or left of the cell. **vthin** says that the chart line should be drawn 'very thin'. **centre** says that text should be drawn in the cell of the cell (not left or right). **small** says that the text should be drawn in a small font. And lastly **bottom** says that the text should be drawn at the bottom of the cell. So it might seem a little daunting at first but each option is pretty easy to understand. The best way of seeing their effects is to just play with them and come up with a combination that you are happy with.

Needless to say there are further options that you can use. The full list of options you can specify is as follows:

alignl	Align the chart element to the left (if a bar) or the top (if a column)
alignr	Align the chart element to the right (if a bar) or the bottom (if a column)
avgl	Long term average. This option should be used where a long term average rather than the current value is needed. The equation used to calculate value is newval = 0.02*currentval + 0.98*oldval[11]
avgs	Short term average. This option should be used where a short term average rather than the current value is needed. The equation used to calculate value is newval = 0.1*currentval + 0.9*oldval[11]
bar	Display the value as a horizontal bar
bgtext	The background colour applies just to the text
bottom	Draw the text at the bottom of the cell
bytes	An option indicating that the value is in bytes. If the user has Units display active then the value will be displayed accordingly.
centre	Draw the text in the centre of the cell
col	Display the value as a vertical column
diff	Display the different between the current value and the last value
left	Draw the text on the left of the cell
ms	An option indicating that the value is in milliseconds. If the user has Units display active then the value will be displayed accordingly.
narrow	Display the bar or column slightly narrow of the cell
opaque	Draw the text with the background colour
percent	The text value displayed should be displayed as a percentage of the maximum
rate	This option indicates that the value should be considered a 'rate'. Often, but not always, used in conjunction with 'diff'. For example, consider the case where we wish to have a column showing how many bytes a channel is moving. We can have a user column that is the **diff+rate** of the bytes sent. MO71 will divide the value passed by the number of seconds since the last refresh thereby giving you a transfer rate.
reversed	Draw the column and bar charts in reversed mode. bar – bar drawn from the cell right edge col – columns drawn from top and bottom edges rather than centre
right	Draw the text of the right of the cell
secs	An option indicating that the value is in seconds. If the user has Units display active then the value will be displayed accordingly.
small	Draw the text using a smaller font
text	Draw the text value
thin	Display the bar or column thinner than the cell. Thin is thinner than narrow.
top	Draw the text at the top of the cell
vthin	Display the bar or column as a very thin chart
zero	Display the bar or column so that the zero point is in the centre
zeroline	Draw a line at the bar or column zero point
w10	Set minimum column width to 10 character wide
…	Constants **w10, w20** , **w30**, **w40** and so on up to **w150** . Each one increasing the minimum width of the column by 10 characters. For 'bar' type displays the column width automatically expands to fill the space available.
w150	Set minimum column width to 150 characters wide

[11]There is no guarantee that this equation will remain for all time. It is included to help the user understand the processing.

Remember when you are using these that each option should only be used once. This is especially true if you are using the addition operator (+). If you do issue a filter such as **usrcol("Current Depth",curdepth,bar+text+text)** and accidentally specify the same parameter more than once then the results are unpredictable but we can be fairly certain it won't be what you wanted!

If you regularly use the same set of options then it may be worth defining them in a suitable variable. For example, consider the filter example:

```
@opts := bar+diff+text+zero+alignl+vthin+centre+small+bottom;
usrcol("Current Depth",curdepth,@opts,300,blue,lred)
```

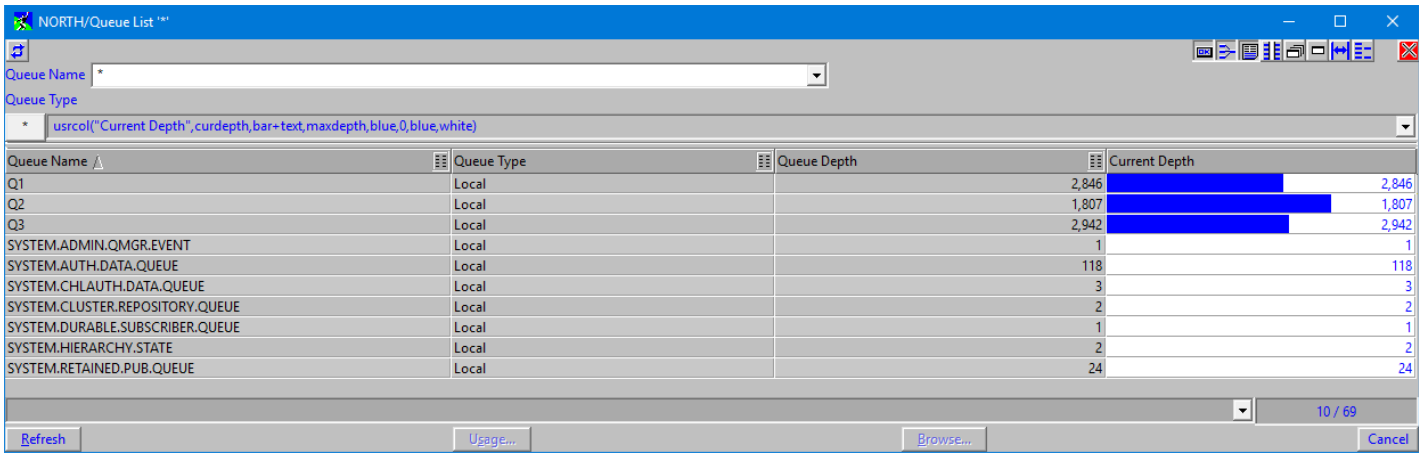
By defining a variable with the option you need then the same options can easily be passed to multiple usrcol() function calls. It also makes it far easier to change if you wish to change some aspect of the display.

We are almost done with the function definition now, however, we're not quite finished. We still haven't quite explained what the last two parameters do. Luckily they are fairly straight-forward. These are FgColour and BgColour. These last two parameters allow you to over-ride the foreground and background colours that are used for the user column cell. If the **bgtext** option is specified then the background colour only applies to the text portion of the display and not the whole cell. Like all the colour parameters the colour can be an RGB value or one of the many Colour constants such as blue, green, yellow etc. Please see Colours on page 272 for the full list.

For example the filter:

```
usrcol("Current Depth",curdepth,bar+text,maxdepth,blue,0,blue,white)
```

would give a display something like the following:

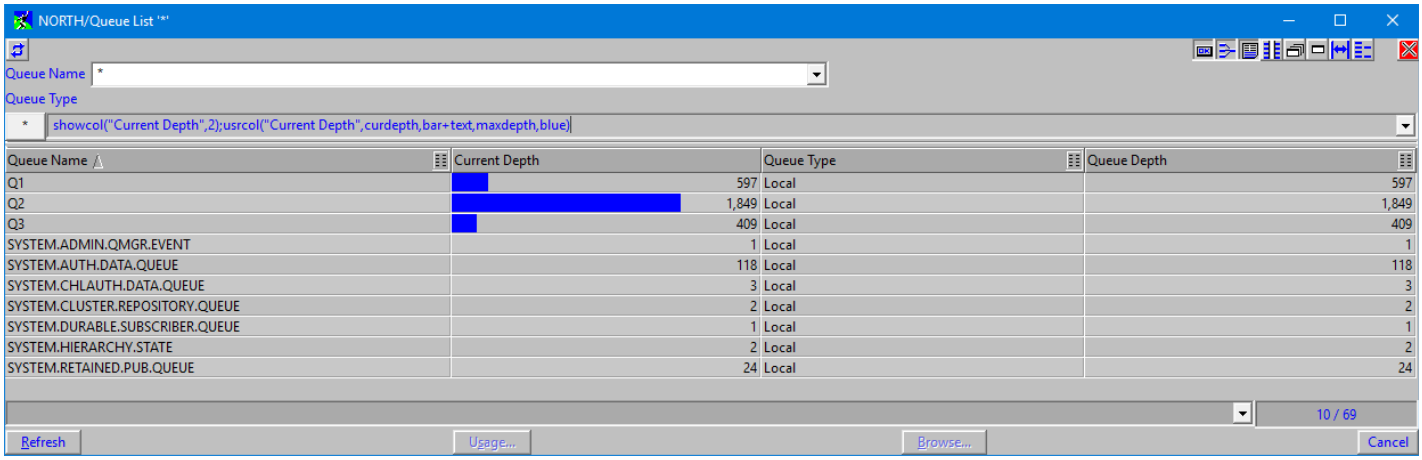


5.2.6.2. Positioning your user column

By default any user columns are added to the end of the display. However, it is easy to move the column to any other position if you like. This is done using the showcol() function. Let's use a usrcol() filter we used earlier and move it to the left.

```
showcol("Current Depth",2);usrcol("Current Depth",curdepth,bar+text,maxdepth,blue)
```

This might yield the following display. Note that the user column has now moved to the second position because that is what we specified on the showcol() function.



5.2.6.3. Using usrcol() without displaying the column

The user column function can be useful even if you decide not to display the actual column. The column is only displayed if **bar**, **col**, or **text** options are used. However, if none of these are used the function will still return the value that it always did do. Consider the following simple filter:

```
usrcol("Current Depth", curdepth, diff)
```

This simple filter will return a non-zero value if the value of current depth changes. In other words if it is used as the filter then the Queue List will only display queues that a 'active'. Or you might choose just to use it to colour the line with something like this:

```
bg(usrcol("Current Depth", curdepth, diff),lgreen)
```

This simple filter will just colour the line light green of any queue which is active but all queues will be displayed in the list.

5.2.6.4. Using usrcol() in practice

Clearly the usrcol() function is very flexible and has many different ways it can be used. However, with that flexibility comes extra complexity. It is not really feasible to type these filters in on the filter line each time you want a user column. Not unless you are very good at typing and remembering syntax anyway! However, there are two solutions to the problem. The first is to use the Filter Manager described on page 52. This allows you to define any filter you like and give it a nice simple name. The second method is to use Predefined Dialogs (described on page168) which allow you to go even further and define a type of dialog display and give that a name. For example, you could have a defined dialog called 'Active Queues' which uses usrcol() in one of the filters we've seen above to display only the queues which are currently being used.

Further examples on the use of the usrcol() function on our blog here.

5.2.7. Output Format String

Some functions, such as **fprintf** and **statusf**, allow data to be output in a formatted string. The format string is modelled closely on the 'C' language printf format string.

%[flags][width][.precision][type]

where :-

flags

- **-** Left aligned
- + Prefixes the output a + or – sign
- **0** Prefixes the output number with '0'
- # For **o,x**, or **X** fields with prefix with 0,0x or 0X respectively.
  For **g** or **G** fields it forces the output to contain a decimal point.

width

An optional positive integer specifying the minimum number of characters printed for this field.

precision

An optional positive integer specifying the number of decimal places, the number of significant places or the number of characters to be printed depending on the data type.

type

- **d,i** Signed integer
- **o** Octal integer
- **u** Unsigned integer
- **x** Hexadecimal integer using lowercase characters
- **X** Hexadecimal integer using uppercase characters
- **e** Real number in the format [-]d.dddd e [- | +]ddd
- **E** Real number in the format [-]d.dddd E [- | +]ddd
- **f** Real number in the format [-]dddd.dddd
- **g** Real number in either **e** or **f** format depending on which is smaller
- **G** Real number in either **E** or **f** format depending on which is smaller
- **S** String

Examples :-

Specification	Output
statusf("%d" , 2)	2
statusf("%05d" , 2)	00002
statusf("%.7s","This is a test")	This is
statusf("%s = %d","Value",7)	Value = 7
statusf("%s = %d","Value")	Value = %d *Note that there aren't enough parameters to fill the inserts*

5.3. Filter Manager

There may be a number of filters which are either used frequently or that are fairly complicated. As you will see in the examples that follow some filters can be many lines long. For these filters it may be more convenient to define the filter globally and assign a name to it. The name should not contain blank characters. To access the filter manager use the *'Action-Filter...'* main menu item or by pressing **<Ctrl-F>**. This will display the following dialog.

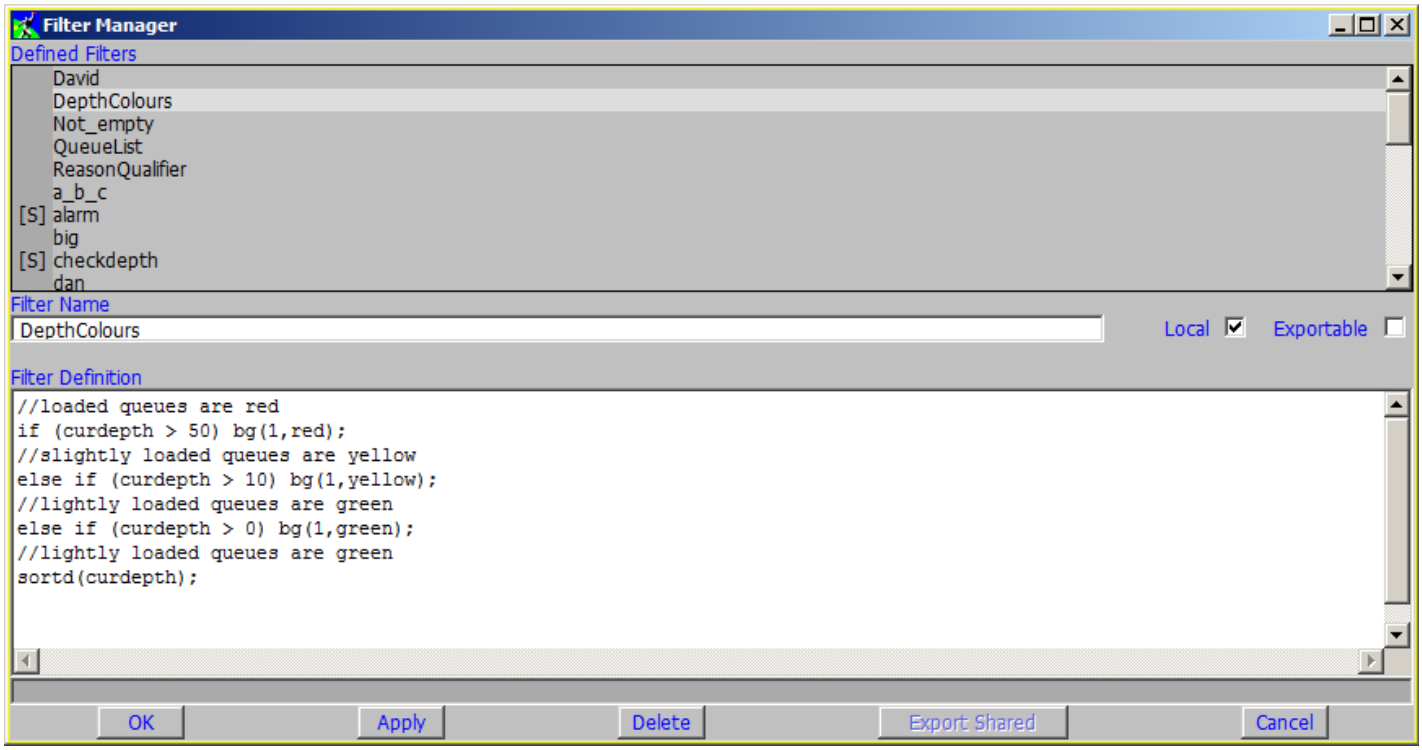


Figure 9: Filter manager Dialog

In this example a filter has been defined called 'DepthColours' which will set the background colour of a queue in a queue list window to a colour representing its current depth. To use this filter a filter of '$DepthColours' should be specified in a queue list window. Some filters can be used with all list types but since the example shown uses the variable 'curdepth' it should only be used on queue list displays. If this filter were used in say a Channel list display and error message saying 'Unrecognised keyword 'curdepth'' would be displayed.

As well as a name and a definition a filter has two other attributes, Local and Exportable. A local definition is one which is defined locally to this instance of MO71, it's definition lives in the MQMON.CFG file. If a definition is not local then it is 'shared'. This means that it's definition was read in from the shared filter file. Please see Shared Filters below for a description of shared filters. Only 'Local' filter definitions can be created or altered. The other attribute, Exportable, indicates whether this filter should be exported to the shared filter file when the 'Export Shared' button is pressed.

Pressing 'OK' or 'Apply' will add or update the currently displayed filter and cause any dialogs using this filter to update their display.

The Delete button will delete the filter currently selected in the list. Before deletion however the user is shown a confirmation dialog.

5.3.1. Comments

As you can see from the example above you can comment your filters by starting with the characters //. This marks the whole line as a comment which allows you to annotate your filter.

5.3.2. Shared Filters

Sometimes it can be useful to be able to generate a set of filters and have those filters used by other machines using MO71. These are known as shared filters. The concept is that you generally have one 'master' MO71 which exports the shared filters and you have many 'reading' instances of MO71 which just import the filters.

Whether an instance of MO71 is a 'master' or not is controlled by the 'Import shared filters' option in the 'Lists' tab of the Preferences Dialog. If this option is checked then MO71 will the import filters it finds in the shared filter file which is also configured in Preferences Dialog.

If the 'Import shared filters' option is not checked then MO71 can export certain shared filters to the shared filter file.

To generate shared filters

To create a set of shared filters you do the following actions.

1. Ensure that 'Import shared filters' in the 'Lists' tab of the Preferences Dialog is not checked
2. Fill in the name of the 'Shared Filter File' you wish to use in the 'Lists' tab of the Preferences Dialog.
3. Bring up the Filter Manager
4. Define or Alter your filters as required
   If you wish a filter to be shared then ensure the 'Exportable' checkbox is checked and press the 'Apply' button.
5. The list of filter names will show shared filters with an [S] in the left most column
6. Once the list of filters is correct press the 'Export Shared' button.
   This will generate the shared filter file and report how many filters were exported[12].

To use shared filters

To use a shared filter file you first need access to file containing the shared filters. This could be exactly the same file as was generated or the file could be copied locally. Similarly the file could be made available on a shared file server.

The steps you should take are:

1. Ensure that 'Import shared filters' is checked in the 'Lists' tab of the Preferences Dialog.
2. Fill in the name of the 'Shared Filter File' you wish to use in the 'Lists' tab of the Preferences Dialog.
3. Restart MO71
   Now, every time MO71 starts the shared filters file will be read ensuring that MO71 has the latest set of definitions.

[12]Ensure that the shared filter file value is correct since this action will overwrite the contents of the file.

5.4. Filter Examples

On the basis that the best way to see what effect filters have is the try them, here are a number of examples (some of use and others purely for demonstration purposes). Note that the following filters are examples of filtering that can be done on the Queue List. Similar filters can be made on any of the other lists.

- **queue = "Q1"**
  Show only queue "Q1"
- **queue == "Q1*"**
  Show only queues with a name beginning "Q1"
- **queue == "*Q1"**
  Show only queues with a name ending "Q1"
- **queue == "*Q1*"**
  Show only queues with "Q1" in their name
- **queue == "????"**
  Show only queues with 4 character names
- **queue = 4**
  Show only queues with 4 character names
- **queue < 8**
  Show only queues with less than 8 character names
- **cur**
  Show only queues which have messages on them
- **cur > 100**
  Show only queues which have more than 100 messages on them
- **ipprocs# | opprocs#**
  Show only queues which are currently in-use.
  The hash characters force the relevant columns to be displayed
- **bg(usage#=xmitq,red)**
  Show all queues but highlight the transmission queues in red.
  The hash character will force the usage column to be displayed
- **fg(qtype=local,red) & fg(qtype=remote,blue)**
  Show all queues but highlight local and remote queue types in red and blue respectively.
- **(qtype=local) | (qtype=remote)**
  Show only local and remote queues
  Note that quotes are **not** used.
- **qtype==”*o*”**
  Show queues of a type which contains a ‘o’ character such as local, remote and model.
- **!initq**
  Show queues which do not have an initiation queue defined.
- **trigctl#**
  Show local queues that have trigger enabled
  The hash will force the trigger control column to be displayed
- **!trigctl#**
  Show local queues that have trigger disabled
  The hash will force the trigger control column to be displayed
- **!trigctl# | !initq# | (!process# & usage# != xmitq) | !trigtype#**
  Show all the queues which have triggering disabled for whatever reason, and display any of the fields provided they have a value.

- **sortd(queue)**
  Sort the entries based on the Queue name in descending order

- **sortd(+queue)**
  Sort the entries based on the length of the Queue name in descending order

- **alarm(cur>100)**
  Switch on alarm if any queue has a depth greater than 100

- **beep(!desc)**
  Beep if there's a queue without a description

- **system(curdepth>maxdepth*0.8,"q -oQ1 -M\"%o is getting full\"")**
  If the queue is more than 80% full then issue the given command. The command given above is actually a call to my Q program (WebSphere MQ SupportPac MA01) which sends a text message to a queue. This demonstrates, in a simple way, how you can monitor the attributes of your objects and have commands issued should certain conditions arise.

- **@time := mqtime(crdate,crtime); @$d# := day$(weekday(@time),2);true**
  On a queue list dialog display a new column showing the day of the week the queue was created.

- **Retention Interval**
  When you define a queue you can specify a value for retention interval which is how long the queue definition is required. The Queue Manager doesn't do anything with this value however the users may wish to delete the queue after this period. So, how can we display the queues which have expired. Well, one way, is to use the following filter.

  @qtime:=mqtime(crdate,crtime); @age#:=(_time - @qtime)/3600; @age# >= retintvl

  This filter will only show 'expired' queues and will display a column which will show the age, in hours, of all the queues.

- **bg(qtype=local,yellow) & fg(usage#=xmitq,red) & !(queue=='SYSTEM*') & sort(queue)**
  This filter shows easy it is to change the colours of the various lines. This filter uses the *bg()* function to set the background colour of all local queues to yellow. It also shows all transmission queues with a foreground colour of red and ensures that the usage field is visible. It also removes all system queues and sorts by queue name.

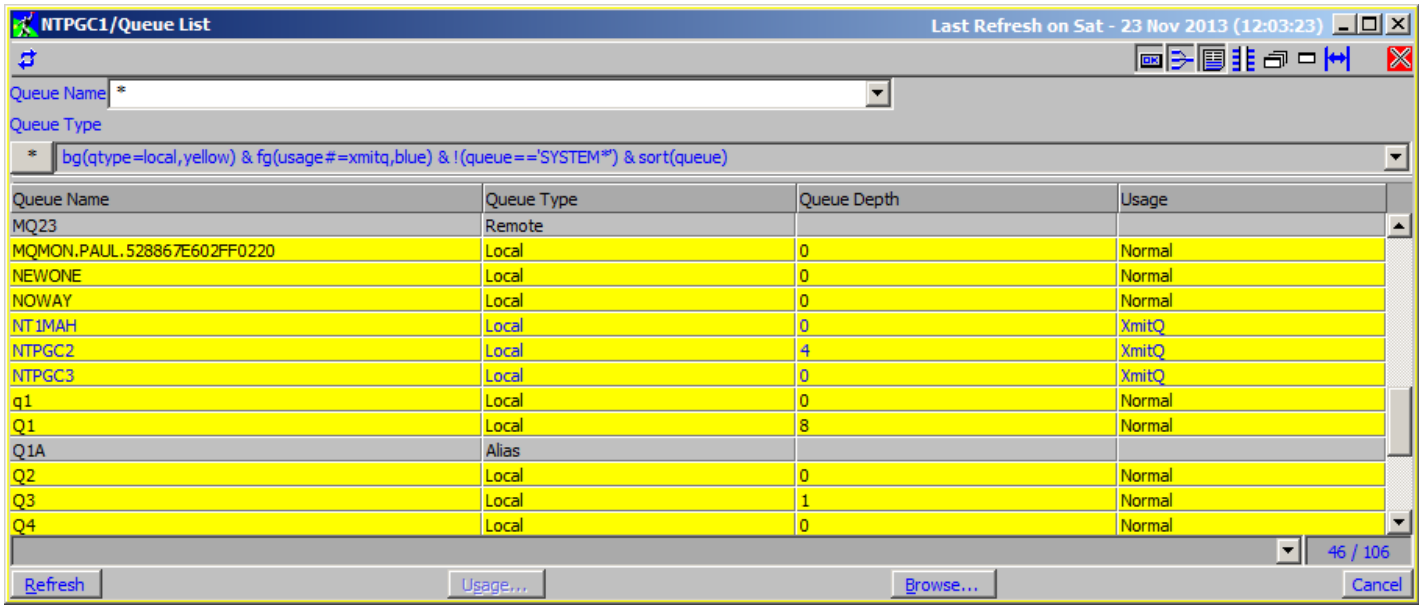


Figure 10: Queue List with colour filter

- **curdepth & bgcell(curdepth > maxdepth*0.8,red,curdepth)**
  This filter demonstrates just how easy it is just to change the colour of one cell in the row. In this case were setting the cell background to red if the depth of the queue is greater than 80% of it's maximum depth. The filter only shows messages which currently contain messages.

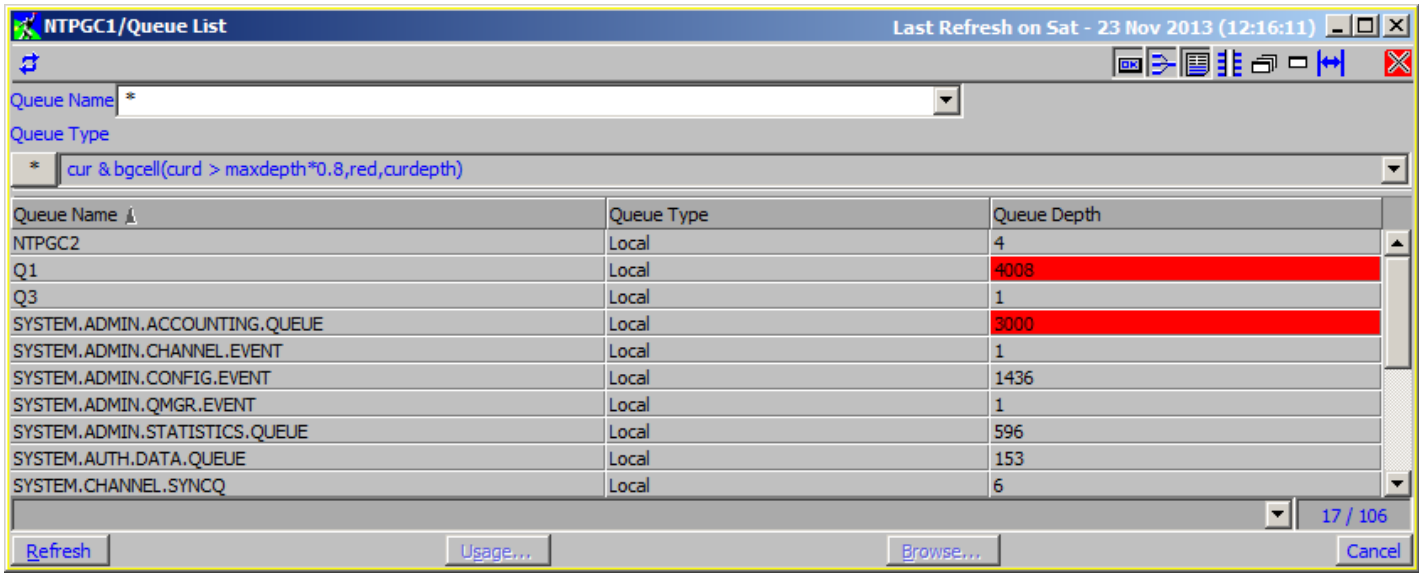


Figure 11: Queue List with cell colour filter

- **Depth History**
  To demonstrate the use of a user variable object insert, the following example will change the colour of a queue list entry depending on the depth of the queue. However, what makes this example different is that the depth of the queue is only considered significant if depth for this queue has been non-empty for multiple refreshes of the filter.

```
if (cur) @depth%o := @depth%o + 1;
   else  @depth%o := 0;
if (@depth%o)
{
  if (@depth%o > 3) bg(1,red);
                else bg(1,yellow);
}
else
{
  bg(1,green)
}
sortd(@depth%o);
```

- **Total number of messages on all queues**
  This example adds up the cur (or curdepth) value for each queue in a queue list. It then displays the total on the status bar after processing the last entry.

```
if (_first)
{
  @total := 0;
  _first := false
}
@total := @total + cur;
if (_last) status("Total messages = ",@total);
true;
```

- **Fullest Queue**

  The following example shows how you can collate information about all the list items and then display the result. This simple filter calculates which queue is fullest and displays it to the status bar.

```
if (_first)
{
  _first:=false;
  @$name := "";
  @depth := 0;
}
if (curdepth > @depth)
{
  @depth := curdepth;
  @$name := queue;
}

if (_last)
{
  if (@depth) status("Fullest queue is ",@$name," with ",@depth," messages.");
        else status("All queues are empty");
}
1;
```

- **Top two depths**

  This example shows the ability of requesting a rescan. This example uses scan 0 to find out what the top two current depths of a queue are (an expansion on the fullest queue example). A rescan is then requested and in scan 1 this data is used to show only the queues which have greater or equal to the top two depths of all queues.

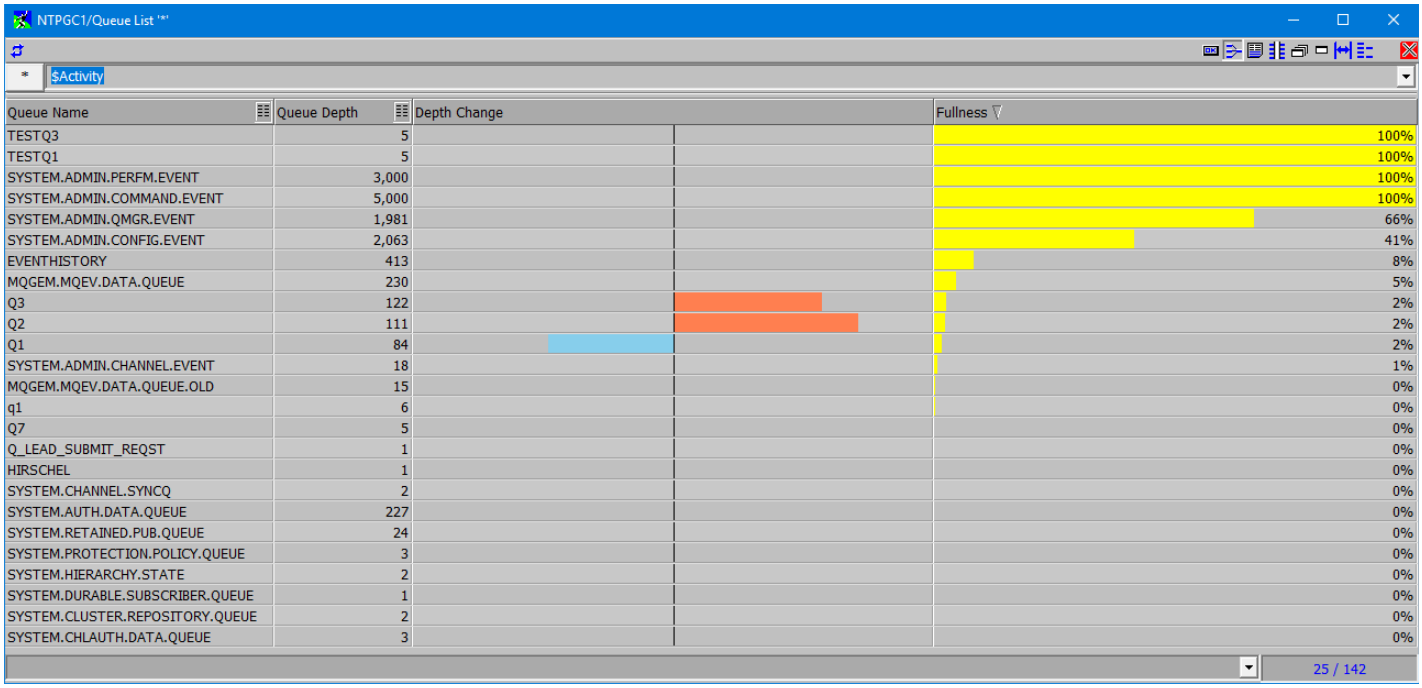
```
if (_first)
{
  if (!_scan)
  {
    @depth1 := 0;
    @depth2 := 0;
    _rescan := 1;
  }
  _first := 0;
}
if (_scan)
{
  (@depth1 & (cur >= @depth1)) |
  (@depth2 & (cur >= @depth2))
}
else
{
  if (cur > @depth2)
  {
    @depth1 := @depth2;
    @depth2 := cur;
  }
  else
  {
    if (cur > @depth1) @depth1 := cur;
  }
}
```

- **Queue Activity**

  The following filter uses the usrcol() function to display how full a queue is and how it's depth is changing.

```
hidecol(all);
showcol(queue,1);
showcol(curdepth,2);
usrcol("Depth Change",curdepth,bar+diff+zero+zeroline,100,lred,lblue);
usrcol("Fullness",curdepth,bar+text+percent,maxdepth,yellow);
sortd("Fullness");
cur
```

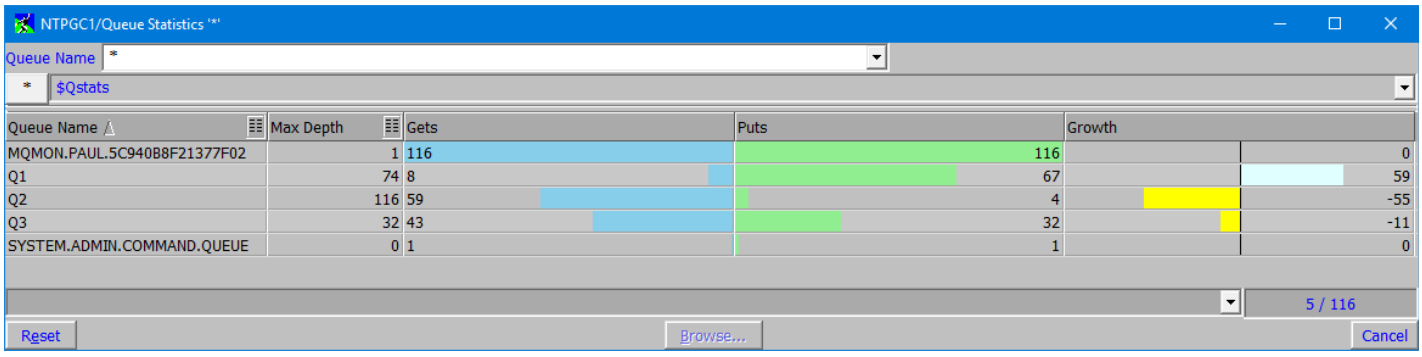
The filter above will generate a display something like the following:



- **Reset Queue Stats**
  The following filter uses the usrcol() function to display the number of puts and gets from the Reset Qstats command as a chart.

```
hidecol(all);
showcol(qstats,1);
showcol(hiqdepth,2);
@maxvalue := 100;
@options  := bar+text;
usrcol("Gets",msgsout,@options+left +reversed,@maxvalue,lblue);
usrcol("Puts",msgsin ,@options+right         ,@maxvalue,lgreen);
usrcol("Growth",msgsin - msgsout,@options+zero+zeroline,@maxvalue,lcyan,yellow);
msgsin + msgsout;
```

The filter above will generate a display something like the following:



Further filter examples can be seen on our blog here.

5.5. Filter Troubleshooting

There are a number of common mistakes, particularly with the more complicated filters, which can often lead to unexpected results when using filters. Check your filter against the list below if you see odd behaviour.

- **Did you remember to return a Boolean ?**
  Remember that as well as processing some data the filter must also return a Boolean value indicating whether the entry should be shown or not. In a number of cases above this is achieved simply by adding the characters **;true** or **;1** to the end of the filter. This ensures that the filter always returns true and therefore the entry will always be displayed.

- **Are you using files ?**
  Remember that a '\' character must be represented by '\\' in a file name.

- **Remember to use := not = for assignment**
  If you're using user variables then you must use := to assign a value. Both x := 3 and x = 3 are syntactically correct but the first is assigning the value 3 to x and the second is comparing the value of x with 3. Make sure you use the right one.

- **A displayed user variable and one that isn't aren't the same variable**
  The act of displaying a variable actually changes where the variable is stored which therefore changes the value. In other words @x# (which is displayed on the screen) is not the same variable as @x. You can prove this with simple test filters

  - **@x# := 5; @x := 2 ; 1**
    This filter will generate a column for **x** all containing 5

  - **@x# := @x# + 1; 1**
    This filter shows a column of all 1's. Why do we not see the number increasing for each entry ? Well, the displayed field is set to zero initially for each entry.

    If we wanted to see an increasing number we'd have to use the following filter.

  - **@x := @x +1; @x# := @x; 1**
    Here we see different numbers and they seem to be increasing but there are not in order. The reason for this is that the order displayed in the dialog is not necessarily the order that the objects are returned from the command server.

- **Remember to switch _first off**
  It is very common to want to do some initialization in your filter like reset counters to zero. To do this you put it inside an **if (_first)** block. However, remember that you yourself must set **_first** to false. If you don't your counters will be reset to zero for every entry in the list.

- **Remember operator precedence**
  For example, suppose you wish to see the queues that don't match the pattern "*OBJ*". At first glance you might think is was **! queue == "*OBJ*"**. However, because the **!** operator is higher in precedence than == it will be executed first which will give a completely different result. The solution is to use brackets. The expression **! (queue == "*OBJ*")** will work fine.

- **Do you see the text "* Error *" ?**
  Remember that not every expression is valid nor does every object contain the same attributes. Consider the very simple filter **@@Fullness# := 100 * curdepth/maxdepth;1** This very simple filter essentially asks for a new column to be displayed which shows the 'fullness' of the queue. ie. A percentage of how close to the maxdepth the queue is. This works fine for Local queues. However, the problem with this filter is that some other queue types don't have **curdepth** or **maxdepth** attributes. As a consequence MO71 will display * Error *; effectively stating that it could not calculate the expression value for this object. You can also get similar effects with issues like dividing by zero. Remember that you can use **if** statements and functions such as **exists()** to ensure that variables used in expressions are available.

Chapter 6. Queue Manipulation

To manipulate messages MO71 needs direct access to the queue since no support is provided by the command server. It is possible to browse messages on other queue managers but you need a program on the remote system that understands the PCF messages generated by MO71. MO71 itself clearly does understand the flows so it is quite possible to have, for example, two NT machines each running MO71 browsing each others queues[13].

There are three ways of displaying the contents of a Queue.

6.1. Browse Queue

This dialog will show the list of messages on the specified Queue.

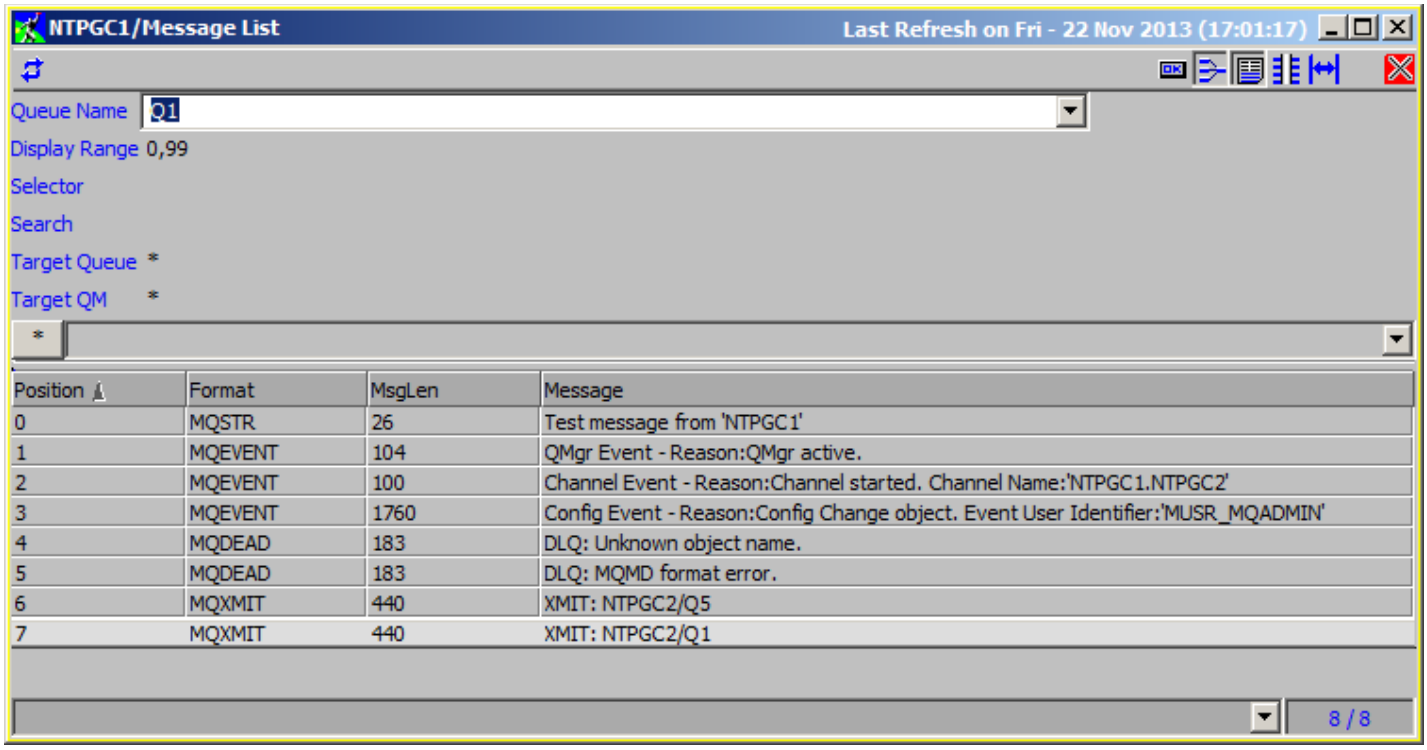


Figure 12: Message List Dialog

This dialog shows a summary of the messages contained in the queue. The first few bytes of the message are displayed or, if the message is a recognised format, a summary of the message is shown. As can be seen from the screen shot above MO71 understands formats such as Dead Letter Queues, Transmission Queues and PCF messages. There are various preference settings which can control the exact behaviour of this dialog please see the 'Browse' tab of the “Preferences Dialog” on page 193 for more details.

Various message operations can be performed from this dialog. For example, messages can be moved or copied to other queues or the messages can be deleted.

6.1.1. Message Selection

By default MO71 uses a Message Token to identify each message. This is a unique token, assigned by the Queue Manager, to each message. The advantage of Message Tokens is that they are unique and fast to retrieve messages. The downside is that Message Tokens do not span cycles of the Queue Manager. If your Queue Manager ends during the life of the dialog it may be necessary to refresh the dialog once it comes back up.

If your Queue Manager is quite old, such as pre-MQ V7.0 then MO71 will use the queue position and/or Message Id to identify a particular message. The problem here, of course, is that neither are absolutely guaranteed to be unique. You can use the ‘Message Selection’ menu to select how the application matches the selected messages against the messages on the queue. The problem is that

[13]I do not provide a description of the messages but they are fairly easy to work out. It should not be too difficult to write an application on your target system which reads the queue and generates the correct responses. For simplicity, however, I suggest that most people stick to browsing queues only on the Queue Manager the application is connected to directly or via a client connection.

IBM MQ does not have to assign a unique Message Id to each message rather it is under application control. Therefore when selecting a message to move the Message Id may not be sufficient, checking the position on the queue and the Message Id is much safer. Of course, even this is still not guaranteed to identify the same message.

Another option in the 'Message Selection' menu is 'Apply to all messages'. When this item is checked then whatever operation is selected will apply to all messages. The button and menu text is changed to make it obvious. So, for example, if you wished to copy all messages from one queue to another you would select 'Apply to all messages' and then press the 'Copy All'. Similarly if you wished to delete all messages, but couldn't use CLEAR QUEUE because the queue was in use, then you could select 'Apply to all messages' and then press the 'Delete All' button.

6.2. Browse Message

This dialog will show the contents of the message formatted into a human readable form. If some characters do not display as expected, it may be necessary to adjust the locale setting. For more information see “Appendix E: Display National Language Characters" on page 276.

The application understands the standard IBM MQ formats and will format out the content of the messages. For example it allows the user to look at messages on Dead Letter Queues, Event Queues, Transmission Queues and Command Queues.

This is the most complex dialog for queue browsing. There are a number of options on the Pop-up menu which let you control what data is displayed.

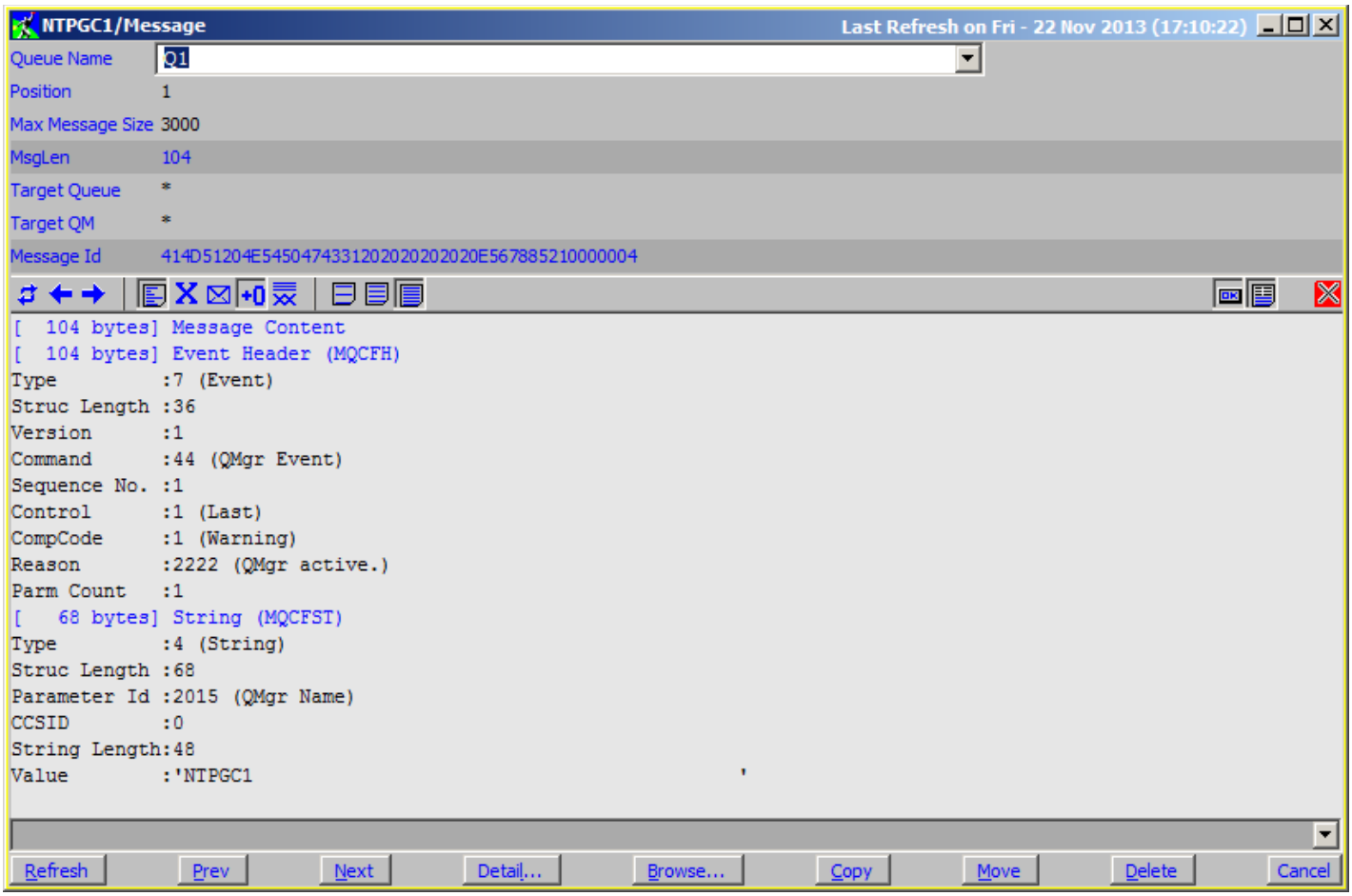


Figure 13: Message Dialog

This dialog allows many different options to control how the message is displayed. The message descriptor can be displayed, the message can be formatted or shown in hex. The message can be converted to local encoding and the detail level can be set.

The content of the queue can be simply traversed using the previous and next buttons (or tools in the toolbar). As in the previous dialog various message operations such as copying, moving or deleting messages can be performed.

6.3. Message Detail

This dialog will show the message descriptor fields and the first few bytes of the message for a particular message on a queue. The dialog options control whether the message is shown just as bytes or a summary of the message as shown in the following screen shot.

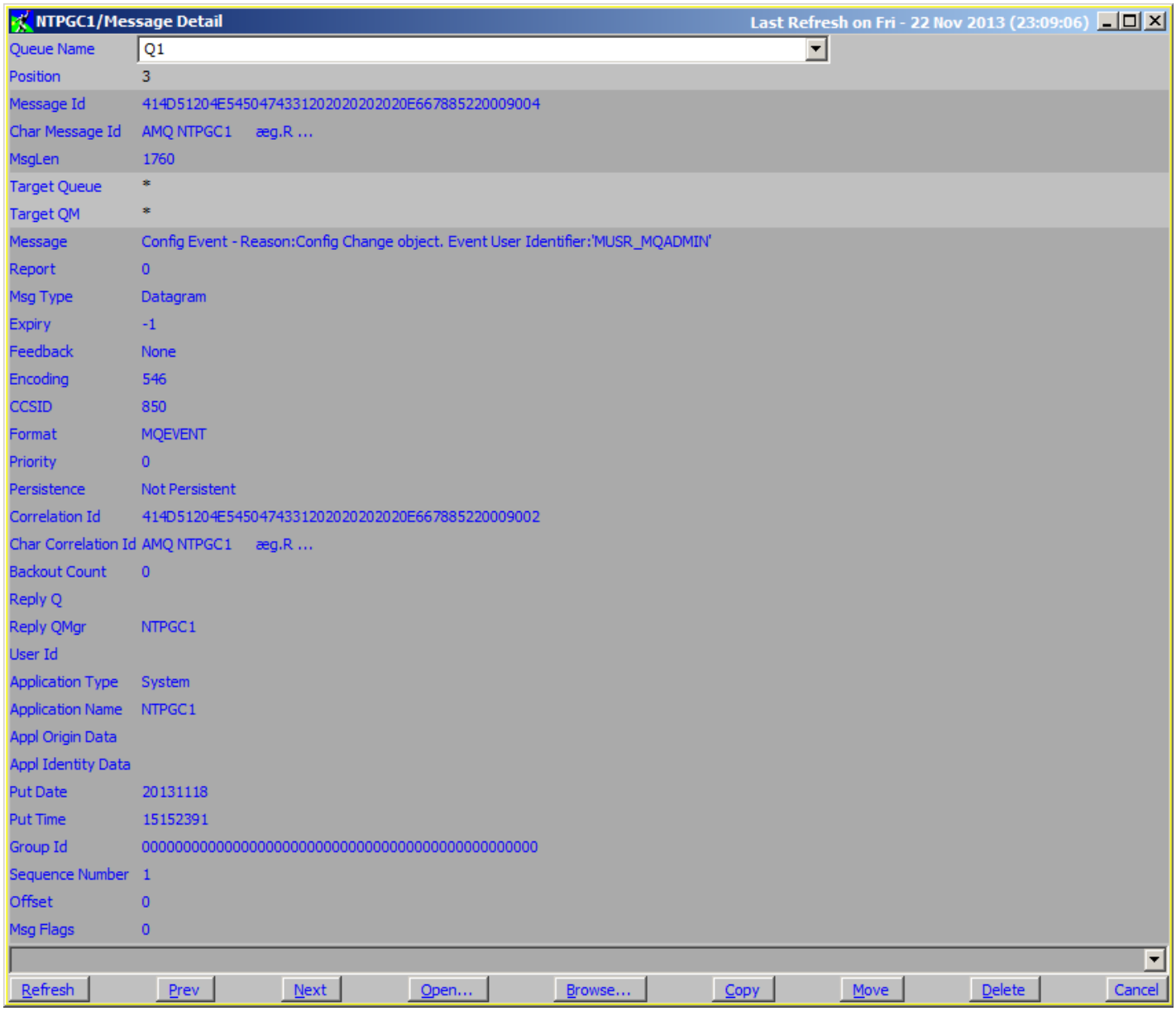


Figure 14: Message Detail

Identification of a message is controlled by either or both of its position and its Message Id. The position is purely an index of the message when the queue was browsed last. If messages are being added and removed from the queue while you're browsing the queue then the same message may well be given a different position number the next time you refresh the display.

6.4. Fields

The fields at the top of the dialogs allow control over the messages displayed and also controls when targeting a message to another queue.

6.4.1. Display Range

When a list of messages is requested you have the option to specify the range of messages you're interested in. By default the range is '0,99', which means the first 100 messages. This prevents accidentally displaying thousands of messages if the queue contains more messages than you expected !

However, if you find that you frequently want to browse a different number of messages than the first 100 you can change this setting in the preferences dialog. See "Preferences Dialog" on page 193 for more details. MO71 will support browsing up to 30,000 messages in a single display. Again you can set this maximum in the "Preferences Dialog". If you really need to browse a message is not in the first 30,000 messages on a queue then you can use the display range to specify the starting point. However, a much more efficient way of doing it is to use the selector field below.

6.4.2. Selector

The selector[14] field allows the user to enter an SQL92 expression which is executed at the server. Descriptions of the format of the expression can be found in the Message Selector Syntax section of the IBM MQ documentation. Simple examples would be:

- **Root.MQMD.Persistence=1**
  Show only the persistent messages
- **Root.MQMD.UserIdentifier = 'clarkep '**
  Show only messages put by a particular user
- **Root.MQMD.MsgId="0x414D51204E54504743312020202020B51261522004F009"**
  Select a particular message
- **myprop > 10**
  Show only messages where the value of property 'myprop' has a value greater than 10.

Note that all field names in selectors are case sensitive.

6.4.3. Search

The search field allows the user to enter a search string for the messages. Only messages which contain this case sensitive search string will be displayed. Note that only the message itself, not the message descriptor is checked for the string. To limit the amount of data retrieved from the queue only the first 30Kb of a message is searched. If you must search the entire message then you can use the Queue load/unload facility described on page 68.

When a search string is used the display range fields control which of the matching messages are displayed. For example, 3,5 will display the 4th, 5th and 6th matching messages. Note that the '**Search Offset**' setting controls whether the portion of the message shown starts at the point where the search string is found or whether it is the start of the message. If the search offset is used the message summary is automatically inhibited since the bytes returned from the command server are no longer at the start of the message.

6.4.4. Max Message Size

When displaying a message only the first 10000 bytes are shown by default. This prevents very large messages causing network delays etc. However, if necessary this value can be increased to whatever size is required. If you frequently browse messages larger than 10000 bytes and find yourself constantly increasing this value then it might be worth increasing the default maximum browse message size in the preferences dialog. See "Preferences Dialog" on page 193 for more details.

If the message being browsed is larger than this maximum message size then clearly the message displayed will be truncated. This may cause error messages to be displayed at the end of the message reporting that the data is incomplete. A common type of message to have this problem is XML since XML messages are often fairly large and any truncation will truncate the end-tags making the message incomplete.

[14] Available on Queue Managers MQ V7.0 and later

6.4.5. Target Queue Manager

When copying or moving messages to another queue you can optionally identify the target queue manager for the message, by default it has the value '*' which has the following effect :-

If the message has a MQXQH or MQDLH header then the queue manager is that which is specified in the header provided the option 'Strip Headers' is selected.

If the message doesn't have a header then it is the local queue manager

6.4.6. Target Queue

When copying or moving messages to another queue you can optionally identify the target queue for the message, by default it has the value '*' which has the following effect :-

If the message has a MQXQH or MQDLH header then the queue is that which is specified in the header provided the option 'Strip Headers' is selected.

If the message doesn't have a header then this is an error and a message will be displayed asking for the target queue name to be identified.

6.5. Pop-up Menu

The operations that may be available are listed below. Some of the most common actions can also be found as buttons.

6.5.1. Next/Prev

Will skip to the next or previous message, if any, on the queue.

6.5.2. Display Format

This option will allow the user to control the amount and format of the data that is displayed.

- **Formatted Message**
  If selected the application will attempt to format the message into human readable text. Once it no longer recognises the format it will display the remainder in hex.
- **Hex Message**
  If selected the message will be printed in hex.
- **Message Descriptor**
  If selected the contents of the message descriptor are displayed before the message.
- **Display Offset**
  In a HEX display the offset of the bytes from the beginning of the data is displayed.
- **Display Linenumbers**
  When selected a column of line numbers will be displayed down the left hand side of the screen. You cannot select or copy the line numbers.
- **Ascii Column**
  In a HEX display this option controls whether a column of ASCII characters is displayed on the right of the display. This can be useful to make sense of all the HEX characters.
- **Auto Detect XML Message**
  When selected the application will examine the first few bytes of the message to see whether it contains XML tags. If it does the message will be formatted as though it were an XML message. If not selected only message with the MQRFH2 format will be treated as XML.
- **Display XML shortform**
  When selected, any XML message will be displayed in an abbreviated form where not all tags and tag characters are displayed to aid readability.
- **Display Hex as chars**
  When selected, the hex fields in the MQ headers will be displayed as character strings as well as in hex. This can be useful if the Message Id or Correlation Id, for example, are character strings.

6.5.3. Detail Level

Will allow the user to control how much detail is shown when structures are displayed.

- **Low** Only the major fields of the structures are shown
- **Medium** The majority of structure fields are shown
- **High** All structure fields are shown

6.5.4. Copy to clipboard

Will copy the selected portion of the message to the clipboard. Selecting text in the message display can be performed either using the mouse or keyboard cursor keys while holding the shift key. Ctrl-A will select the whole message. Ctrl-C will perform the clipboard copy operation.

6.5.5. Copy shown to clipboard

Will copy all the message text, as displayed, to the clipboard. Only the visible portion of the message is copied. To copy the entire window to the clipboard, the standard windows action is <Alt>+<PrtSc>. This will work for practically any window.

6.5.6. Copy all to clipboard

Will copy the entire message to the clipboard, including the data that is not currently visible.

6.5.7. Edit

This menu will invoke the MQGem product MQEdit and pass it the Queue Manager name, Queue name and Message Token of the message so that MQEdit open at the same message and allows it to be edited[15]. This option will be greyed out if MO71 does not detect a valid MQEdit licence. You can override this by checking the 'MQEdit Installed' option in the Browse tab of the Preferences Dialog.

6.5.8. Copy

Will copy the selected messages to the queue identified by the Target Queue and Target Queue Manager fields.

This option is not available if the context setting is ‘pass’.

6.5.9. Move

Will move the selected messages to the queue identified by the Target Queue and Target Queue Manager fields.

6.5.10. Delete

Will delete the selected messages from the queue.

6.5.11. Put Message

This will present a put message dialog which allows the user to put a simple message. For more information see “Put Message” on page 189.

6.5.12. Load/Unload Messages

This will present a dialog which will allow the user to load or unload messages between a queue or file or between two queues. For further information please see “Queue load/unload facility” on page 68.

6.5.13. New

This will present a blank dialog of a type suitable to define a new object of the type displayed in the list. Not all list types will have this option.

[15]Clearly in order to run the MQEdit product it must be installed and you need a valid licence. MQEdit V9.0.1, or later, is required for correct functioning of this feature.

6.5.14. Convert

By default browse operations will convert the message to the code page and encoding values of the workstation. The codepage used to convert to can be changed in the preferences dialog if the windows codepage is not suitable for the message being browsed. By the deselecting the *'Convert'* menu option messages will not be converted when they are browsed. Note that message copy, move and delete operations never convert the message.

6.5.15. Message Summary

This menu option controls whether the message is shown just as the first few bytes of the message or whether MO71 should try to summarize the contents of the message. MO71 understands MQ message formats such as event messages, PCF messages, dead letter queue headers and transmission queue headers. The summary will only be based on the first few bytes of the message. The number of bytes can be controlled, to some extent, by settings in the "Preferences Dialog".

6.5.16. Properties

This menu controls whether message properties are returned in a header at the front of the user message (see MQGMO_PROPERTIES_FORCE_MQRFH2). Currently this is the only way in which message properties can be viewed when browsing a queue.

6.5.17. Strip Headers

Messages on an IBM MQ queue may have a transmission queue or dead letter queue header pre-pended to the actual application message. By default it is assumed that these headers should be maintained if the message is moved or copied to another queue[16]. This helps to ensure that the full message is maintained. However, there are times when these headers should be stripped off, for example, when moving a message from a transmission queue or Dead Letter Queue to a new target.

6.5.18. Ignore CR/LF

This option controls whether carriage and line feed characters in the message data should be adhered to or not. By switching on ignore the data will flow in a single line.

6.5.19. Multi Transaction

Normally all copy, move and delete operations will be done under one transaction, this avoids the problem that a system crash or some sort of failure will leave the operation only partially completed. All the operations will either complete or the transaction will be backed out. There is a queue manager attribute, max uncommitted messages, which dictates the maximum number of messaging operations allowable by an application in a single transaction. If you are trying to move or copy more messages than this limit then clearly it cannot be done in a single transaction. In these cases you must select the *'Multi Transaction'* option and take the small risk that the operation may only partially complete.

6.5.20. Search Offset

When using a search string in a message list display this setting controls whether the 100 byte message portion that is displayed is from the start of the message or from where the search string is found within the message.

6.5.21. Message Selection

Operations such as Copy, Move and Delete will apply to the current message if only a single message is displayed or the selected messages if a list of messages is displayed.

A third option is to have the operation applied to all the messages on the queue. This is achieved by selecting *'Apply to all messages'* in the *'Message Selection'* menu. If this menu is selected then the action buttons will change to reflect the 'all messages' status. Because this is a potentially hazardous operation the 'All message' mode is removed as soon as any action is selected.

[16] Earlier versions of the program had this option on as default. However, having 'strip' off is regarded as a safer option.

6.5.22. Context

When copying or moving messages between queues the context of the message will also be copied or moved by default. However, there are other options :-

- **Set all (default)**
  The context information of the copied or moved messages will be same as the original. This does require a high level of authority on the target queue.
- **Pass**
  Will pass all the context from the source message to the target message. Pass context is only valid on a move operation not copy since pass context is not valid after a browse.
- **Default**
  Default context is used. The identity and origin context of the new messages will be that of the MO71 program itself.

Chapter 7. Copying, Moving and Deleting messages

MO71 provides two ways that you can manipulate messages on a queue. You can browse a queue and select an operation for the messages. Or you can use the Queue load/unload facility to copy/move messages between files and queues according to a criteria you can set. The browse queue method is generally for use when you just have a few messages that you want to transfer. The Queue load/unload method is for when you want to apply the operation to the whole queue or file. It also allows you to specify conditions such as age of message or the presence of a piece of text.

7.1. Browsing the queue

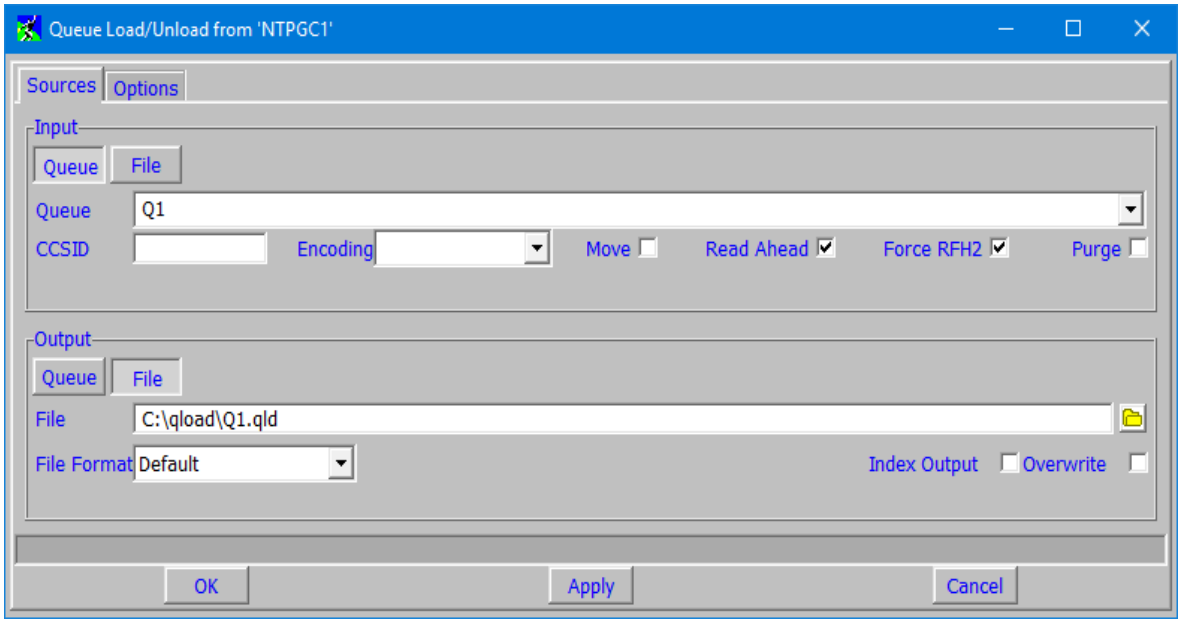
If you bring up a 'Message List' dialog you can select messages you wish to copy, move or delete. All you need to do in this case is select the messages you wish to operate on and then press the appropriate action. You can also, if you wish, apply the operation to all messages on the queue. If you wish to do this, then select the 'Message Selection – Apply to all messages' context menu option. This is all described in more detail in the previous section Chapter 6. Queue Manipulation.

Depending on how long the operation takes and the settings in MO71, a Progress Dialog may be displayed which will show you how far MO71 has got through the operation. For more information on the Progress Dialog please see 7.3. Progress Dialog on page 73.

7.2. Queue load/unload facility

From various places, such as the queue list and message browse dialogs a menu option is provided which will present the queue load/unload dialog. This allows the user to unload messages from a queue and put them in a file, copy messages from one queue to another, or load messages from a file to a queue. This can be useful for a variety of reasons such as backing up your queues, saving a set of messages for later re-use, unloading a queue so that you can do some editing of the message or message attributes or for sending someone else a copy of your messages.

The load/unload facility is essentially the same code as in our QLOAD command line product but presented in a window. As a consequence unloaded files can be shared between them[17].



The dialog is split in two halves. For both input and output a choice has to be made about whether to use a file or a queue. So, for example, if the user chose an input of a queue and an output of a file then this would move messages from a queue to a file. In other words, it would unload the queue to the file. Note that it will not destructively get the messages; rather a copy of the messages will be taken unless the options 'Move' or 'Purge' are selected.

Both input and output can be of the same type. If they are both queues then you can do a copy of messages from one queue to another. If both are files then it allows you to reformat the output of the file. If the 'Auto Update browse lists' preference option is selected then any dialogs browsing the relevant queues will be refreshed after the queue load/unload operation.

[17]Files can also be shared with the IBM MQ product command utility dmpmqmsg

The various options are:-

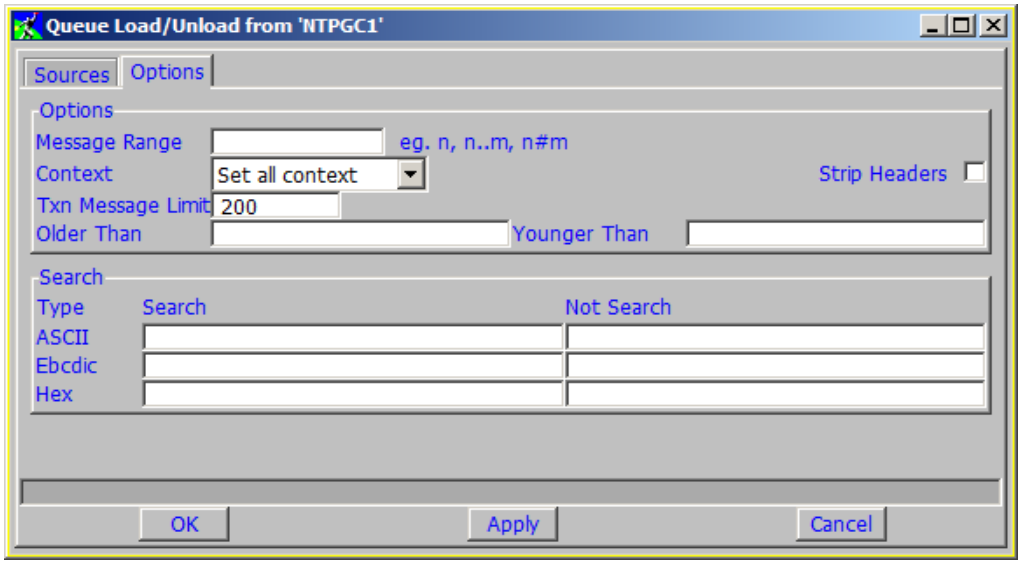
- **CCSID**
  If specified the messages will be ‘got’ from queue converted to the requested codepage.
- **Encoding**
  If specified the messages will be ‘got’ from the queue using the byte encoding requested.
- **Move**
  If this checkbox is checked then the messages will be removed from the source queue otherwise they will be copied.
- **Read Ahead**
  If this checkbox is checked then MO71 will use a read-ahead handle if it can to retrieve the messages. Read ahead offers a significant performance advantage over client connections, particularly if those client connections are over slow network links. Read Ahead will only be active if either no message filtering is applied or messages are being copied rather than moved.
- **Force RFH2**
  If this checkbox is checked then all message properties will be unloaded and captured in the output file as part as the message payload (in an RFH2 header).
- **Purge**
  This option is only in effect when the move option is specified and some form of filtering is in effect in the options tag. When selected it says that messages not matching the selection should be removed from the queue (and are therefore lost since they **will not** be written to the target). If this option is not selected then messages not matching the selection will remain on the source queue.
- **Async. Put**
  If this checkbox is checked then MO71 will use an Async response put handle to put messages to a target queue. Async. Put offers a significant performance advantage over client connections, particularly if those client connections are over slow network links.
- **File Format**
  The following file formats are available:-
  - **Default**
    The message format is just a hex string output. This is the most compact format but not too friendly to the human eye.

    ```
    X 54657374204D65737373616765
    ```
  - **ASCII**
    Message is presented in ASCII as much as possible. This is useful if the message contains a fair degree of ASCII which needs to be edited.

    ```
    S "Test Messsage"
    ```
  - **ASCII Column**
    Message is presented in hex but with a column of ASCII characters to the right-hand side. Any character in the message which cannot be represented by a printable ASCII character will be displayed as a ‘.’ (dot).

    ```
    X 54657374204D65737373616765                  <Test Messsage>
    ```
- **Index Output**
  When selected the produced file will contain comments which indicate which message index each message is. This is useful when reloading only part of a file to identify each message by its index value.
- **Overwrite**
  By default the application will prompt the user if the output file already exists. If this option is checked then any existing file will be overwritten without requiring user confirmation. On the other tab a number of options are available to control which message are loaded/unloaded and what message attributes should be taken.

The options tab allows the setting of various values to control which messages are moved/copied.



- **Message Range**
  This option allows only a certain number of the messages to be processed. For example
  - **5** Would only process message number 5
  - **5..8** Would only process messages 5 through 8, inclusive.
    To process all messages from message 5 onwards enter a string like **5..999999**
  - **5#8** Would only process 8 messages starting with number 5
  - **#8** Would process the first 8 messages.
- **Context**
  Context setting which control how the origin and identity context is transferred between the input and output source. The value of 'set all context' will ensure that all of the context fields are transferred and would therefore tend to be the most obvious choice. Note, however, that 'set all context' requires the most authority.
- **Strip Headers**
  This option controls whether headers, such as transmission queue and Dead Letter Queue headers should be stripped before the message is moved or copied.
- **Transaction Message Limit**
  If you are moving message from one queue to another then this must be done under a transaction to be totally reliable (in the case of persistent messages it is also much more efficient). However, there is a limit to how many operations you can perform in a single transaction so this value allows you to say how many messages form part of the transaction before the transaction is committed.
- **Older Than**
  This option allows selection based on the age of the message. The parameter can be entered free format using the keywords, **seconds, minutes, hours, days, weeks, years**. Only the initial parts of the keywords need be specified. For example, **1 hour 15 minutes** or **1 h 15 m** are equivalent.
- **Younger Than**
  This options allows selection based on the age of the message. The format of this parameter is the same as the **'Older Than'** field.
- **Search**
  You can choose to process only messages that contain a certain string. This string can be in ASCII, EBCDIC or hex. You can also do the opposite search - that is for messages that do not contain a certain string. You can use any combination of the search string options together. If more than one option is used, all search strings must match for the message to be processed.

Depending on how long the operation takes and the settings in MO71 a Progress Dialog may be displayed which will show you how far MO71 has got through the operation. For more information on the Progress Dialog please see 7.3. Progress Dialog on page 73.

7.2.1. File Format

The format of the file that MO71 uses is the same as QLOAD. The file is deliberately human-readable to allow a user to update the file to easily make changes to the messages before loading them onto a queue, perhaps on a different system.

The file has a free format. Spaces and blank lines are ignored unless between quotes. When generated the file may contain spaces to make the file more readable, but they do not affect the message format stored in that file.

The first column contains the key for each line. This can have a number of different values, some of which we have seen already. These are listed in “Table 1: Meaning of column one symbol in file format”.

Column 1 value	Meaning
S	The text shown on this line is ASCII string text
X	The text shown on this line is hex
A	The text shown on this line is an attributed in the Message Descriptor (MQMD)
*	The text on this line is a comment and will be ignored.

Table 1: Meaning of column one symbol in file format

The second column should contain a blank; this makes the file more readable.

If the first column contained the letter ‘A’, then the next three characters will represent the field that is being shown. For example, ‘FMT’ shows that this line displays the format of the message. The full set of these field labels is listed at the end of this section.

7.2.1.1. Example - Changing the user ID

The message descriptor (MQMD) contains a user ID which, depending on your security settings, may be used for access control when putting a message onto a queue. You may wish to change this user ID before reloading the messages onto a queue on a different system because the IDs on that system use a different naming convention. Before loading the queue from your file of messages, you would edit the file changing the field identified as USR to your desired user ID.

```
:
A RTM MQ24
A USR HUGHSON
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC30000000000000000000
:
```

7.2.1.2. Attribute Format Reference

The fields in the Message Descriptor (MQMD) are formatted in the file with a three character string representing the attribute name. “Table 2: Message descriptor attribute representations” lists the full set of these strings and which field they represent.

Message Descriptor attribute	File format representation
Report	RPT
MsgType	MST
Expiry	EXP
Feedback	FBD
Encoding	ENC
CodedCharSetId	CCS
Format	FMT
Priority	PRI
Persistence	PER
MsgId	MSI
CorrelId	COI
BackoutCount	BOC
ReplyToQ	RTQ
ReplyToQMgr	RTM
UserIdentifier	USR
AccountingToken	ACC
ApplIdentityData	AID
PutApplType	PAT
PutApplName	PAN
PutDate	PTD
PutTime	PTT
ApplOriginData	AOD

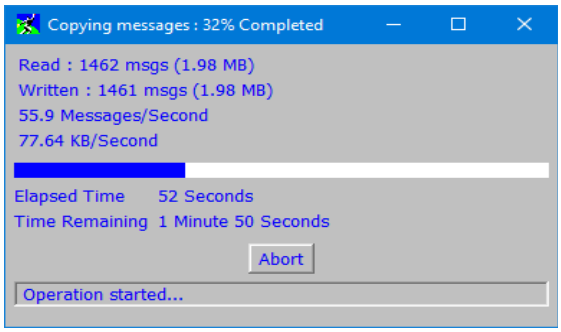
Table 2: Message descriptor attribute representations

7.2.1.3. Recognised file formats

MO71 recognises the file format described above but also recognises the file format used as output from the sample browse program AMQSBCG. In other words, you can take an AMQSBCG output file and pass it as an input file to the load/unload dialog.

7.3. Progress Dialog

Copying, Moving and Deleting large numbers of messages can take a considerable time. As such it is useful to get some feedback about how far along the operation has got. This is what progress dialogs are all about. By default the dialog is displayed if the operation takes longer than 3 seconds to complete. However, you can change this by setting the value of 'Progress Dialog Delay' in the Dialogs tab of the Preferences Dialog.



As you can see the progress dialog will show you how many messages have been read and written so far. It also calculates the average number of messages per second which are being processed. Note that the average message speed counts both the input and output messages. So, in this case we have processed 2,932 messages in 52 seconds – or around 56 messages/second. By the same token MO71 tells you how many bytes per second we are moving.

This is followed by a visual representation of how far through the process MO71 believes it is and an estimate on how long it will take. This is, of course, fraught with hazard. Any time any piece of software makes a prediction of how long something will take it opens itself up to complete ridicule! This is merely the best guess and is based on 'future messages being much like the ones I have seen'. As such the actual answer could vary wildly. For example the queue could be in use. If more messages are added or removed from the queue during the processing then clearly this will affect how long the operation takes. Another possible problem is varying message sizes. If the first few messages on the queue are small and these are followed by lots of 100 MB messages then again MO71 may report a earlier completion than is possible. Needless to say if the messages get progressively smaller then in all likelihood MO71 will complete faster than its initial estimate.

Finally, there is an 'Abort' button. Click this button if you wish the operation to stop. Exactly where it stops is, of course, not an exact science. If you are fast enough then perhaps nothing has happened. However, it is more likely that 'some' messages have been moved/copied. You may need to clean up these messages if you want to completely undo the operation.

As we mentioned before the 'Progress Dialog Delay' in the Dialogs tab of the Preferences Dialog controls whether the Progress Dialog is displayed. However, even if you choose not to display the Progress Dialogs for a long time that does not preclude you from displaying it if you wish. Progress Dialogs will be shown in the 'Windows' menu of the main window. By selecting the Progress Dialog in the Windows menu the Dialog will be shown. Similarly if you close the dialog and the operation is still continuing the dialog is merely hidden and can be redisplayed by selecting it again in the Windows menu.

7.4. Long Operations

MO71 is not really designed for moving large numbers of messages around not least of which because MO71 may well be connected to the remote server over a client connection. Where possible you are probably better to use a local application, something like our QLOAD product. However, there is no denying that being able to just quickly move or copy a few messages in your standard Administration GUI is convenient, quick and easy. However, you need to bear in mind that MO71 will use the same connection to the Queue Manager for the message move/copy as it does for Administration. This means that until the move/copy operation is complete you will not be able to do other useful stuff, such as display your queues and channels.

Most copy/move operations will complete in just a few seconds. However, if you find yourself regularly issuing copy/move commands which take a long time to complete it might be worth either running a different instance of MO71 or creating another location to the same Queue Manager and using just that one for the copy/move operation. For instructions on how to create more than one connection to the same Queue Manager please see section 3.2.3 Creating two locations for the same Queue Manager on page 10.

Chapter 8. Naming Objects

One of the important aspects of MQ Administration is to try and stay organised. An MQ configuration which is just allowed to grow without any control can soon prove to be very messy and hard to manage. One of the key methods of staying organised is to develop and enforce a naming standard for all your MQ objects.

There is perhaps no right or wrong way of naming MQ objects so exactly what your particular standard is is not that important. It is more important that you do have a standard and that you ensure that you stick to it. However, to get you started, here are a few suggestions which you decide to adopt which have proved useful in other MQ installations.

1. **Use upper-case names**
   MQ names are mixed case and are case sensitive. However, sticking to upper-case names has a number of advantages. They are more distinctive when reading design documents. There is less room for confusion when coordinating definitions with another administrator. When using MQSC there is less room for error since MQSC will, by default, upper-case any unquoted entered strings.

2. **Don't use long Queue Manager names**
   The z/OS product only allows four character names for Queue Manager but on Distributed you can have up to 48 characters. One could easily argue that four is too few and 48 is far too many. Aim for Queue Manager names less than 10 characters if you can. They are easier to remember, easier to type and don't take up so much space in reports. However, remember the golden rule....all Queue Managers in your network should have unique names.

3. **Use an application group high-level qualifier**
   Queue Manager resources, like file systems, are shared amongst multiple application groups. It is therefore advisable to use a unique high-level qualifier per application group to signify the ownership of the MQ objects. Furthermore this high-level qualifier should be placed at the front of the object name. So, for example, application group 'Sales Information and Forecasting' could have a qualifier of **SIF**. The use of qualifiers has a number of advantages. Firstly it avoids name clashes. As long as the high-level qualifier is unique then you won't run into the problem of two application groups developing applications which use the same named queues. Secondly it is much easier to separate objects on display. The base MQ product only supports wildcards at the end of the strings. So, for example DIS Q("SIF.*") is supported however DIS Q("*.SIF") is not[18]. Having high-level qualifiers at the start of the name can also be useful when declaring security profiles.

4. **Avoid punctuation characters except '.' and '_'**
   MQ does allow % and / to be used in object names. However, these can be hard to read and lend themselves to misunderstanding. For example, a common way of representing a Queue Manager and Queue combination is to write QMXYZ/SIF.SALES.FORECAST. This is easy to parse and understand. However, if you use punctuation characters in object names this becomes QMXYZ/SIF/SALES/FORECAST which is much easier to misinterpret.

5. **Include the target Queue Manager in channel objects**
   It is useful, as an aide-mémoire if nothing else, to include the name of the Queue Manager where messages are going to down a channel. Consequently a quick look at a channel status gives you a quick indication of the current messages flows. So, for example, a channel name such as TO.QMXZY would be acceptable.

6. **Use a Queue Manager usage suffix**
   In many installations users have development Queue Managers, testing Queue Managers and production Queue Managers in the same network. This is potentially hazardous. Clearly you do not want to risk messages leaking from your test systems to your production systems. This is one of the reasons you are strongly recommended to always have uniquely named Queue Managers to avoid unintentional cross-talk. However, it can also be useful, to add a suffix to the name of your objects. This reduces the chance of cross-talk still further and can be a useful documentary aid to show the intended use of an objects. So, for instance, in the example above you could consider calling the queue SIF.SALES.FORECAST as SIF.SALES.FORECAST.DEV on the development Queue Managers and SIF.SALES.FORECAST.TST on the test Queue Managers. You could even go so far as to call it SIF.SALES.FORECAST.PRD on the production Queue Managers. This would make it very clear to an administrator that this was a production object and that they should be very careful about issuing command such as CLEAR QLOCAL.

[18]Of course this format of command is also very useful. In fact allowing wildcards in any part of the object name has long been a requirement against the MQ product. If this is of interest to you and you would like this form of command supported then take a look at MQGem product MQSCX. This, extended MQSC processor supports unlimited wildcards and a whole host of features besides.

Following these simple guidelines will help you keep your objects organised and will help to avoid some potential pitfalls. Of course ensuring that the guidelines have been adhered to can be more difficult and time consuming than it might first appear. Often MQ installations have object definition running into many hundreds. What we need is a simple way to ensure that the guidelines are followed and to warn if they aren't. To this end MO71 offers three features concerned with object naming.

8.1. Object Name Patterns

The key to any object name checking is clearly having some form of schema to check against. As we said before there is no right and wrong way to name MQ objects and each MQ installation will have done something slightly different. So, what we need is a way of describing our particular pattern of object names. There are two ways that you can specify a name pattern and the intention is that you would use both in conjunction with each other. Firstly you have the global name pattern which applies to all of the Queue Managers in your domain. Secondly you have the local name pattern which is specified in the location dialog. The local name pattern just identifies the aspect of the name which is specific to this Queue Manager. Usually you would use the local name pattern just to specify the Queue Manager usage suffix.

8.1.1. Global Object Name Patterns

The global object name pattern is the way in which your naming convention is described. MO71 defines a very simple pattern schema for this which is as follows[19]:

Character Sequence	Meaning
()	Required section
[]	Optional section
|	Represents OR. For example, ABC | XYZ says that the characters ABC or XYZ must be present
*****	Matches 1 or more upper-case characters up to the next punctuation character
******	Matches 1 or more upper-case characters
$	Matches 1 or more any-case characters up to the next punctuation character
$$	Matches 1 or more any-case characters
#	Matches exactly 1 numerical digit (i.e 0-9)
<#>	Matches zero or more digits (i.e 0-9)
<qmgr>	Matches the 'current' Queue Manager name

So, for our queue example above we may have a queue naming pattern such as:

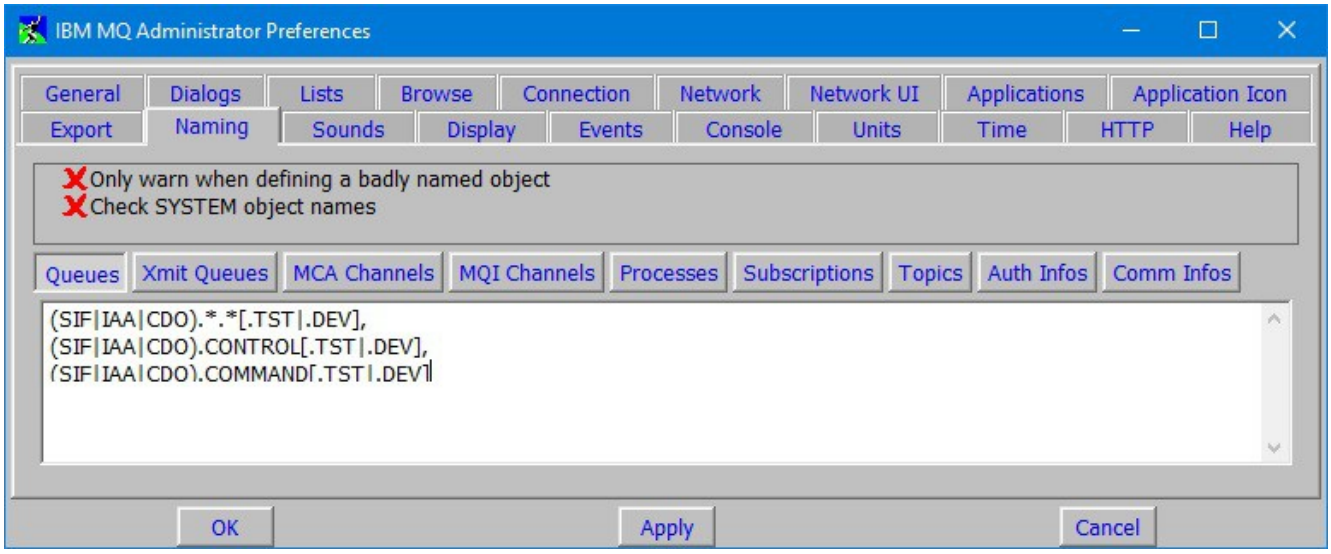
```
(SIF|IAA|CDO).*.*[.TST|.DEV]
```

The pattern can be made as specific or as general as you like.

[19]Clearly MO71 could have used something like regular expressions. There is no doubt that regular expressions are powerful. However, with that power comes a lot of complexity. In addition the syntax of regular expressions doesn't blend well with MQ object name. For example, the '.' (dot) character is very commonly used in MQ objects and it has a special meaning (a wildcard) in regular expressions.

8.1.2. Global Object Name Pattern Specification

The global object name patterns themselves are specified in the Preferences Dialog on the Naming tab. Here you can specify the name patterns for a variety of MQ objects. The most commonly used ones are those for queues and for channels. The key thing here though is that these patterns should apply to all of your Queue Managers.



You can see from the dialog that you can specify a different pattern for 'normal' queues and for transmission queues. This is because transmission queues do not follow the same naming scheme as 'normal' queues. For example, they are frequently just the same name as the target Queue Manager. Similarly you can specify different naming patterns for MCA Channels (Senders,Receivers etc) and MQI Channels (ClntConn and SvrConn), again this is because these two different types of channels are often named differently.

For each object type you can specify a comma separated list of patterns. The patterns themselves cannot contain blanks or newline characters but you can have newline characters between each pattern.

If multiple patterns are defined for an object then clearly an object is considered to be correctly named if it matches any of the patterns.

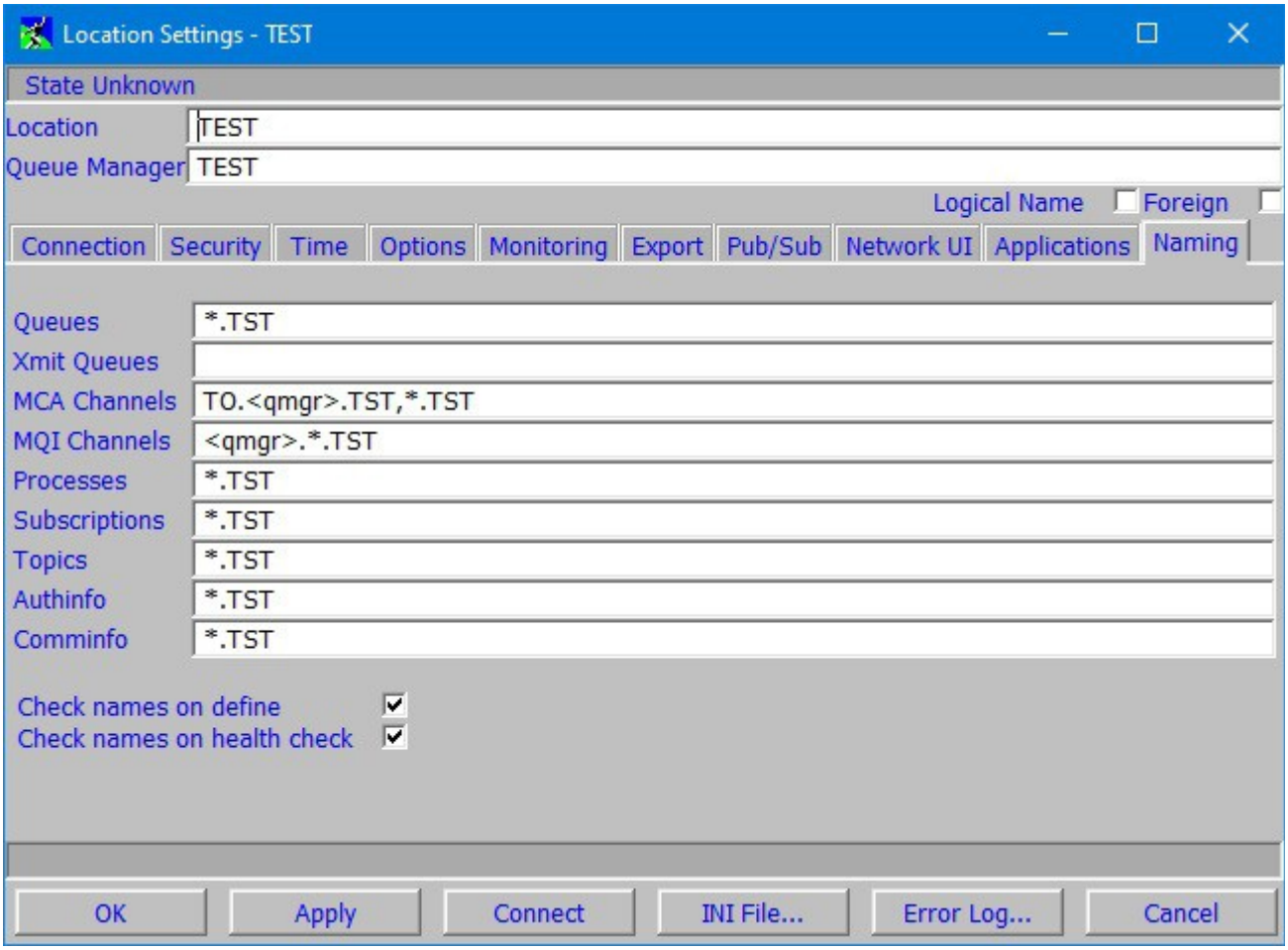
This dialog also allows you to specify two options:

- **Only warn when defining a badly named object**
  If a pattern is specifies for an object and an attempt is made to define an object with a non-matching name then a message is displayed preventing the definition. However, if you wish, by using this option, you can downgrade this message to a warning. This means that the user is given an indication that the object may be incorrectly named but if they wish they can continue and define the object anyway.
- **Check SYSTEM object names**
  Normally the expectation is that patterns do not include SYSTEM objects. An MQ Administrator should already be very wary of defining SYSTEM objects. So, if an Administrator does define a SYSTEM object MO71 will allow it. However, by checking this option you can cause MO71 to ensure that the object name does match the pattern. In this way you can configure MO71 to either prevent the definition of SYSTEM objects or limit the SYSTEM definitions in some way.

8.1.3. Local Object Name Pattern Specification

In the location dialog there is also a 'Naming' tab. This allows you to specify a simple local pattern for the objects.

You don't have the full scope of the global naming patterns mentioned earlier since they are not being used to specify the full name pattern. Instead they are just used to identify that portion of the name that is specific to this Queue Manager. To that end you can only specify the wildcard character '*' and the Queue Manager identifier <qmgr>. So, on our test Queue Manager we may have a dialog which looks like this.



Note that in this case the wildcard character '*' really does mean any character and should not be confused with the global name pattern version of '*'.

You can see that we can specify a comma separated list of patterns but that these patterns tend to be a lot simpler than the patterns specified in the Preference Dialog. This is because all we need to do is specify those elements of the name which are tied to this Queue Manager. In this case we are essentially saying that the MQ objects should all end with the string .TST.

8.2. Object Define Name Checking

A check-box in the naming tab in the location dialog controls whether MO71 will check whether your names match the name patterns at define time. Any defined objects must match both the local and global name patterns if defined. You can choose whether MO71 should just display a warning dialog or whether the define should be prevented entirely. This is controlled via an option in the Preferences Dialog.

8.3. Object Name Health Check

An option in the naming tab of the Location Dialog controls whether MO71 will check any names at health check time. If checked the Verify Display of the Network Display window contains a list of problems discovered in the configuration. This check is performed for the major objects types, queues, channels, processes and topics. Being able to suspend this name check is useful if

you have certain Queue Managers, perhaps belonging to a different administrative domain, which do not follow your own naming convention.

8.4. Object Name Matching

Since version 6.0 MO71 has supported comparing object definitions between two Queue Managers. However, previously this was limited to 'same named' objects. So, queue SIF.SALES.FORECAST on one Queue Manager could only be compared with SIF.SALES.FORECAST on another Queue Manager. As we have discussed earlier this can be a little limiting since you may well want to compare your test and production environments.

So you may well be wanting to compare SIF.SALES.FORECAST.TST on your test Queue Manager with SIF.SALES.FORECAST.PRD in your production Queue Manager. You can now do this using the Queue Manager compare patterns. It is possible that you may wish to compare SIF.SALES.FORECAST.TST with SIF.SALES.FORECAST. This can be done using the patterns *.TST and * respectively. Please see Comparing Queue Manager Objects on page 145 for a description of how to compare MQ object definitions.

Chapter 9. Network View

The network view allows you to relate the various objects in the various Queue Managers. It is started by the menu option *'View-View Network'* from the main window. By selecting *'Action-Auto Start'* from the network dialog menu itself the Network view can be set to start automatically whenever MO71 starts up.

Both manually defined objects and cluster objects will be displayed in the Network View.

The main window can either display one or two windows. If two windows are displayed a sliding bar allows the user to set the proportion of the main window given to each sub-window.

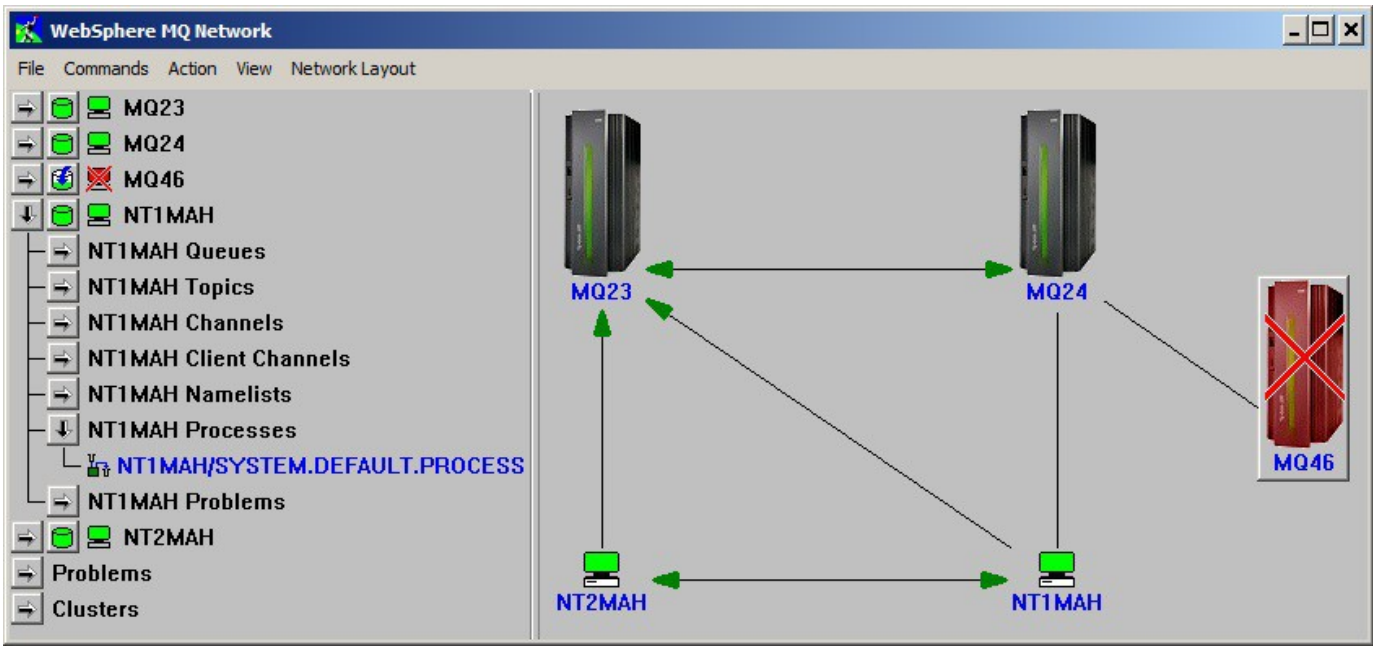


Figure 15: Network Window

9.1. Network Menu

A number of menu options are provided

9.1.1. File

- **Preferences**
  Show the Preferences Dialog where various application global values can be set. In particular the Network and Network UI tabs of the Preferences Dialog will change the appearance and behaviour of the Network view.
- **Exit**
  Ends the Network display dialog.

9.1.2. Commands

Shows the list of commands available for the currently selected Queue Manager

9.1.3. Action

- **Re-Analyse**
  Causes the program to re-analyse the currently retrieved data, for example check resolved queue paths, and re-display results.
- **Refresh Data**
  If selected a new set of definition and status data will be retrieved from the Queue Manager. A choice is provide whether all Queue Managers or just all the connected Queue Managers are refreshed. There is an option on the main window to refresh just the single selected Queue Manager.
- **Auto Start**
  Option to selected whether the Network display should be automatically started and shown when the MO71 program is started.

9.1.4. View

- **Search**
  Controls whether a search bar is displayed in the 'Verify' display.
- **Search Status**
  Controls whether the search status is displayed in the 'Verify' display.
- **Descriptions**
  Controls whether the objects descriptions are shown in the 'Verify' display.
- **Show MQ Version**
  Controls whether MQ version for the Queue Manager is displayed in the various network displays.
- **Show Channel Status Counts**
  If selected then the last count of the number of MCA and Clients channels running on this queue manager is shown. The high water values are shown in brackets. If no response has been received from this location yet then N/A is displayed.
- **Show Foreign Locations**
  Controls whether foreign locations are displayed in the network displays. Foreign locations are 'read only' locations which are usually detected through either monitoring or channel status displays. You can't issue commands to a foreign location.
- **Problem Selection...**
  This menu will bring up the Problem Selection dialog where you can:
  - choose which health problems are checked for
  - see a short description of the problem
  - see which instances have been found.
- **Show only problems**
  When selected the verify window will only show the problems lines for the selected Queue Managers. This allows you to easily check which Queue Managers have potential outstanding health issues.
- **Left/Right Window**
  This menu option allows you decide which of the displays should be shown either on the left or right of the panel. By selecting 'none' you can choose for only single display to be shown.

9.1.5. Network Layout

- **Show location name/Show Queue Manager name**
  Allows control over which name is used in the display
- **Scale**
  When checked all Queue Managers will be scaled to fit on a single page. With scaling switched on moving Queue Manager at the extremes of the window can produce unexpected results.
- **Show Grid**
  When selected an array of dots is displayed. The size of the grid is determined by the Grid size specified in the Preferences dialog.
- **Snap all to Grid**
  When selected all Queue Managers will snap to their nearest grid co-ordinate. This can be useful to align Queue Managers for neatness.
- **Highlight Channels**
  When selected the channel lines from the selected Queue Manager are displayed with a thicker line in the highlight colour to make it easier to see which Queue Managers the selected Queue Manager is connected to.
- **Display layouts**
  See Network Display section
- **Centre QM**
  Will move the selected Queue Manager to the centre of the network display. This can be useful if the current position of the Queue Manager is unknown.
- **Save layout**
  When you are happy with the layout for your Queue Managers you can save the current settings by selecting this option. If you want to back out subsequent changes, pressing the ‘User layout’ option will go back to this save point. Note that the save is purely to memory. You must then press ‘Save Configuration’ on the main window if you want your values written to disk. As usual all changes will be written to disk if the application ends normally.If the layout is changed and the dialog is ended the application will automatically prompt the user as to whether the current layout should be saved.

A number of the windows display a hierarchical tree view known as a container. These containers are very similar to Windows container views displayed in applications such as Windows Explorer. However, an explanation of the keys available is included later in this document.

The sub-windows that can be displayed are described below.

9.2. Network Display

The Network display shows the topology of the Queue Manager network according to the channel definitions. I discovered while writing this code that displaying a set of interconnected nodes in a reliable, eye-pleasing fashion is actually very difficult. For small networks a simple circle is hard to beat. For large networks the problem becomes very difficult very quickly. Ultimately there is probably no substitute for allowing the manual placement of the Queue Managers in the display. I have therefore initially allowed three types of display selectable from the *'Network Layout'* menu.

- **Circle layout**
  This simply displays all Queue Managers in circle. All Queue Managers in the network are displayed.
- **Diagram layout**
  This layout uses the links between the Queue Managers to try an devise a reasonable looking topology. Only interconnected Queue Managers are displayed.
- **User layout**
  Allows the user to manually position each Queue Manager by clicking on the Queue Manager[20], holding the mouse button and moving it to the desired location. Note that it is much easier to do this with *'Scale'* switched off otherwise you can get some very strange results.

9.2.1. Moving and Selecting Objects

At any time you can re-arrange any of the objects on the diagram by selecting them and moving them accordingly. By holding the <Ctrl> button when you click you can select multiple objects. In addition you can select all objects in a box by holding the <Ctrl> button as you click on the background. Drag the box to enclose the objects you want to be included. All selected objects will be moved as a single block.

When you move an object it may 'snap' into position. Snapping helps align the objects in the display and is set in the Preferences Dialog. You can choose between 'No snap','Snap to Grid' and 'Snap to Qmgr'. Snapping to grid will try to snap any objects as they are moved to fixed grid of positions. Snapping to Qmgr will try to align the object to any other Queue Managers which are 'close'. When you are moving multiple objects you will notice that each object may snap at different times. Often this is what you want since it maximises the amount of object alignment however it can appear disconcerting at first. At any time while moving an object you can temporarily suspend any snapping by holding the <Alt> button.

You can also 'snap' the object to it's starting position if you wish. For example, suppose you wish to move an item left or right without any vertical adjustment. You can do this by pressing <Shift> as you make you move. The item will tend to stay on the same axis. However, if you move the item too far away from the object axis then the actual position with be honoured and the axis snapping is ignored. Once you have positioned all the objects to your satisfaction select the 'Save Layout' menu option.

9.2.2. Display Format

The network display will show the Queue Managers that have been selected as part of the 'network'. If a pair of Queue Managers have at least one channel pairing between them then a line will be drawn between the two Queue Managers showing that messages can flow between them. Only a single line will be shown regardless of how many channel pairs are defined between the Queue Managers. To indicate the state of the channels between the Queue Manager pair the ends of the lines may have a number of different symbols. Since a single line could represent a number of different channels it is necessary to define some order of precedence. This list is therefore given in precedence order.

- **Arrow** At least one channel is running in direction of arrow
- **Lightening** No running channels and at least one channel in retrying or binding
- **No entry symbol** Channel stopped
- **No symbol displayed** Channel inactive

[20]Note that moving a Queue Manager manually while in any layout will change the layout to User.

See “Appendix A: Icons” on page 267 for examples of what these icons look like.

In the preferences dialog you can set how often the channel status information is retrieved from the Queue Managers. See the Network tab of the “Preferences Dialog” on page 193 for more details. The preference dialog also contains a number of configuration options which allow you to modify the way the network display looks.

Double clicking on a Queue Manager in the Network Display will display the Application View for that Queue Manager, provided Application displays are enabled in the preferences. For more information please see Chapter 10.Application Monitoring on page 89.

9.3. Verify Display

This window will display the definitions that MO71 has stored for each of the Queue Managers in the network[21]. Each Queue Manager contains a hierarchical display. The levels of the hierarchy are expanded by pressing on the horizontal arrow, and collapsed by pressing on the vertical arrow. The complete hierarchy is shown in the 'levels display' but essentially it contains all the major objects defined for the Queue Manager. In addition it can show the Applications that MO71 has seen running on the Queue Manager. This can very useful to get a quick feel for what applications are running and what objects they have open.

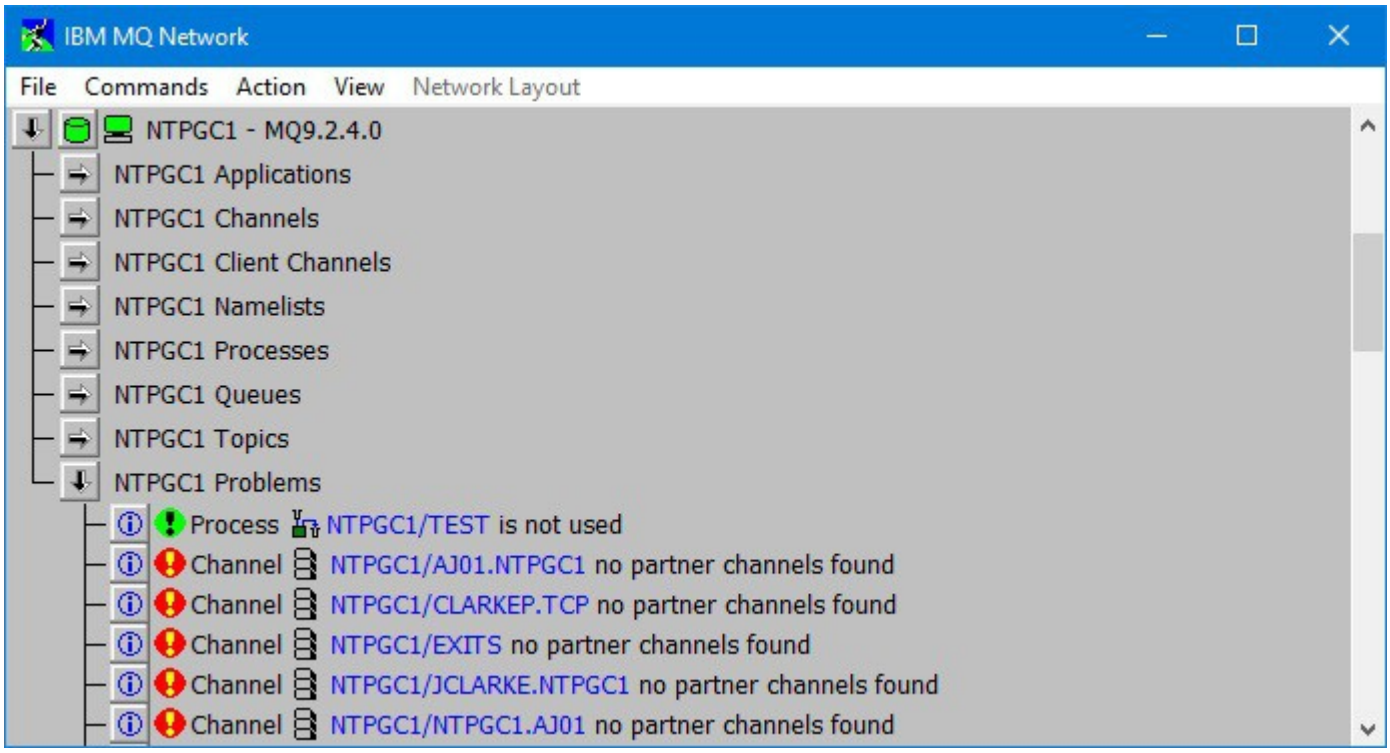


Figure 16: Verify Window

Some of the objects will also have sub-trees to relate each object to other objects in the Queue Manager and other Queue Managers in your network. For example a remote queue will expand to show which transmission queue it resolves to, which in turn will expand to show the remote Queue Managers that the transmission queue services. Each of those Queue Managers will expand to show the channels used to get to them from this transmission queue. Once at the target Queue Manager the path resolution process continues until ultimately it will show the local queue that the remote queue resolves to.

- Similarly, local queues will expand if they are resolved to from another queue in the network.
  Path resolution can be a complicated, time-consuming task and as such can be switched off in the Preferences dialog. For details see "Preferences Dialog" on page 193.

The channel objects will expand if a partner channel (i.e. one with the same name) is detected in the network.

- An additional ***problems*** sub tree may be displayed after the channels sub tree if enabled in the Preferences dialog (see "Preferences Dialog" on page 193). The Problems sub tree contains a list of all the possible problems detected by the application in this Queue Managers configuration. Note that this is just a guide and a highlighted problem may not in fact be a problem. For example, the application expects each Queue Manager to have a defined Dead Letter Queue which is a strong MQ recommendation. There are, however, good reasons why the user may not wish to use a Dead Letter Queue. For this reason the user may use the problem selection window (see "Problem Selection" on page 86) to disable the checks which are not wanted. It should be pointed out that the problems displayed are not an exhaustive test, while I do intend to increase the number of consistency checks over time, there is no guarantee whatsoever that all problems will be detected.

In the object hierarchy various queue manager names, queue names and channel names will be displayed. These are really a form of push button. Clicking on the name with a mouse or pressing <space> while the object has focus will bring the up a dialog of that object allowing the user to examine its attributes and make changes if necessary.

[21]Note that the Queue Manager will only appear in the list when definitions are received from the Queue Manager

9.3.1. Verify Display Search

It is possible to search the verify data for certain strings, including wildcards. By selecting 'Search' from the *'View''* menu a search field is displayed above the verify data. By selecting 'Search Status' an additional line is displayed which shows exactly what is being search.

The format of the search string is fairly straightforward. A comma (or space) separated list of search strings can be entered. The search strings can contain the normal wildcards '*' or '?'. '*' represents 0 or more characters, '?' represents exactly one character. If a list is specified the results show the results or either search; in other words the searches are additive.

When data is entered into the search field the search will applied after the configured delay interval. This is set in the 'Network' tab of the preference dialog.

9.3.2. Automatic Search Wildcards

A preference option, in the Network tab, controls whether '*' are added to the entered search string automatically. For example, with the option selected, entering 'abc' will actually search for '*abc*'. In other words any entry containing the string 'abc'. By enclosing the search string in double quote '"' characters the automatic wildcard addition can be suspended. Entering '"abc"' will just search for the string 'abc' without any additional wildcards. In fact, the double quote can be entered independently at each end of the search string. So, if one wishes to search for entries which start with the string 'abc' one can enter "abc.

9.3.3. Case Sensitivity

Whether the search is case sensitive or not is controlled by a preference option in the Network tab.

9.3.4. Search Qualifiers

It is possible to qualify the search so that only certain objects or fields are searched. This allows the search results to be significantly reduced, for example, only show the queues containing this string. Qualifiers are entered as a list between '<' and '>' brackets. So, for example the search string **abc <q c>** will just search for queues and channels containing the string 'abc'. The search string **abc <q c desc>** will only search the description field of queues and channels for the string 'abc'. The full list of qualifiers which can be used are:

Qualifier	Qualifier Shortform	Short	Effect
Application	**app**		Application Name field
Channel	**chl**	**c**	Channel Objects
Cluster	**cl**		Cluster field
Description	**desc**	**ds**	Description field
Mqmaint	**mai**		MQ maintenance
Name	**nm**		Name of object
Namelist	**nl**		Namelists
Process	**prc**		Process objects
Qmgr		**m**	Queue Managers
Queue		**q**	Queue Objects
Reference	**ref**		Reference For example, the base queue of an alias queue definition.
Tag			Application Tag
Topic		**t**	Topic Object
TopicStr	**ts**		Topic String field

The abbreviated names, if given, can be used instead of the full name.

It can be useful, in some cases, to effectively use the qualifier on its own. For example, the search string *** <q>** will just show entries relating to queue objects. Similarly **?* <q desc>** will show only queues containing descriptions.

9.4. Queue Managers Display

This window displays all the configured Queue Managers and allows the user to indicate which Queue Managers are part of the 'network'. The 'network' of Queue Managers is the subset of Queue Managers which will be acted upon by the Network View. Not all of your configured Queue Managers might be considered part of your immediate network and they can therefore be excluded from analysis. Each Queue Manager can either be part of or excluded from the network.

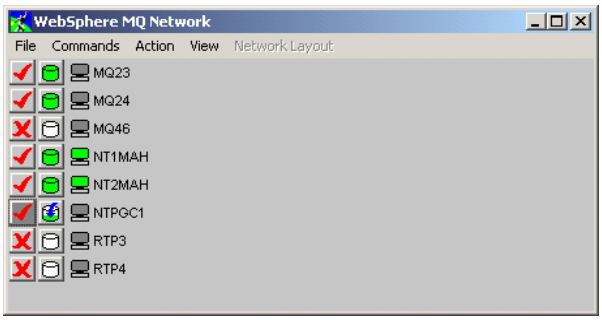


Figure 17: Queue Managers Display

A Queue Manager is part of the network if a 'tick' appears next to its name. A Queue Manager is not part of the network if a 'cross' appears next to its name[22]. Clicking on the icon (or pressing <space> while it has focus) will alternate between the two choices. Each location can be defined as *associated* with a number of network names. This allows many Queue Managers to be included or excluded from the network view in a single mouse click. See "Network Names" on page 88 for a description of this.

The next icon after the tick/cross indicates whether object definitions have been received from that Queue Manager. The icon can either be empty (white), full (green), overdue (red) or requesting (half full with arrow). At any time the user can press this button and the application will get a new set of definitions from the Queue Manager. See "Appendix A: Icons" on page 267 to see what these icons look like. In the 'Network' tab of the preference dialog there are values which can set by the user to control the automatic retrieval of the object definitions and what constitutes 'out of date' definitions.

9.5. Levels Display

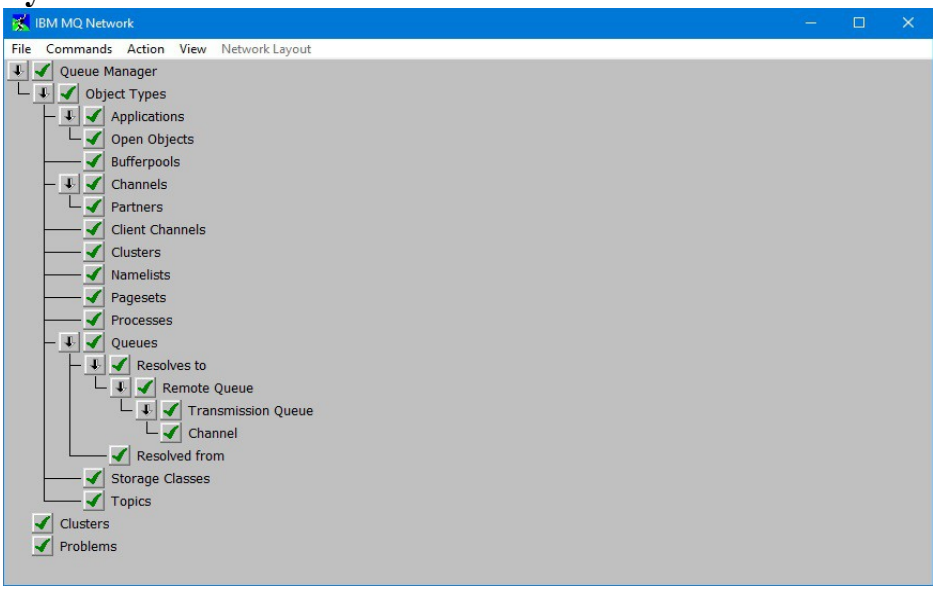


Figure 18: Network Levels Display

The Verify window can contain quite a complicated hierarchy. This 'Levels' display shows the various elements of this hierarchy. By enabling or disabling various levels elements can be removed/displayed in the Verify window. Selecting and removing levels in the hierarchy only affects what is displayed, no change is made to the internal data and no data is retrieved from the remote Queue Managers.

[22] A preference option is available which will just display a blank rather than a cross.

9.6. Problem Selection

This window displays the complete list of problems the application tries to detect. By default all problems will be checked. However, by removing the check mark next to a problem the application will no longer check for this situation. If no problem detection at all is required a single check box in the Preferences Dialog allows all checking to be switched off (see “Preferences Dialog” on page 193 for details).

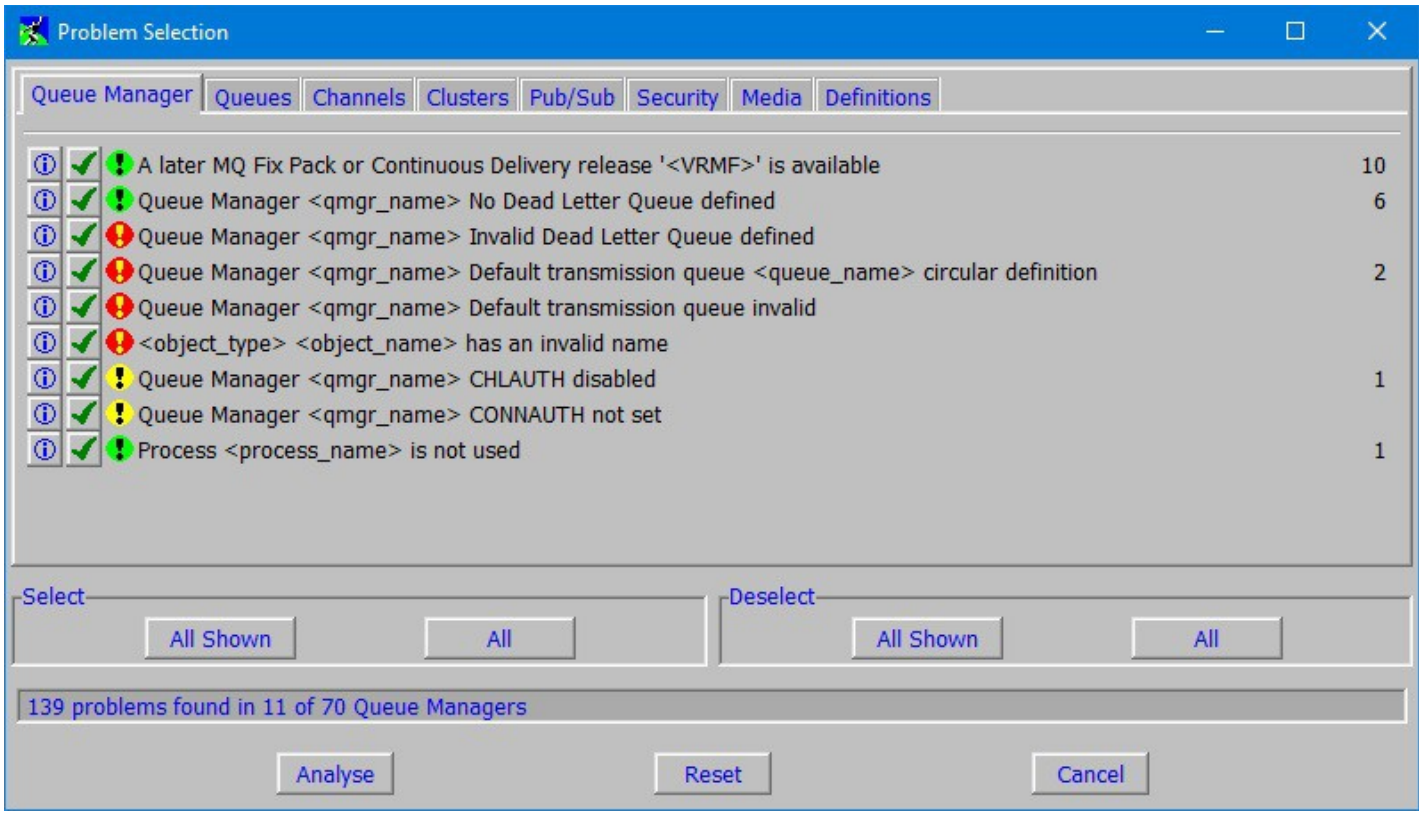


Figure 19: Problem Selection Dialog

Above we see an example of the selection dialog. We can see that the dialog is split into tabs for each category of health check. Note that a specific problem could appear in multiple tabs. For example, checking that a channel has a defined Cipher Spec, is a problem which will appear in both the Channels and the Security tab.

Navigating the dialog is fairly straight-forward. As we mentioned before any problem which has a check mark will be verified in the Network view for the selected Queue Managers. Once the Network view does the analysis it may find some instances of that problem and on the right-hand side of the selection will be displayed how many instances of the problem it found.

The following buttons are provided:

Select	**All Shown**	-	All problems for current tab are selected
	All	-	All problems are selected
Deselect	**All Shown**	-	All problems for current tab are deselected
	All	-	All problems are deselected
Bottom	**Analyse**	-	Problems are re-analysed and new counts shown
	Reset	-	The problem selection is reset to those that were selected when the dialog was started. This is useful if you wish to just concentrate on a subset of problems and can help to make it clearer which Queue Managers still have issues.

We can see in the example above that we have found quite a number of problems. Bear in mind that this was done on a test system where problems are deliberately created! Suppose we are interested in more information about one of the problems. We can click on the blue i symbol and MO71 will show you a bit more information about the problem and, if the network view is running, a list of the instances where that problem was found.

So, suppose we clicked on the blue i next to the "No Dead Letter Queue" problem we might get the following.

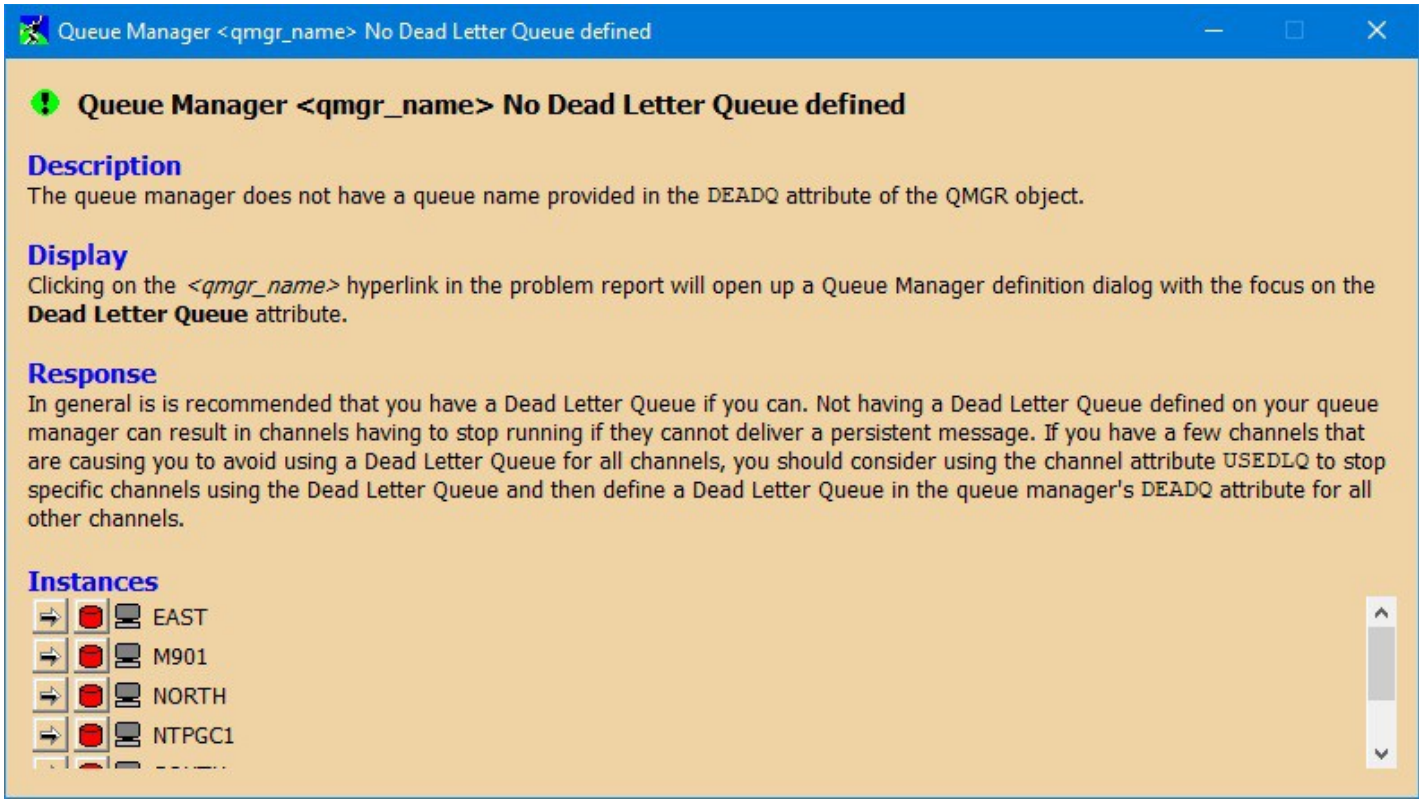


Figure 20: Problem Description Dialog

You can see that MO71 gives a brief explanation of the problem and why it might indeed be a problem and lists how many instances of the problem it has found (if any). Of course for your installation this 'problem' might not be an actual problem but might be the result of installation policy. You may have decided, in this case for example, not to run your Queue Managers with a Dead Letter Queue. This is perfectly fine and, as we mentioned before, you can tell MO71 to not bother checking for this situation. However, if you decide to fix the problem then expanding the instance descriptions will generally lead you to a description with links you can press on which will bring up the appropriate dialogs to change the definitions. Alternatively by right-clicking on the instance line the context menu will be displayed giving access to the full range of commands to that Queue Manager.

The list of problems is clearly an area which could be expanded in the future. I would be interested in hearing anyone's suggestions for problem checks. In particular I am interested in problems which are not easily seen from a single object definition but rely on the interaction between two or more objects.

More discussion around the subject of Queue Manager problem detection can be found on our blog here.

9.7. Network Names

Each location definition has a field to specify a comma (or space) separated list of Network Names. The names themselves can be anything which makes sense for the particular Queue Manager organisation. For example, you could have each location be part of three network groups. The first identifying the type of machine, the second identifying the geographical location, and the third identifying the business function. So, we could specify a value of **'iSeries, MidWest, DataCenter'**. Each name can contain any alphanumeric character but not a space since a space indicates the start of the next name. Each name must be 20 characters or less.

Each unique name is displayed in the Network Names window together with a check mark. Selecting the check mark will then either include or exclude all Queue Managers with that network name from the network view. This allows easy inclusion of many Queue Managers in a single mouse click if analysis and displays only required on a subset of the Queue Managers. In the example above, for instance, it would be easy to only display the machines located in the MidWest or only the iSeries machines. In addition to the defined network names the standard Network Name of 'All' is always available which, as its name suggests, applies to all Queue Managers.

Note that Queue resolution and problem analysis applies to only those Queue Managers actually in the network view. Consequently if some Queue Managers are excluded full path and channel resolution is not possible. This may lead to problems being reported, for example, "No partner channels found", which are caused purely by the resolving Queue Manager not being included in the analysis.

Chapter 10. Application Monitoring

MO71 will periodically query what applications are connected to the Queue Manager. The monitoring frequency can be set in the location and preference dialogs. MO71 will query not only what applications are running but what MQ objects they currently have open. This information is persisted across restarts so, over time, an internal picture is generated of what applications connect and what objects they open. Application monitoring can be useful for the following reasons:

- It can provide a quick way of viewing what is currently running.
- A history of which applications use which MQ objects is built-up over time.
  For example, you can see which applications regularly use a particular queue.
- Limits on the minimum and maximum connections by an application can be set.
  If these limits are exceeded then MO71 can notify the user. For example, if an important application is not running.
- The state of the applications can be displayed in a application view diagram.

The application data is viewable in a number of different places.

- **Under the Queue Manager in the main window**[23]
  A list of the known applications is displayed. Double clicking on any application will show a dialog containing its definition.
- **In a verify window in the Network or Application views.**
  Under each Queue Manager is the list of known applications. Clicking on the application name will show a list of the matching applications.
- **In an application view**
  The application view allows the user to view the applications in a diagram. This allows the user to see the current state of the applications and their relationship with queues in pictorial form.

10.1. Application Monitoring Configuration

By default application monitoring is enabled but it can be disabled if it is not required. This is done in the Applications tab of the Preferences Dialog. In the Applications tab of the Preferences Dialog you can also set the default refresh intervals to retrieve the application data. If you would like to be more specific and set the refresh sequence by Queue Manager then you can do that in the Applications tab of the Location Dialog.

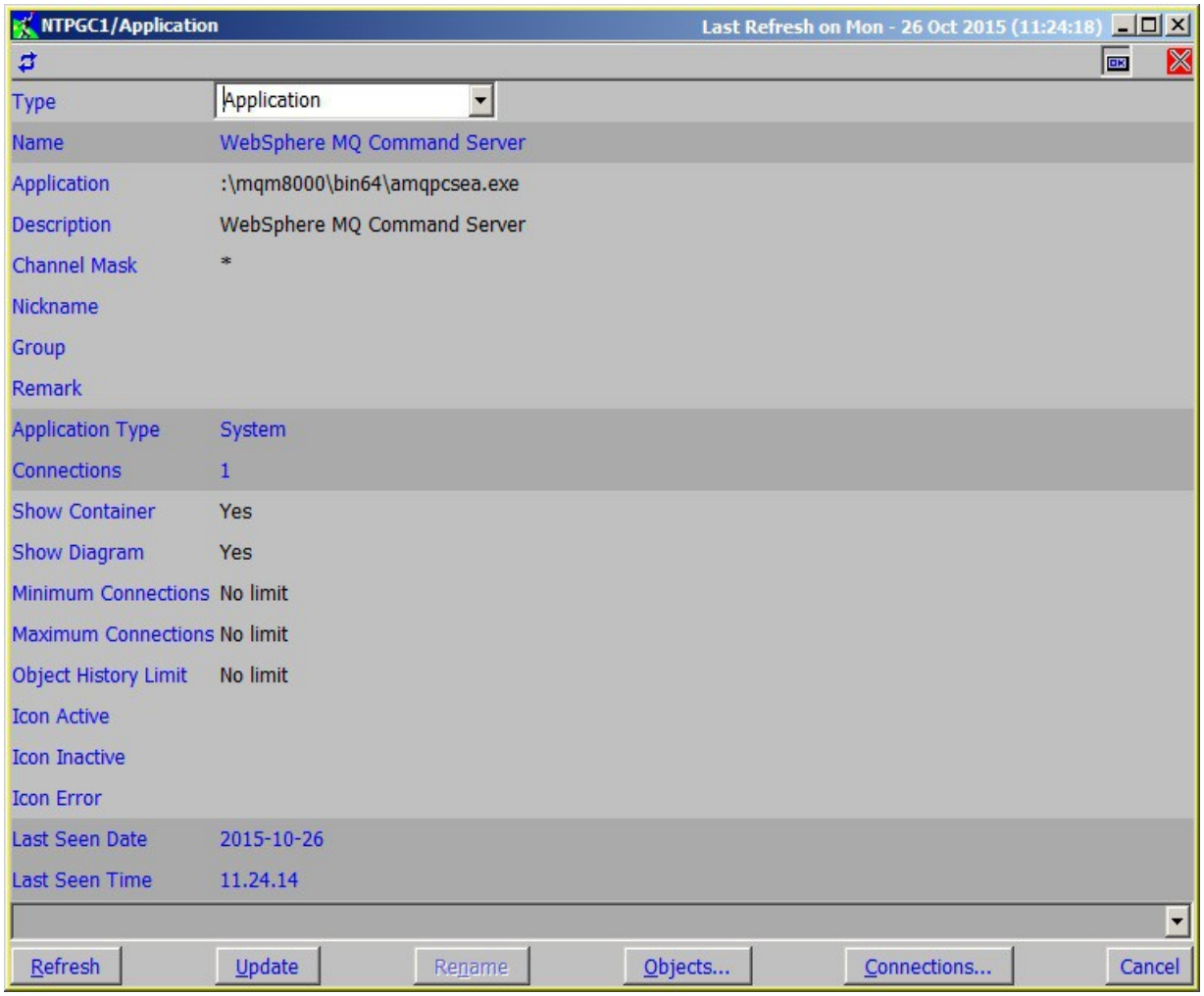
The application definitions themselves are not generally defined by the user. Instead they are automatically defined as data is retrieved from the Queue Manager. However, an application definition can be modified in various way. So, let us look at an application definition.

[23]Providing the preference option 'Show objects in main window' is selected

10.2. Application Definition

An application definition is created in response to a DISPLAY CONN(*) command issued to the Queue Manager. The definition consists of various fields some of which can be modified by the user. The key value is the 'Application' and this is what identifies this application definition as different from any other.

If you bring up an application definition you will see a dialog which looks something like this:



A description of the application fields is given below:

Field	Description
Type	This field identifies the type of record. There are a number of different related records. These are: • **Application** An application record • **Object** A record detailing an application use of an MQ object • **Group** A group record for grouping applications together
Name	The name by which this application should be shown. The name used depends on what values are set in the application. It will be the first of the these fields which are set.... **Nickname** **Description** **Application**

Application	The Application field is the value returned from the Queue Manager for this application. Normally the value of 'Application' will be the *appltag* value from the Queue Manager. However, for CICS and IMS applications it will be the TRANSID and PSBNAME fields respectively.
Description	The Description field value returned from the Queue Manager.
Channel Mask	The Channel Mask field is used to distinguish different, identically named, applications which are using different channels. Please see 'Channel Mask Processing' on page 92 for more information.
Nickname	The nickname field allows the user to identify the application using their own name. This can be useful for two reasons. Firstly, a simple nickname is much easier to remember and recognise than, say, a path and program name. Secondly multiple applications can be given the same nickname. This allows multiple applications to be effectively treated and shown as the same application.
Group	A group name can be assigned to an application. Group records will automatically be created whenever a group name is specified. As their name suggests, groups are useful for grouping multiple applications together. For example, you could group all standard system applications in a group called 'IBM MQ' and perhaps all your tool applications in a group called 'Tools'. Grouped applications will be shown grouped together in the main window and in verify windows.
Remark	The remark field allows the user to write a short remark against this application. Exactly what you use this field for is up to you but you could, for example, write the phone number of the application group who owns the application. This could be useful if you see the application behaving badly!
Application Type	The application type as returned from the Queue Manager.
Connections	The current number of connections to the Queue Manager from this application.
Show Container	A user settable value which states whether this application should be shown in the main window and verify displays.
Show Diagram	A user settable value which states whether this application should be shown in the application view diagram.
Minimum Connection	A user settable value which the user can set if they wish. By default it has the value 'No limit'. However if a value is set then the application will be considered 'in error' if the number of connections is below this limit. Preferences options in the 'Applications' of the 'Preferences Dialog' control whether an application errors will affect the Queue Manager icon and whether they are written to the console.
Maximum Connections	A user settable value which the user can set if they wish. By default it has the value 'No limit'. However if a value is set then the application will be considered 'in error' if the number of connections is above this limit. Preferences options in the 'Applications' of the 'Preferences Dialog' control whether an application errors will affect the Queue Manager icon and whether they are written to the console.
Object History Limit	A user settable value which the user can set if they wish. By default it has the value 'No limit'. However if a value is set then the number of object definitions stored for this application will not exceed this limit.
Icon Active	A user settable value which gives the location of the user icon which should be displayed in the application view diagram when the application is active. Please see Appendix A.7. User Icons on page 269 for more information.
Icon Inactive	A user settable value which gives the location of the user icon which should be displayed in the application view diagram when the application is inactive. Please see Appendix A.7. User Icons on page 269 for more information.
Icon Error	A user settable value which gives the location of the user icon which should be displayed in the application view diagram when the application is in error. An application is considered in error when it has either more or less connections than the configured limits. Please see Appendix A.7. User Icons on page 269 for more information.
Last Seen Date	The last date that this application was seen active.
Last Seen Time	The last time that this application was seen active.

An application record is normally created manually by data being returned from the Queue Manager. The exception to this is where you may want to define an application definition with a different Channel Mask value. However they are created the records can then be modified. This can be useful for various reasons. For example:

- **A nickname can be given to an application**
  This can make it more memorable or it can be used to allow different applications to be treated as the same.
- **An application can be added/removed from a group.**
  Grouping applications together can help to organise them in main and verify windows.
- **Connection limits can be added**
  By default an application can have any number of connections. However, by setting limits you can be notified if an application has either too few or too many connections.
- **Remark**
  A user remark can be added to an application definition. This can be a useful memory jogger about the application.
- **Icons**
  To make the application view diagram even more distinctive it is possible to provide your own icons to be displayed for this application. You could, for example, use the application group logo. Or you could choose to have a picture which represents what the application does. Please see Appendix A.7. User Icons on page 269 for more information.

10.3. Channel Mask Processing

IBM MQ Applications are sometimes not terribly good at identifying themselves. This is often most noticeable in the Java world where you might get completely different applications identifying themselves as 'java.exe' or similar. As time goes on the expectation is that the IBM MQ product will make it easier and easier for applications to give themselves unique identifiers. However, at the moment one technique that people often use is to run these applications over unique SVRCONN channels. This means that the connections can be associated to an individual application using the channel name. This technique has been extended to Application monitoring.

You can create your own Application record, with a different Channel Mask, by using the Create... option on the right mouse context menu, after editing an existing record. By default the Channel Mask will be '*', which will match all applications regardless of what channels they are running over. Suppose you have an application that runs over a particular channel, say CUSTOMER, and wanted to separate it's application monitor. You could create an application record with a Channel Mask of CUSTOMER (plus of course the application tag and description values). The next time MO71 receives the application data, the data for this application would be associated with the CUSTOMER record. The Channel Mask can contain the standard wild-cards '*' and '?'.

10.3.1. Channel Mask Priority

Since an channel mask can contain wild-cards it is, of course, possible that more than one application record could match a returned application. In this case the application will match the most specific definition. The main rule is that the more actual characters, rather than wild-cards, in a value the more specific it is. Of course a mask containing no wild-cards at all would be the most specific of all. So a table of priorities would look like this:

Table of priority.....highest priority at the top
A
?ABC?
ABC
AB?
AB*
A*B*
*AB
A*
*

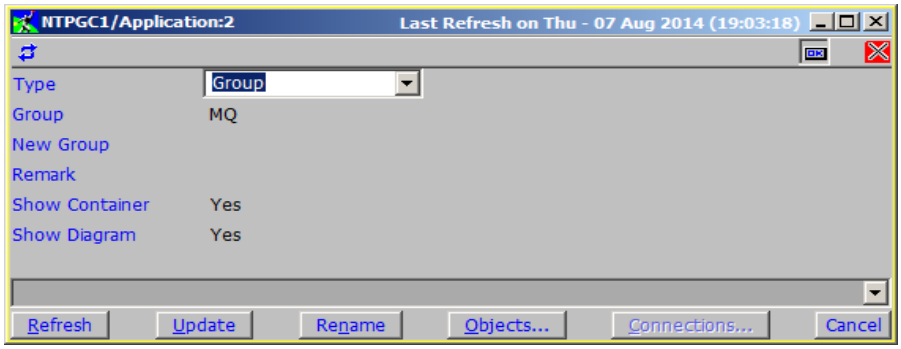
Note that if you do manually create application records then you may need to tell MO71 to 'forget' object records that have been attached to the previous, non-specific, application record.

As we have discussed there are two other types of application record. Let's now discuss these and their fields.

10.4. Application Group Record

An application group record is created automatically as the user fills in a group name for an application record. Any applications who share the same group value are contained in the same group. An application can only belong to a single group. The main purpose of a group record is to allow applications to be grouped together in the main and verify windows. For example, you could group all the standard MQ applications such as Channels, Listeners, Command Servers etc into a single group called “MQ”. You could group all you tools into a group called “Tools”. Whatever makes sense to you for your installation.

If you created and display a group definition you'll see something like this:

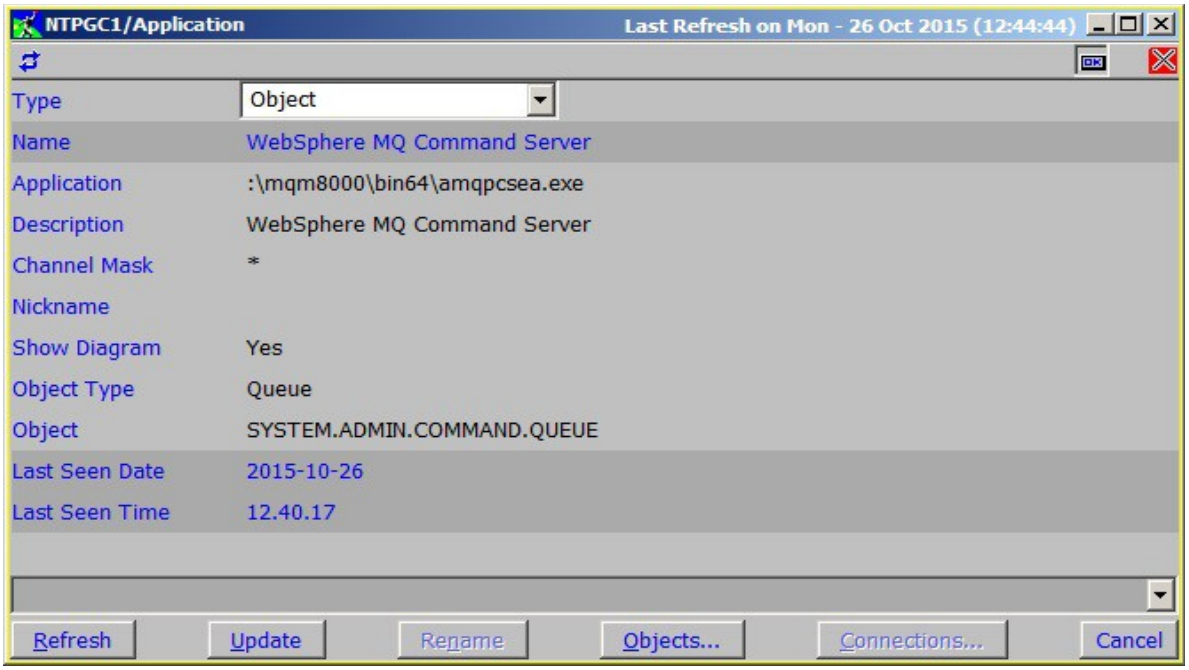


As you can see there isn't a great deal to a group record. Let's discuss the various fields:

Field	Description
Type	**Group** indicating that this is a group record.
Group	This group name.
New Group	This field allows you to rename the group. Enter the new name and click the 'Rename' action.
Remark	The remark field allows the user to write a short remark against this group.
Show Container	A user settable value which states whether this group should be shown in the main window and verify displays. Note that the setting applies to all applications in the group. This allows a simple way for multiple applications to either be shown or removed from a display.
Show Diagram	A user settable value which states whether this group should be shown in the application view diagram. Note that the setting applies to all applications in the group. This allows a simple way for multiple applications to either be shown or removed from a display.

10.5. Application Object Record

An application object record is created automatically as data is returned from the Queue Manager. As MO71 learns that a particular application uses a particular MQ object it will create an object record. The data is persisted across restarts. Over time a picture of which applications use which objects is built-up[24]. These records can be useful to discover the behaviour of applications and their usage patterns. Let's look at an object record and discuss the fields. If you bring up an 'Application List' and specify a 'Type' of 'Object' and press refresh MO71 will show you the object usage it has seen so far. Double click on one of the entries and you should see a dialog that looks something like this:



This record is showing us that, not surprisingly, the MQ Command Server uses the system command queue.

Field	Description
Type	**Object** indicating that this is an object record.
Name	The name of the application using the object
Application	The application value using the object
Description	The application description field
Channel Mask	The active channel mask for the object record
Nickname	The application nickname (if any)
Show Diagram	A user settable value which states whether this object should be shown in the diagram. Only queue objects are shown in the diagram, the value is ignored for other object types. Note that setting this value to 'No' will prevent the queue being shown for all applications.
Object Type	The type of MQ object
Object	The name of the MQ object
Last Seen Date	The last date that MO71 saw that this application was using this object
Last Seen Time	The last time that MO71 saw that this application was using this object

[24]Note however that MO71 only sees periodic snap-shots of usage. If an application opens and immediately closes an MQ object then it is possible for MO71 not to see the usage.

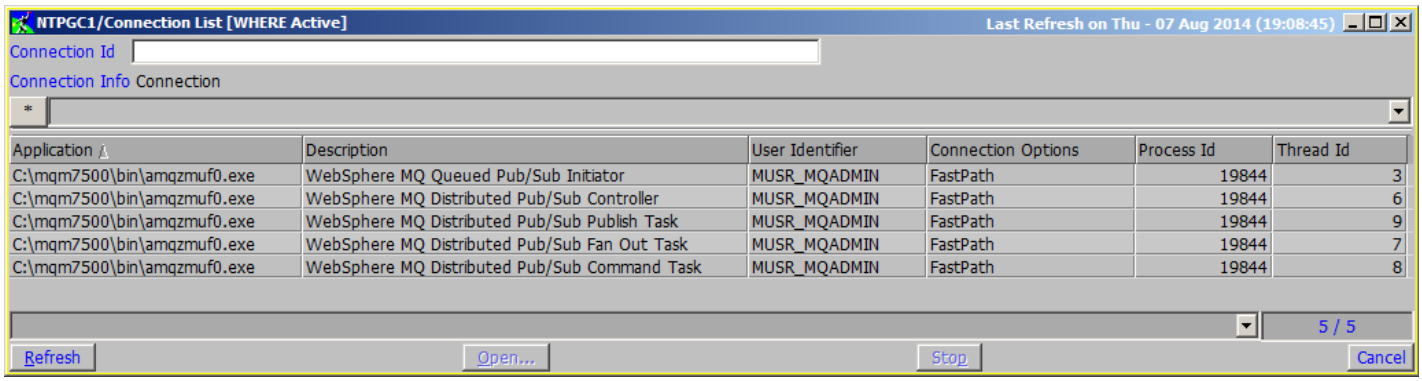
10.6. Application Record Actions

The application record dialogs offer various actions which can be performed. Most of these like 'Refresh' and 'Update' are common with other dialogs so there is no need to explain their function. However, there are some actions which are specific to application records. These are:

Action	Meaning
Clear Group	This action will remove the application from any group.
Objects...	This action will display an application list dialog containing the object records that are known for this application. Essentially this shows all the objects that have been known to have been used by this application
Connections...	This action will display a connection list dialog which shows the current connection information for this application.
Forget	This action tells MO71 to forget about this application or object. This can be useful if the use knows that an application is no longer applicable to this Queue Manager or it's use of a particular MQ object was an aberration. Note how that if MO71 sees the application or object again the record will be recreated.
Rename	This action allows the user to rename a group record.
Set Group...	This action will add an application to a group. It is useful in the list display to add lots of applications to the same group in a single action. When selected a dialog is shown for the user to specify the name of the group.
Unposition	This actions tells MO71 to forget about any positioning information for this record.
Show in Diagram	This action will add the application, group or object to the application view diagram. This allows for multiple items to be selected for display at once
Hide from Diagram	This action will hide the application, group or object from the application view diagram. This allows for multiple items to be selected and hidden from display at once.

10.6.1. Application connections dialog

If you select 'Connection...' from an application record MO71 will show you a connection list dialog for the given application. For example, you might see a dialog like this:

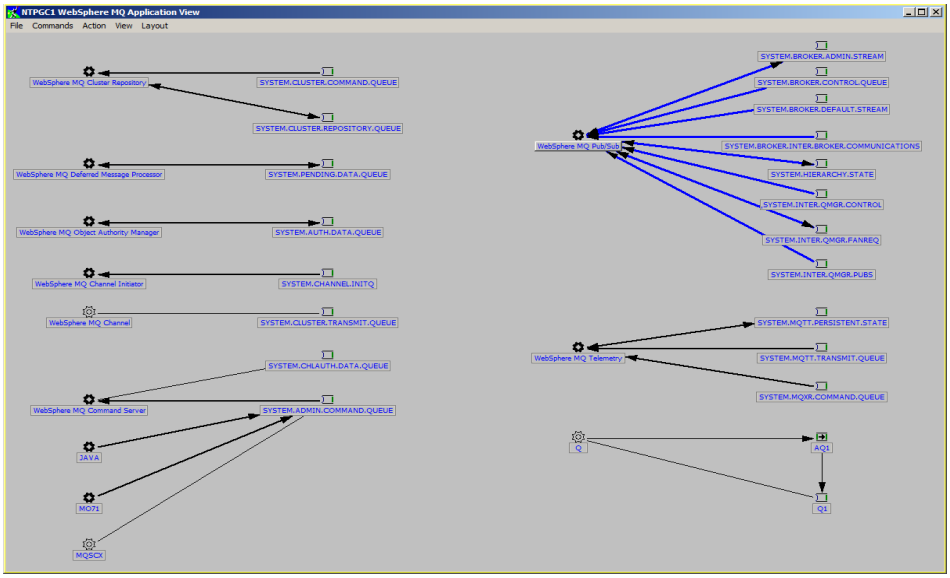


This dialog is almost exactly the same as the connection list dialog you are familiar with. However, there is one key difference and the clue to this is given by the words '[WHERE Active]' in the titlebar. These words tells you that this dialog is limited to displaying applications of a particular name. This filtering is automatic and uses the MQ WHERE clause to limit the data retrieved from the command server. This dialog will only show you a subset of the connections to the Queue Manager[25]. Due to limitations in the WHERE clause showing all the connections of a group or a nickname set of applications is not possible. This is because the WHERE clause only allows a single expression. If you need to see the full list of connections to the Queue Manager then you should bring up a new Connection List dialog in the normal manner.

[25]This WHERE clause can be removed using the where() filter function

10.7. Application View

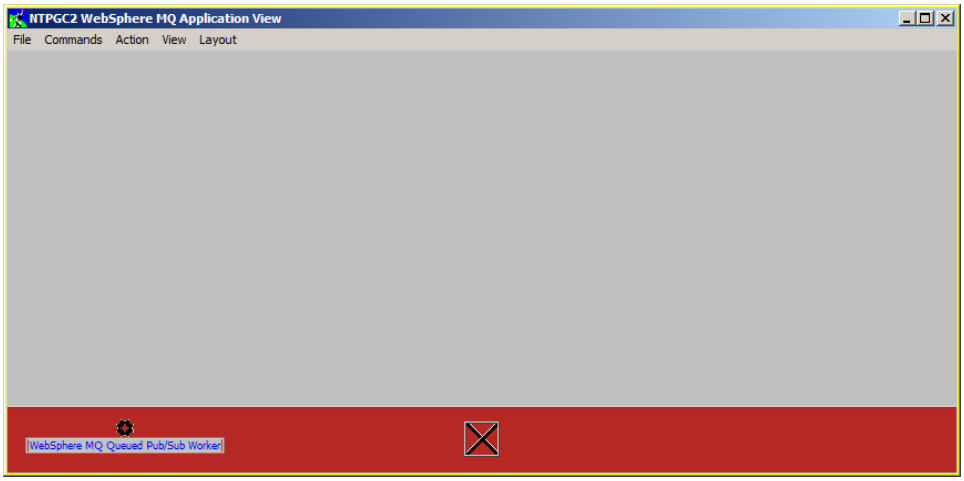
The application view lets the user see at a glance the status of the various applications which have, at one time or another, connected to the Queue Manager.



It shows the applications and the queues they access. The arrows in the links indicate the direction of message travel. If a link is currently active then it is shown in bold.

What is perhaps clear from the outset is that everyone will have a different way in which they want to lay out their applications. As such there is no 'standard' diagram that you modify, instead each application and queue that you care about needs to be positioned. The good news, of course, is that, once positioned, it's place is persisted across restarts. It is recommended that before you start to build your diagram you ensure that you have set-up your application values. The key thing to do is decide what objects you wish to display on the diagram and whether you are going to give multiple applications the same nickname. This can reduce the number of items you need to position on the diagram. For example, if you look at the diagram above you see that we have changed all the various MQ Pub/Sub applications to have the same nickname of 'WebSphere MQ Pub/Sub'. As a consequence rather than having to position 7 different Pub/Sub applications only a single Pub/Sub application need be positioned. Likewise you can set 'No Diagram' on any applications you have decided are not interesting enough to put on the dialog. You can hide records from the diagram using the application lists display and the 'Hide from Diagram' menu. This allows you to hide many items all with a single click. Once you feel you have configured the applications as you want them then you should display the application view.

When you initially display the application view you will get a window that looks something like this:



The diagram is essentially blank and a red bar is displayed along the bottom of the window. This red bar is known as the 'Unpositioned Bar'. The bar is displayed whenever there is an application or queue which has not yet been positioned. Applications

are shown on the left and queues are shown on the right. All you need do is move the applications and queues to the place on the window that you would like to see them. If you decide that an application or queue should not be shown then you can drag it to the cross in the centre of the unpositioned bar. It is recommended that you position any queues first if you can. By giving preference to queues you are more likely to leave enough space in the diagram for all the definitions. As you move the queues a link line will be shown to the application which uses the queue which again helps you to position the queue in an appropriate place.

10.7.1. Moving and Selecting Objects

At any time you can re-arrange any of the objects on the diagram by selecting them and moving them accordingly. By holding the <Ctrl> button when you click you can select multiple objects. In addition you can select all objects in a box by holding the <Ctrl> button as you click on the background. Drag the box to enclose the objects you want to be included. All selected objects will be moved as a single block.

When you move an object it may 'snap' into position. Snapping helps align the objects in the display and is set in the Preferences Dialog. You can choose between 'No snap','Snap to Grid' and 'Snap to Object'. Snapping to grid will try to snap any objects as they are moved to fixed grid of positions. Snapping to object will try to align the object to any other objects which are 'close'. When you are moving multiple objects you will notice that each object may snap at different times. Often this is what you want since it maximises the amount of object alignment however it can appear disconcerting at first. At any time while moving an object you can temporarily suspend any snapping by holding the <Alt> button.

You can also 'snap' the object to it's starting position if you wish. For example, suppose you wish to move an item left or right without any vertical adjustment. You can do this by pressing <Shift> as you make you move. The item will tend to stay on the same axis. However, if you move the item too far away from the object axis then the actual position with be honoured and the axis snapping is ignored.

Once you have positioned all the objects to your satisfaction select the 'Save Layout' menu option.

10.7.2. Application View Menu

Many of actions available in the application view have already been discussed in the Network View. However, there are a few others which are specific to applications which are explained below:

Action	Meaning
Refresh Application Data	Selecting this menu will cause MO71 to refresh its data about this Queue Managers' current application connections
Refresh All Data	Selecting this menu will cause MO71 to refresh all its data about this Queue Managers' objects
Hide Unpositioned Bar	Selecting this menu will hide the unpositioned bar. Note that with this selected the user will not get any feedback that some of the applications or queues are not shown on the window.
Show Only Active applications	Selecting this menu will mean that only active applications, and the queues they used, are shown on the diagram.
Show Queues	This menu option chooses whether queues are show or not.
Show Resolve Queues	This menu option chooses whether resolved queues, such as queues pointed to by alias queues, are shown or not.
Show Temporary Queues	This menu option chooses whether temporary queues should be shown on the diagram. Due to their dynamic nature normally temporary queues would not be shown. Note that any state, such as usage and diagram position, is not saved across restarts of MO71.
Show Only In-use Queues	This menu chooses whether only queues currently in use are shown.
Show Only Active Links	This menu chooses whether only links between objects that are currently in use are shown.
Save Layout	This menu option saves the current positions.
Restore Layout	This menu option restores all the objects positions to their last save point.

Chapter 11. General Actions

11.1. Dates and Times

Dates and Times in MQ, such as an object's last altered time, have a number of problems.

- **The time is displayed in two separate fields.**
  This makes sorting in lists very awkward because sorting by either field doesn't yield a time sorted list.
- **The format is fixed and not always easy to read.**
  For objects it is something like : Date : 2018-06-04 Time: 09.36.38
  For messages it is something like: Date : 20190528 Time: 06431570

  Of course everyone has their own preferred way of dates and times being displayed. One of classic confusions is whether dates are YYYYMMDD or YYYDDMM however it goes far beyond that. MO71 allows you to have pretty much whatever format you prefer.
- **Timezones**
  Many MQ installations span multiple time-zones. This can cause confusion since it is very difficult to take these time-zones into account 'on the fly'. Sorting of dates and times in a multi-Queue Manager list also does not naturally sort itself in the order that things actually happened – just in the order that MQ reports the time. For example, looking at the start time of a channel can vary wildly on the sender and the receiver although clearly they started at roughly the same time.

MO71 removes all of these problems by allows you to referring to a combined date/time field rather than the individual date and time fields. These original date and time fields can still displayed if you wish but you now have the added field which will display the date and time in any format you wish as well as always display the time in your local time-zone.

There are two main configurations you need to make:

1. In the 'Time' tab of the Preferences Dialog you specify the format of time you wish to see.
2. In the 'Time' tab of the Location Dialog you specify the time-zone of the Queue Manager.

The other choice you can make is whether these combined times should be shown to you in your local time, or as the Queue Manager sent them to MO71. This is done by checking the 'Show local times' checkbox in the the 'Time' tab of the Preferences Dialog. Even if you choose to display the field in the original time-zone the combined date/time field will still sort correctly with regards to other Queue Managers. In a list, the combined Date&Time fields have their own type of column filter. This filter allows you to filter the list in various ways such as 'until a particular time' or 'from a time for 2 hours' etc. For an explanation of the options available please refer to 'Date&Time Column Filters' on page 30.

11.1.1. Recent Dates

MO71 also allows you to highlight dates which occurred recently. Very often, if you are tracking down a problem, it can be a recent change which caused an issue. It can be useful therefore to highlight dates which happened recently. MO71 allows you specify a different time format for times which happened either in

- the last 24 or 48 hours
- or Today or Yesterday

In the first case you are only interested in the elapsed period of time. In the latter case you are interested in the demarcation of days. You might for example have a format which says **“Today %TT”** which, rather than having the date, makes it very clear that this particular thing happened today.

11.2. Units

Many of MQ attributes have an associated unit. Certainly for the novice MQ user, and even for the more seasoned professional, it can be difficult to remember which unit a particular object uses. Is an attribute specified in seconds or milliseconds etc etc. By using units the actual value can be more obvious. For example a maximum message length of '4194304' is less understandable than '4 Megabytes'.

How and where Units are used in MO71 is controlled by the 'Units' tab of the Preferences Dialog described on page 209.

11.2.1. Types

The following is the list of Units used in MO71.

Category	Unit Type	Full Names	Short Names	Abbreviated Names	Special Value
Time	**Hours**	Week Day Hour	Wk Dy Hr	w d h	999999999
	Minutes	Week Day Hour Minute	Wk Dy Hr Min	w d h m	999999999
	Seconds	Week Day Hour Minute Second	Wk Dy Hr Min Sec	w d h m s	999999999
	Tenths	Week Day Hour Minute Second Tenth	Wk Dy Hr Min Sec Tenth	w d h m s Tenth	-1
	Milliseconds	Week Day Hour Minute Second Millisecond	Wk Dy Hr Min Sec ms	w d h m s ms	999999999
	Microseconds	Minute Second Millisecond Microsecond	Min Sec ms us	m s ms us	999999999
Size	**Megabytes**	Gigabyte Megabyte		GB MB	
	Kilobytes	Gigabyte Megabyte Kilobyte		GB MB KB	
	Bytes	Gigabyte Megabyte Kilobyte Byte		GB MB KB B	
	4K Blocks	4K Blocks		4K Blk	
Percentage	**Percentage**	%	N/A	N/A	

For a list of the MQ fields and the units they use, please see https://mqgem.wordpress.com/ibm-mq-units/

11.2.2. Style

The style setting for a unit controls how that value should be shown. Let's take as an example 95 seconds. There are a number of different ways one could express this time. As 95 seconds, as 1 minute 35 seconds, as 1.58 minutes. The style controls which of those should be used for each Unit type.

Style	Example
Auto	Let MO71 try and display the value using the simplest format. This format will look for multiples of Units but also powers of 10. For example on a bytes field, 4194304 bytes is displayed as 4 MB but 4000000 is displayed as 4,000,000 B
Decimal	Certain fractional values are allowed. For example, **3.5 minutes** but not 3.6 minutes
One-Decimal	Always displayed as a single decimal value eg. **3.62 minute**. Note that this format may be inexact but it does generally use the least screen real estate.
Parts	Display the number as the sum of the individual unit parts. For example **1 week 3 days 5 hours 12 minutes**

The display style can be set by Unit category and fields of that category will mostly be displayed in that style. However, there are some fields whose values tend to change frequently. These are output fields which are usually measuring something – for example the average time a message has been on a queue or the number of bytes sent down a channel. These values change so frequently that showing their exact value is usually of little interest. For example, showing that a channel has moved '14 Megabytes 31 Kilobytes 435 Bytes' is far too exact if the value changes a second later. Instead what you want is for these fields to be displayed in an easy to consume approximation. These 'transient status' fields, as we call them, will by default be shown in 'One-Decimal' format. The above channel might just say '14.1 Megabytes' for example. However, if you prefer you can change this in the 'Units' tab of the Preferences Dialog. Which fields are consider transient are shown in the link given in the Types section above.

11.2.3. Location

There are various places where Unit values could be applicable. These are set in the 'Units' tab of the Preferences Dialog described on page 209. Essentially for each location you can specify whether units are used and whether the full, short or abbreviated Unit names should be used.

11.2.4. Data Entry

Not only are Unit values displayed but they can also be entered in at the entry field. Any format can be used regardless of the Unit display setting. It can be much easier, and less error prone, to enter 10MB rather than 10485760.

So, any of the following can be used:

- 3.5MB
- 2MB 100KB
- 300min 30s

Not that if a value is entered which does not result in an integer value, such as **3.24 MB** then an error message such as

“Not a whole number (3.240000 x 1048576)” will be displayed.

11.2.5. Special Value

Some units have a special value, for example 999999999 seconds, which is used to signify 'infinity'. Such values will still be displayed as 999999999 rather than the resulting Unit string.

11.3. Exporting Definitions

Most of the command dialogs allow you export the displayed data. This can be very useful for a number of reasons, most notably.

1. You want to embed the information into some other document
   In this case the information needs to be exported in human readable text form.

2. You want to capture the information so the object can be defined on another Queue Manager
   In this case you really want the data exported in MQSC ‘define’ format.

On most of the dialog displays an export option is available in the pop-up menu. The dialog displayed allows the user to select the type of export, a file name to export to if required and control what type of data should be exported. The values chosen will be saved for this type of dialog and restored on application restart. Note that depending on the type of export not all fields may be available.

- **All Fields**
  This option is only available for the main object list dialogs. It controls whether just the fields contained in the list are exported or whether all the fields of the object should be exported.

  Normally this option would be unchecked however if you are exporting MQSC definitions then this may mean that you would get an incomplete object definition. In such a case you may wish to consider checking this option.

  Note however that the behaviour of an 'All Fields' export is slightly different from a 'Normal' export.

  - In order to export all of the object fields MO71 must go to the Queue Manager and re-download the object definitions. The export is therefore done asynchronously.

  - 'All Field' exports are done as the object definitions are returned from the command server rather than directly from the list. As a consequence the order of objects is the sequence returned rather than the sorting order of the list.

  - Similar to the point above, if exporting multiple Queue Managers then the objects will be grouped by Queue Manager rather than the grouping/order they may be in the list. This is useful, particularly for MQSC output, where the definitions may be interspersed.

  - If exporting in 'Text' format the objects will be exported as a set of objects rather than as a single list containing all fields.

  Remember that if you want greater control over the output of the export, such as field or sort order, then you can just add all fields to the list and then do a normal, rather than an 'All field', export.

- **Export Type**
  As described above this option allows the user to choose between straight text format, comma separated values (CSV), XML, JSON or ‘MQSC DEFINE’ format. For 'text' and 'MQSC' export types you can choose to to have a title line written at the start of the data, this is set in the 'Export' tab of the “Preferences Dialog”.

  There is one other export type which is 'Datapoints'. This is available only in the monitor status dialog. It requests that rather than exporting the contents of the dialog the actual monitor data points are exported instead. They are exported in CSV format suitable for loading into a spreadsheet or post-processing in some way.

- **Export Selection**
  This is only available in list dialogs. It allows you to choose to export only those entries that are currently selected or the entire list. Note that any current filter will be in effect, and you can use the 'Export Filter' and 'Dialog Fields' options (see below) to ensure a reader of your exported report is aware of this. Also note that only entries shown in the list can be exported.

- **Export to File or Export to Clipboard**
  A choice is presented as to whether the data should be exported to the file name given below or directly to a clipboard for pasting into another application.

- **File Name**
  The name of the file the program will write to. The export file format can contain character inserts which are described in chapter 24.8 File Name Inserts on page 228.

- **Overwrite**
  When checked the program will overwrite any previous version of the file. Without this option checked the program will prompt the user for an overwrite decision if the file already exists.
- **Append**
  By default the program will overwrite any existing file. However, with the append box checked the data is written to end of any existing information in the file.
- **Always Header**
  This options is used to specify whether a header should always be written when appending to a file. If unchecked the header will only be written at the top of the file.
- **Default Objects**
  Specifies whether the default objects should also be exported.
- **System Objects**
  Specifies whether the system objects should be exported.
- **Default Differences**
  Specifies whether only the differences from the default definitions should be exported. This is particularly useful when exporting in MQSC mode since you get the minimum command that is needed to re-create the object.
- **Field Title Units**
  Specifies whether any field Units should be included in the field names on CSV exports.
- **Field Value Units**
  Specifies whether any field Units should be included in the field values on CSV and XML exports.
- **Dialog Fields**
  Specifies whether the Dialog fields themselves should also be exported.
- **Export Filter**
  Specifies whether the filter string should be exported.
- **Blank Fields (only certain export types)**
  This options means slightly different things depending on when it is being used.
  - **Export of an object (such as a single object or an 'All Fields' export from a list)**
    This option requests that fields that would normally not be shown, because their value is empty, be also included in the export.
  - **Export from a list (only CSV)**
    It specifies that columns from the list that are completely blank and would normally be excluded both from the display and from the export to the file, are to be included in the CSV file export. This makes it much easier to append or merge multiple export files since the number of output columns remains constant.
- **One Line (MQSC only)**
  For an MQSC export this option controls whether the MQSC should be output in a single line. A single line allows multiple objects to be compared easily since each attribute is given it's own column.

11.3.1. Export MQSC format

When data is exported only the data currently contained in the dialog is exported. This can be really useful when exporting text and only certain fields of an object should be exported. However, care should be used when exporting lists in MQSC format. Because only the data in the dialog is exported it is quite possible to export an invalid MQSC definition. To ensure a complete definition is exported either

1. Make sure that the 'All Fields' checkbox is selected in the export dialog
2. Use *'Alter List'* in the list window first to ensure that all the fields you need are in the left hand window i.e. in the **List Contents**. Select *'OK'*. This will retrieve all information about all objects. You can now export the objects from the list knowing that all data for all fields is available.
3. Alternatively, if all objects of a particular type are required you can export them from the location dialog directly. Please see “Location Dialog” on page 214 for more information.

If you are exporting an **MQEV** dialog in MQSC format then the default file type will be .mqx rather than .mqs indicating that you need to run the script file in **MQSCX** rather than just runmqsc. To be clear **MQSCX** can process either .mqc or .mqx files but you need to ensure that you use **MQSCX** with an .mqx file.

11.4. MQ Maintenance

MO71 will make a check to the IBM web site (https://www.ibm.com/support/pages/recommended-fixes-ibm-mq) if configured, to see what are the latest MQ Maintenance levels are available for your Queue Managers. This information can be used in two ways:

11.4.1. Display latest MQ Maintenance levels

If you just want to have a quick check of the latest MQ Maintenance levels available then select '***Show IBM MQ Maintenance Levels...***' from the main application File menu. This will display a simple dialog showing the latest maintenance available for the IBM MQ products. The information is obtained from the Web page above so clearly this feature requires and an available internet connection.

11.4.2. Check latest MQ Maintenance for a Queue Manager

Of course what one is often interested in is 'is there a later maintenance for the Queue Manager I am using?'. After all it is generally advised to try and stay current with Fix Packs as they are released. This can be quite a mission to keep track of, especially if you have large numbers of Queue Managers. With MO71, it is easy.

All you need to do is bring up a 'Queue Manager List...' dialog. Now in that dialog select 'Show Queue Manager List' and select the Queue Managers you are interested in. Next you would ensure that 'Maintenance Level' is in your displayed lists of fields. If it isn't then select it from the 'Alter List...' dialog. Now, all you need to do is press 'Refresh'.

MO71 now shows you all your Queue Managers and the maintenance column of the list will contain a value if there is later maintenance available for that Queue Manager. In other words, if you are already at the latest version then the entry for that Queue Manager will be blank. This means that you need only enter a filter of 'mqmaint' and MO71 will just show you those Queue Managers which are behind on their maintenance level. If for some reason MO71 can not access the MQ maintenance web page then the value 'Unknown' will be shown.

11.4.3. Network View Problem Analysis

MO71 will also check whether a Queue Manager is at the latest maintenance level when you analyse your MQ Queue Managers in the Network View. If a Queue Manager is back-level then MO71 will report this in the Queue Managers list of problems.

11.5. Splash Screens

Splash screens are windows displayed to the user, triggered by a program event, to notify the user of information. What is displayed to the user is entirely up to you. You can display text, pictures, even a web page! Splash screen settings are configure in the 'General' tab of the 'Preferences Dialog'. There are currently two events that can cause a Splash Screen; when MO71 initially starts and when MO71 connects to a Queue Manager location. Splash screens can be used for a variety of reasons.

- **To give users general information about the system and ownership**
- **To notify users of the type of Queue Manager**
  For example, you might have a splash screen on all your production Queue Managers to alert the user that this is a production system and that any changes should be treated with care.
- **To notify users of outages and maintenance schedule**
  Suppose you need to bring a queue manager down for maintenance. You could have a splash screen which would tell any user why the system is down and when it is likely to come back. This could even prevent phone calls in to your department of people complaining that they can no long connect to their favourite Queue Managers!

Because of these different scenarios, the configuration of the Queue Manager Splash screen can be given in two parts. We have the notion of a 'Normal' Splash screen and a 'Specific' Splash screen. The idea is that the 'Normal' splash is the screen that is normally shown to the user – this would just give general information about the Queue Manager, for example its type (Production/Test/Development), its maintenance schedule and its owner. Then we have the 'Specific' Splash screen which is shown to tell the user of specific, more unusual events. For example that the Queue Manager is currently down and the expected time it will be restarted.

Whenever MO71 is about to display a Queue Manager Splash screen it will look for a specific Splash screen first and only if it doesn't find it will it display the Normal Splash screen. Of course you may decide that you don't wish to use one or other type of Splash screen and that is perfectly fine. If there is no Splash screen of that type then MO71 will ignore it and use the other one.

Splash screens are shown to the user in response to a user action. They will not be shown just because a Queue is auto connecting but as the result of the user issuing a command on a dialog. If the Splash screen is shown then the command will be discarded and

the user will need to reissue it. Further the command will not be accepted until the Splash screen is closed. This is to try and force the user to read and accept the Splash screen before issuing commands against this Queue Manager. Once a Splash screen has been shown for a Queue Manager it will not be shown again until connection is ended.

Configuring a splash screen is very simple. All you need to do is put the location of the Splash file in the appropriate entry field. The splash file can be a number of different formats.

Splash Type	Example	Why that type?	Notes
Text file	c:\MO71\splash\startup.txt	Ends **.txt**	
RTF file	c:\MO71\splash\startup.rtf	Ends **.rtf**	Pictures are currently not supported.
HTML file	c:\MO71\splash\startup.html	Ends **.html** or **.htm**	JavaScript is not currently supported
Remote Web Page	http://www.mqgem.com/MO71/startup.html	Starts **http://**	JavaScript is not currently supported

In each case MO71 will attempt to show the contents of the file to the user in s window. Bear in mind that the intention is that these windows of information should be kept small and concise. If necessary you can have links to other information which can be displayed in a browser.

If a file does not exists (or is not accessible) then no error message is displayed, it is just that the user does not see a splash screen. In this way the administrator can control, centrally, whether a splash screen is shown or not. For example, suppose the department has a central file server. You could configure all of your MO71 users to use a splash startup file of 'h:\central\MO71\splash\startup.rtf' and, let's say, normally this file doesn't exist. However, periodically you could add the file, perhaps to tell users about a maintenance window the following weekend, and so the next time the user starts MO71 they will be told about the scheduled maintenance.

11.5.1. Configuration

Configuration of the Splash screens is done in the General tab of the Preferences Dialog described on page 193. On this dialog there are three entry fields which allow you to enter a Splash screen of each type. At this point you may be wondering how you can enter a single value in the Preferences Dialog which produces a different result for different Queue Managers. The answer is that you can use inserts in the file name.

Two inserts are available:

Insert	Example	Effect
%m	c:\MO71\splash\%m.rtf	The %m insert will be replaced by the location Queue Manager name
%g	c:\MO71\splash\%g.html	The %g insert will be replaced by the locations Queue Manager group name This makes it easy to have a different Splash screen for each logical group of Queue Managers

There is a test button to the right of each setting. This allows you to quickly verify that the Splash screen file is specified correctly and the contents is shown. For the Queue Manager splash screen dummy values of 'mqgemqm' and 'mqgemgroup' are used for the Queue Manager name and group name inserts. If you wish to see what the Splash screen looks like for a particular location then you can press the 'Show Splash Window' button in the Options tab of the relevant Location Dialog described on page 216.

11.5.2. Location and Size

By default Splash Screens are shown in the centre of the monitor and are half the size of the screen. The user can re-position and resize the window but these new values are not remembered by MO71 since the next time the window is displayed it could contain completely different content. However, if required you can set the preferred size in the configuration. The way you do this is to simply add the notation **[*xsize*, *ysize*]** to the end of the file name.

For example **c:\MO71\splash\%g.html[1000,600]** requests that the window is initially sized at 1000 pixels by 600. If a size is set to zero then the default size for that dimension will be used. If a size is given that is larger than the size of the screen area then the screen are size is used. So, for example, **c:\MO71\splash\%g.html[99999,99999]** will display the window and completely fill the monitor.

11.6. Master Configuration File

Many MO71 installations consist of a central department of core MQ experts and a number of satellite departments who are aware of MQ but have wider responsibilities. In these installations it can be handy to have the basic MO71 configuration done centrally in terms of add location definitions and ensuring that their connection parameters, such as the channel definition, are correct. However, you still want the users of MO71 to be able to have their own settings, such as filter, dialog positions etc, remembered across restarts so you don't want to be too prescriptive with the configuration file you give them.

The solution is to use a master (or joint) configuration file. This essentially allows you to import central location definitions from a master configuration file each time MO71 is started.

There are two formats of master configuration file available

1. **MO71 Configuration file**

   ```
   mqmonntp -j h:\MQ\central\MO71.cfg
   ```

2. **XML Configuration file**
   The master configuration can be provided as an XML file. This can either be an XML file such as an MQ Explorer configuration or an XML file that has been independantly generate of the same format.

   To identify that the configuration file is XML then just ensure that the file type of the file is **.xml**
   ```
   mqmonntp -j h:\MQ\central\MO71.xml
   ```

This command tells MO71 to merge any location definitions it finds in this central file with its own definitions. Any new locations are added and any existing locations are modified to match the central file's connection parameters. Note that not every parameter associated with the location is migrated, they are mainly those related to the actual connect.

The list of fields migrated are:

- Location
- Group
- Network Names
- Via Qm
- Reply Queue
- Reply Prefix
- Command Queue
- Server Queue
- CCDT URL
- Client and any Client Channel Definition
- MVS
- Auto Connect
- 'Userid and Password' connect option
- 'Security Exit Only' connect option

11.7. Licence Command

In most MO71 installations the administration of the software licences is done centrally and the licence is copied out to the relevant machines where MO71 is actually run. To make this easier and more automatic MO71 allows the user to specify a command which can run on the machine that will do the copy for you, this is known as a licence command. MO71 will run the command if it detects that there is no valid licence, or if the current licence has less that 30 days left to run.

The command itself is configured in the General tab of the Preferences Dialog. For example we could specify something like:

d:\nttools\getMO71licence.cmd %d

The **%d** is replaced by MO71 with the current configuration directory.

The contents of the command file itself can be whatever you choose but in general it would be something pretty simple such as a copy command. For example:

```
copy h:\licences\MO71\mqgem.lic %1
```

So, with this combination whenever MO71 invokes this file a copy of the latest licence from **h:\licences\MO71** will be copied to the MO71 program directory. In other words, provided there is valid licence in this directory then MO71 will pick it up and start using it. Of course the command file could be more complicated. You could, for example, make an FTP call to retrieve the file from a central server. Or you could send an email to alert a user of the situation.

11.7.1. Command Inserts

There are two command inserts which can be used

Insert	Replaced with
%d	The configuration file path (the default location for the licence file)
%r	The number of days remaining on the current licence

11.7.2. Command File Exit Code

If the invocation of the command fails or it returned a non-zero exit code then MO71 will display a message box to explain the situation. To deliberately return a bad exit code use the **exit** statement. For example:

```
exit 12
```

Because of the way Windows processes program exit codes command files should not return an exit code of 259.

11.8. Printing

The main window and the object dialogs have a *'Print...'* menu item allowing the user to send the information in the dialog to the printer.

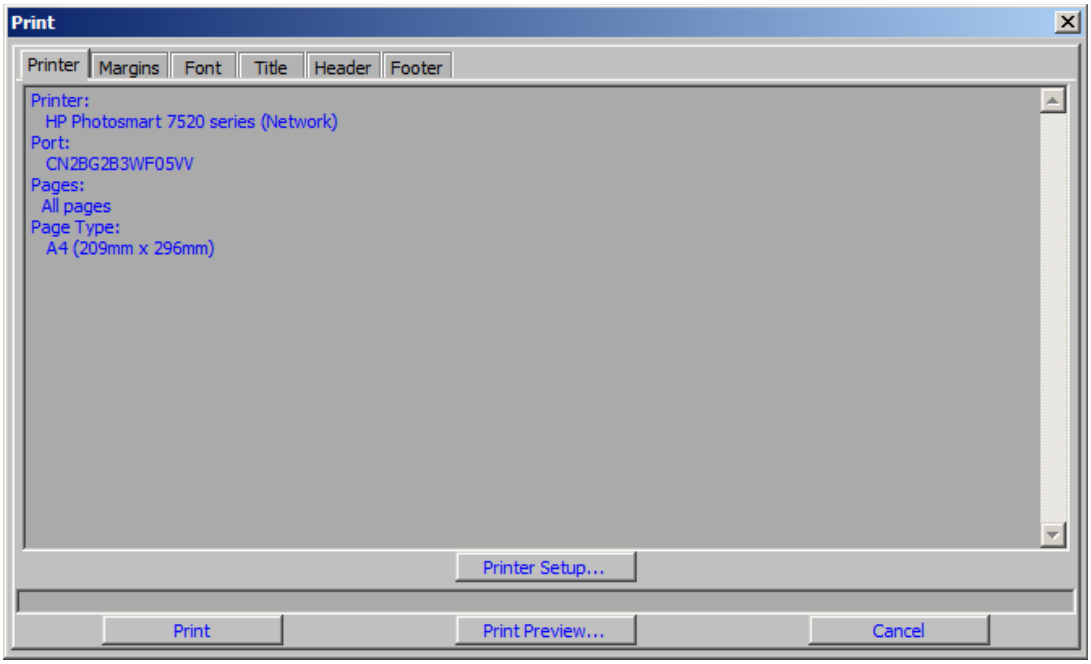


Figure 21: Print Dialog

Selecting the *'Print...'* menu item will present a print dialog box. This dialog displays the general parameters which will be used for printing.

- **Printer Details**
  Displays information about the selected printer and paper size
  To change the printer and the general values chosen you can select the 'Printer Setup...' button.
- **Printer Setup**
  Pressing this button will bring up the system printer dialog. The exact format of this dialog is system dependent. However, it should allow the user to select the printer, the paper size, the paper orientation, pages to print and probably a whole host of printer specific options.

  On some systems the button to apply the changes in the printer dialog is 'Print'. No printing will actually take place when this button is pressed. The printer values will be stored and returned to MO71. Only when the user presses 'Print' in MO71's print dialog will any data be sent to the printer.
- **Print Preview**
  Pressing this button will display a simple window showing what the print output would look like with the current settings. This allows the user to modify various values like margins, font size, paper orientation and line spacing to ensure the data fits neatly on the page. As values are changed in the main print dialog the preview will change immediately showing the effect of the value[26]. The print preview menu allows the user to select how many pages to display and what the size of each page is relative to the size of the window.
- **Print Button**
  Only when this button is pressed will any data be sent to the printer. The button will change to a cancel print button which allows the user to cancel the print operation if necessary.

The rest of the options are selected in various Tabs.

[26]On Windows 95,98,Me the preview window is a monochrome bit map so the print colour will not be reflected in the displayed output. This is due to a bit map size limitation on these platforms.

11.8.1. Margins

This Tab will show graphically the printable area of the page. There are two types of setting:-

- **Margins**
  These fields allow the user to set the vertical and horizontal margins on the paper. Values are in millimetres.
- **Overlap**
  When printing large dialogs it could be that to print all the information the output will need to be spread across multiple pages. In this case the program prints each page as though it were part of one large logical piece of paper. After printing the pieces can then be placed side by side to see the whole output.

  If multiple page output is required the Overlap values tell the program how much of an overlap is required between each page. An overlap ensures that a relatively seamless join can be made between the multiple sheets. Overlap values are specified in millimetres.

11.8.2. Font

- **Choose Font**
  This button will display a dialog allowing the user to select the required font for printing. This font is not related in any way to the fonts selected for dialog screen display. The currently selected font is displayed in the box to the left of the Choose Font button.
- **Line Spacing**
  This field allows the user to change the spacing between successive lines in the print output. The value is expressed as a percentage of the currently selected font height. For example a value of 50 would leave a gap of half (50%) of the height of the font between each line.

11.8.3. Title, Header, Footer

These Tabs allow a user specified string to be printed along with the printed text. The title is printed once, across all the top most pages. Headers and Footers are printer at the top and bottom of each page respectively. It is useful to be able to put values rather than just a known string at the top and bottom of the page. For example, you may wish to print out the date and time as a footer. To allow this, inserts may be specified in the text. For example you might want to add the page number with something like:

```
Page %n
```

For a complete list of the inserts available please see Text Inserts on page 227.

11.8.4. Printing Lists

When printing lists the application will print titles for the fields at the top of the upper pages. The titles shown will honour the ‘List Titles’ choice made in the list dialog window. However, there can be a slight difference with the ‘Auto’ setting. The ‘Auto’ list title setting tells the dialog to use short or full titles depending on whether there is room on the dialog. In printing there isn’t quite the same concept of ‘room on the dialog’. For printing therefore ‘Auto’ means

- **Full title** if field data length is greater than full title length
- **Short title** if full title is greater than the all the data lengths

Chapter 12. Queue Manager Monitoring

There are three types of Queue Manager monitoring:-

- **Direct Connection**
  If MO71 is directly connected to the Queue Manager either via local bindings or via a client connection then a message is sent to the reply queue and then immediately got back again.
- **Normal monitoring**
  A program on the receiving Queue Manager should send the monitor messages back to the monitor program. In other words, it should use the Reply Queue and Reply Queue Manager of the monitor message. It must set the Reply Queue and Reply Queue Manager to its own values. In addition, it must change the first character of the message from 'm' to 'r'.
- **Loop back**
  In loop back monitoring the monitor queue at the remote location must 'point' back to the Reply Queue defined in the local Queue Manager's location settings. Using this method it is not necessary to have any code other than the base IBM MQ product on the target system which allows very simple monitoring of any MQ platform. Since you must define a remote queue at the remote location pointing back to the Reply Queue you cannot use loop back monitoring if your reply queue is a model queue.

Using any of the methods will cause an MQ message to make a complete round trip between the monitoring program and the target Queue Manager. The application keeps track of when it needs to send a message and whether a reply is outstanding. If a message is not received within a reasonable time then the application will give a visible and, if required, an audible alert that a Queue Manager is not responding. In this way it is simple for an operator to check whether all of the Queue Managers and the Channels between them are functioning correctly.

12.1. Command Strings

A command string is a command you wish MO71 to issue to the operating system. The command is always submitted as a background process. The actual format of the command and what capabilities you have depend on the OS.

The three places where you can specify a command string are :-

- **In the location dialog**
  Here you are given the opportunity to give two commands. The idea is that, during monitoring of a remote location, monitoring will either succeed or fail. As it makes the transition from one to the other the appropriate command will be issued. This allows the user to have a downstream application which is driven from MO71 which takes some action when it is told that, for instance, monitoring to a remote location is now failing.

- **In the filter string using the system() function call**
  The system() function, in the filter, takes two parameters, an expression and a command string. If the expression is evaluated to TRUE the command will be issued. This allows the user to check for virtually any combination of object attributes and if they match a certain criteria then a command will be issued. Note that the system() function is executed for each item in the list returned. So, for example, if a filter of system(1,"myprog.exe") is used on a queue list then as many instances of myprog.exe will be started as there are queues in the list. You should use the expression to make sure that the program is only started as many times as necessary.

- **In an event definition**
  In the 'command' field of the event definition you can specify a command to be run if an event of that type is received.

12.1.1. Format

The format of the command string is whatever the underlying OS supports. Essentially it should be a program name followed by zero or more parameters which are governed by the program. The Administrator will treat everything up to the first space as the program name and pass everything else as parameters. There are two main formats:

- **Starting a program**
  pass the name of the program followed by any parameters, for example

  q -m QM1 -oWARNING -M "Queue Manager seems to be down"

- **Running a command or batch file**
  to pass a command to the command interpreter you must just start the command with the word 'cmd'. For example

  cmd copy d:\logs\mylog_txt d:\logs\archive

Remember that if your command contains spaces then you may need to enclose the command or that portion of the command in quotes. Ensure that you use the right quotes though, copying and pasting from a document are rarely the right characters. You are better off re-typing the command or at least the quote signs. If you need embedded double quotes in a string you can precede it by a backslash.

If you have trouble getting the command to work then a useful diagnostic aid is to pipe the output to a file. Often the problem is something simple like a parameter formatting problem. To pipe the output to a file issue a command along these lines:

cmd myprog -z XX > c:\temp\myprog.out 2>&1

Provided that the program writes error messages to a file then it should now be possible to determine the cause of the failure by looking at the contents of the file.

12.1.2. Substitution Characters

A substitution character is a way of simply modifying the parameters passed depending on the object generating the command. Substitution will occur anywhere in the command string. The following characters are supported:-

%l : will be replaced by the location name of the location

%m : will be replaced by the Queue Manager name of the location

%o : will be replaced by the object name in the list. If the command being issued is not related to a list then it will be ignored and removed from the command string

12.1.3. Examples

"myprog.exe"

"myprog -m OS2PGC1"

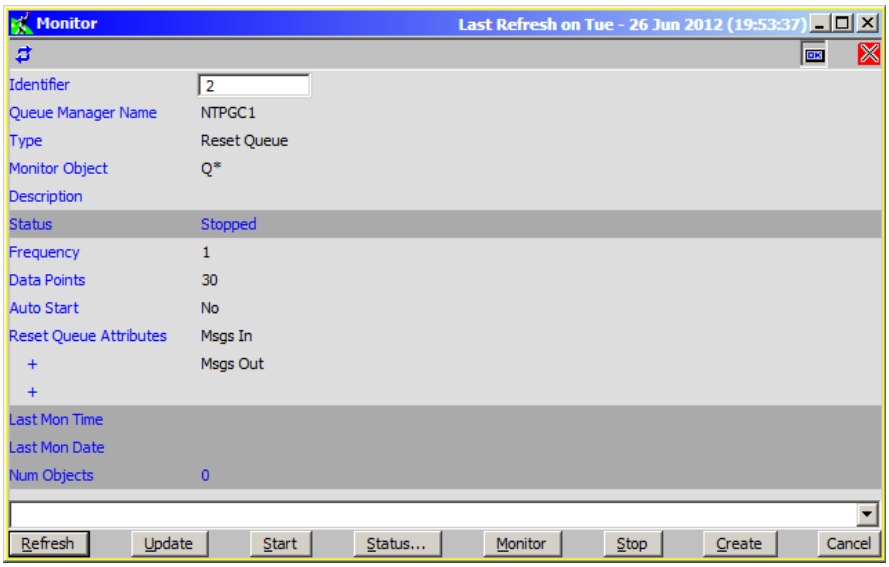
"myprog -t "Failure for object %o on Queue Manager %m"

cmd pageme bill "All seems well"

Chapter 13. Attribute Monitoring

Attribute monitoring allows you to monitor certain status values of Queue Manager attributes. The mechanism used it to periodically send a command to the command server and the responses are then remembered as a set of datapoints. Which objects, which attributes, the frequency and the number of datapoints retained are all specified in a monitor object. As always, I try and ensure that the code is as efficient as possible, however users must be aware that defining a monitor which monitors every attribute for every object at 1 second intervals could cause a significant workload on the Queue Manager.

The first step to monitoring an attribute is to create a monitor object. The dialog required is accessed from the main window under the 'monitor' menu. One is presented a menu something like this.....



13.1. Monitors

13.1.1. Monitor Fields

The following fields are available on a monitor dialog.

- **Identifier**
  Each defined monitor has a unique 'Identifier'. This is assigned automatically when the monitor is created so usually this field can be ignored. However, if you wish to retrieve the definition for a particular monitor you can do so by entering it's Identifier.
- **Queue Manager Name**
  This field defines which Queue Manager will be monitored. Wildcards are not supported.
- **Type**
  This field decides essentially what type of object is monitored. For example, Queue Status or Channel Status. It controls what command is issued to the command server. Note that not all commands are supported by all Queue Managers[27].
- **Monitor Object**
  The name of object which will be monitored. Wildcards are supported subject to the command being used above. In the example above all queues starting with 'Q' are being monitored using the RESET QUEUE command.
- **Description**
  A text description to help identification of the monitor.
- **Status**
  What the current status of the monitor is, for example, Running or Stopped.
- **Frequency**
  How frequently MO71 should poll for the data in seconds. Note that the most frequent monitor allowed is every second. Care should be taken with really frequent monitors however since it can cause a considerable number of messages to be generated.

[27]Care should be taken with RESET QUEUE. This command is unusual in MQ status commands in that, as it's name suggest, it RESETs the data. Issuing the command 'changes things'. So, only use this command if you are sure that no one else is relying on RESET QUEUE behaviour. Having multiple monitors issuing this command against the same objects will also give strange results.

- **Data Points**
  How many data points should be maintained in the cache. Once the cache for this monitor is full then as new data is gathered so old data will be discarded. Together with the frequency value, these two values control how much history of the data change is kept. For example, 60 data points with a frequency of 1 minute will show the last hour of activity.
- **Auto Start**
  In order to start gathering data the monitor must be 'Started'. This can either be done manually or the monitor can be defined as 'Auto Start' in which case the monitor will start as soon as MO71 itself does.
- **Attributes**
  This field allows you to specify which data items you wish to gather. For some commands more than one attribute may be specified, for example both 'Msgs In' and 'Msgs Out'.
- **Last Monitor Date/Time**
  These fields are output only and show the last time a monitor request was made.
- **Number Objects**
  This output field shows the number of objects for which data has been received for this monitor.

13.1.2. Actions

These are the action available to monitors.

- **Create**
  Clearly a monitor must first be created in order to do any monitoring. The act of creating a monitor will define a unique 'Monitor Identifier' which can be used to uniquely refer to this monitor object and all the responses received. Defining two monitors with exactly the same attributes will do just that, define two monitors. No attempt is made to detect duplicates or coalesce the definitions.

  The monitor object definitions will be saved when MO71 is ended. However, the monitor data points themselves won't be. You can 'export' the data to a file, as we'll see later, but MO71 will not automatically harden the data and present it again when MO71 restarts[28].
- **Update**
  If an update is made to a monitor object the effect is mostly instantaneous. It is not necessary to Stop/Start the monitor. For example, updating the Frequency of data retrieval or number of Data points kept.
- **Start/Stop**
  A monitor will only gather data while it is 'Running'. The monitor can either be started manually or defined as auto-Start where it will be started as soon as MO71 itself starts. If a monitor is started but fails, for example the Queue Manager is not available, then it will go into a retry loop.
- **Monitor**
  Selecting this option will cause a monitor request to be sent now rather than waiting for the frequency time. This can be useful if you've just fixed a problem or changed characteristic, for example 'cleared a queue' or 'started a channel' and wish to immediately see the effects on the monitor.
- **Status...**
  This action will show a list of all the attributes returned for this monitor object.

[28]This is something I have considered doing and may be done in a future version of MO71 if there is sufficient interest.

13.2. Monitor Status

When a monitor runs it will generate monitor status objects. There will be one status for each object and attribute combination. It follows therefore that each monitor object may have zero or thousands of monitor status values. The easiest way to look at the status values is to display the 'Monitor Status List'. This is accessed either by pressing the 'Status...' button on a monitor definition, which will show the status of just this one monitor. Or you can press the 'Monitor List Status...' menu in the main menu in MO71. By default this menu option will show a list of all the monitor status. The list contains various fields which can be used to filter which status values are returned.

13.2.1. Actions

These are the main actions which can be performed on the monitor status.

- **Start/Stop**
  You can start or stop the monitor directly from the status. However, bear in mind that you are starting/stopping the monitor itself not just one object of the status. All objects returned from the monitor command will be affected.
- **Graph...**
  The data points returned can be graphed at any point. Please see the next section for a description of what is supported. Selecting multiple entries in the list and then pressing the 'Graph...' button will display a single graph of all the data points. Further data points may be added to the graph as required.
- **Reset**
  If, for some reason, you wish to delete the data points from a monitor status entry but keep the entry itself then you can use this 'Reset' action.
- **Delete**
  You can delete the monitor status completely, which will also delete all the data points. Note, however, that if the monitor is still running and data is being returned for the same object then the monitor status may be re-created.
- **Export**
  If you select the 'Export...' menu option you can export the Monitor Status dialog in the same way as any other dialog. However, there is also a new export type of 'Datapoints'. Selecting this option and pressing 'Apply' or 'OK' will cause the data points themselves to be written to the output location in a command separated value list. This can be useful, for example, to load into a spreadsheet. From there you can perform calculations on the data or draw charts which are more flexible than the ones I provide.

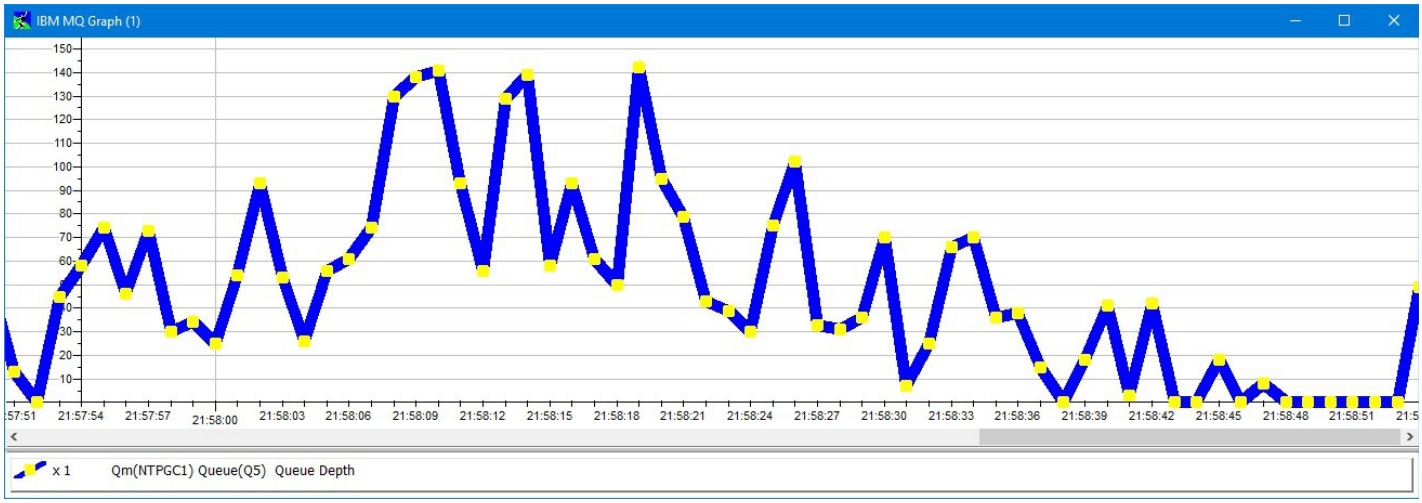
Chapter 14. Graphing

There are a number of different data items which can be graphed in **MO71**.

1. **Object Attribute**
   These were discussed in the previous chapter – see Attribute Monitoring. This allows you to explicitly monitor a whole variety of object attributes at your required intervals. From a monitor status display select 'Graph...'
2. **Activity Trace**
   If you are looking at Activity Trace data then you can ask to have the MQI calls graphed.
3. **MQEV Statistic and Accounting data**
   If you have an **MQEV** configured in your network that is catching Statistics or Accounting records then these records can be graphed directly in MO71. From the main menu under the 'Monitor' menu select 'Graph...'. This will initially display an empty graph. Graph lines can be added by pressing mouse button 2 and selecting data lines from the list.
4. **List Dialogs Fields**
   You can right-click any field in a list dialog ask **MO71** to display the distribution of the field as a pie-chart.
5. **Sample Graph**
   The Graph tab of the Preferences Dialog (described on page 212) contains button which will display a simple sample graph allowing you to play with the graphing options.

Many of sets of data may be graphed in multiple graphs at the same time but it can only appear at most once in any one graph. You can have a graph showing lines from difference monitor objects in the same graph but it is possible that the data does not line up ideally. Consider two lines, both with a frequency of 10 seconds but the time at which the monitor is made are not at exactly the same time. If you displayed the data as a line graph then it would not really matter. However, displaying the data as columns could mean that the data columns overlay on each other.

The graph window itself is split into two areas. The top area is the graph and the bottom area is the legend showing the colours assigned to each line.



In this example we see the depth of queue 'Q5' sampled every second. Each graph line has a line in the legend which gives a number of pieces of information:

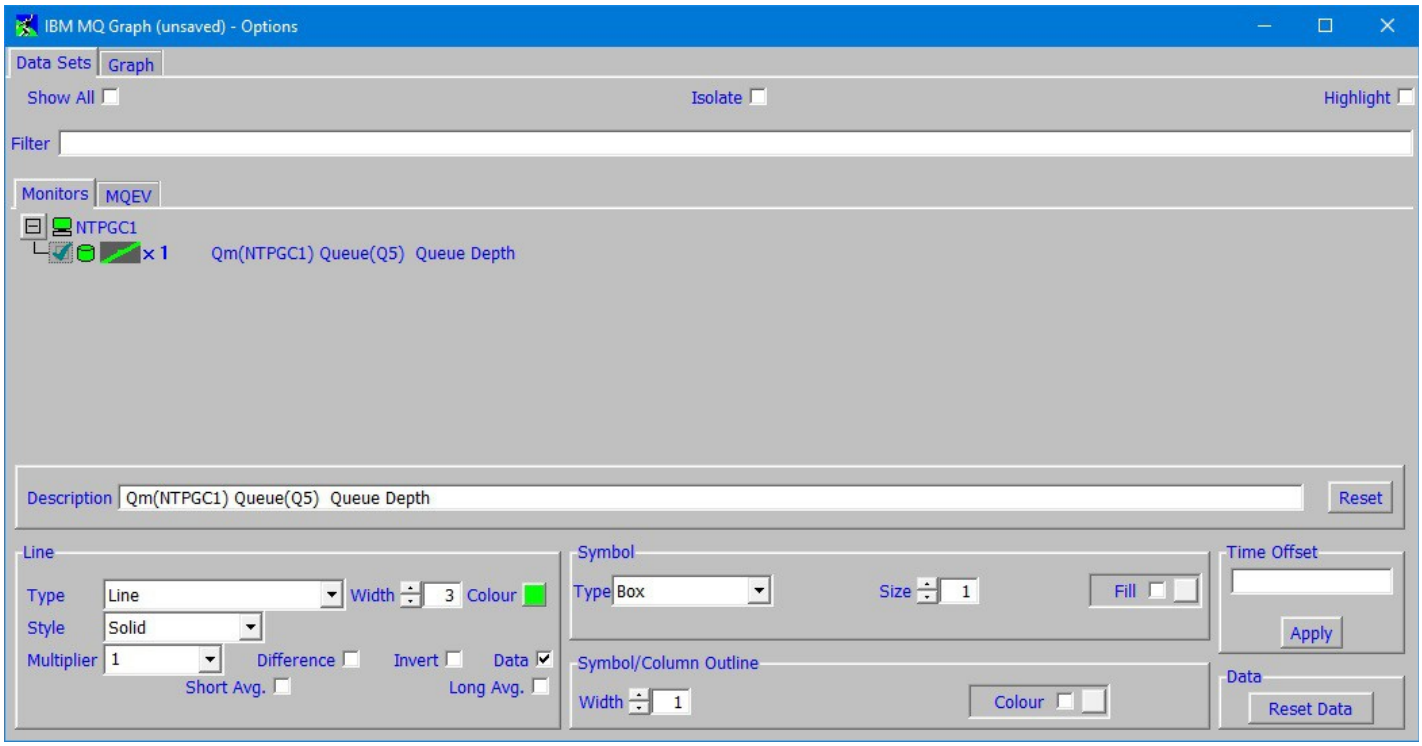
- The type of line, column, stack or pie segment and colour being used
- The symbol and symbol colour being displayed if it is a line
- The scale being applied to this line
- Information about the line content. In this example it's data about queue 'Q5' on queue manager 'NTPGC1.

The size of the legend is adjustable. In addition by double-clicking on the graph area the legend area can be removed completed. Similarly double-clicking on the graph area again will bring it back again.

Adding and deleting lines as well as adjusting the display options is done using the 'Graph Options' dialog which is displayed by pressing mouse button 2 somewhere in the graph window. Alternatively, double-clicking on a line in the legend will start the dialog and pre-select that line ready for adjustment.

14.1. Graph Options Dialog

The graph options dialog can, at first sight, seem quite complicated. However it is fairly straight forward.



The dialog has two main tabs.

1. **Data Sets**
   This tab allows you to Add/Remove lines from the graph and set how they should each be displayed as well as see what other sets of data are available to be displayed.
2. **Graph**
   This tab is concerned with global values which affect the whole graph.

14.1.1. Data Sets

This tab is split into three horizontal sections.

Top Section

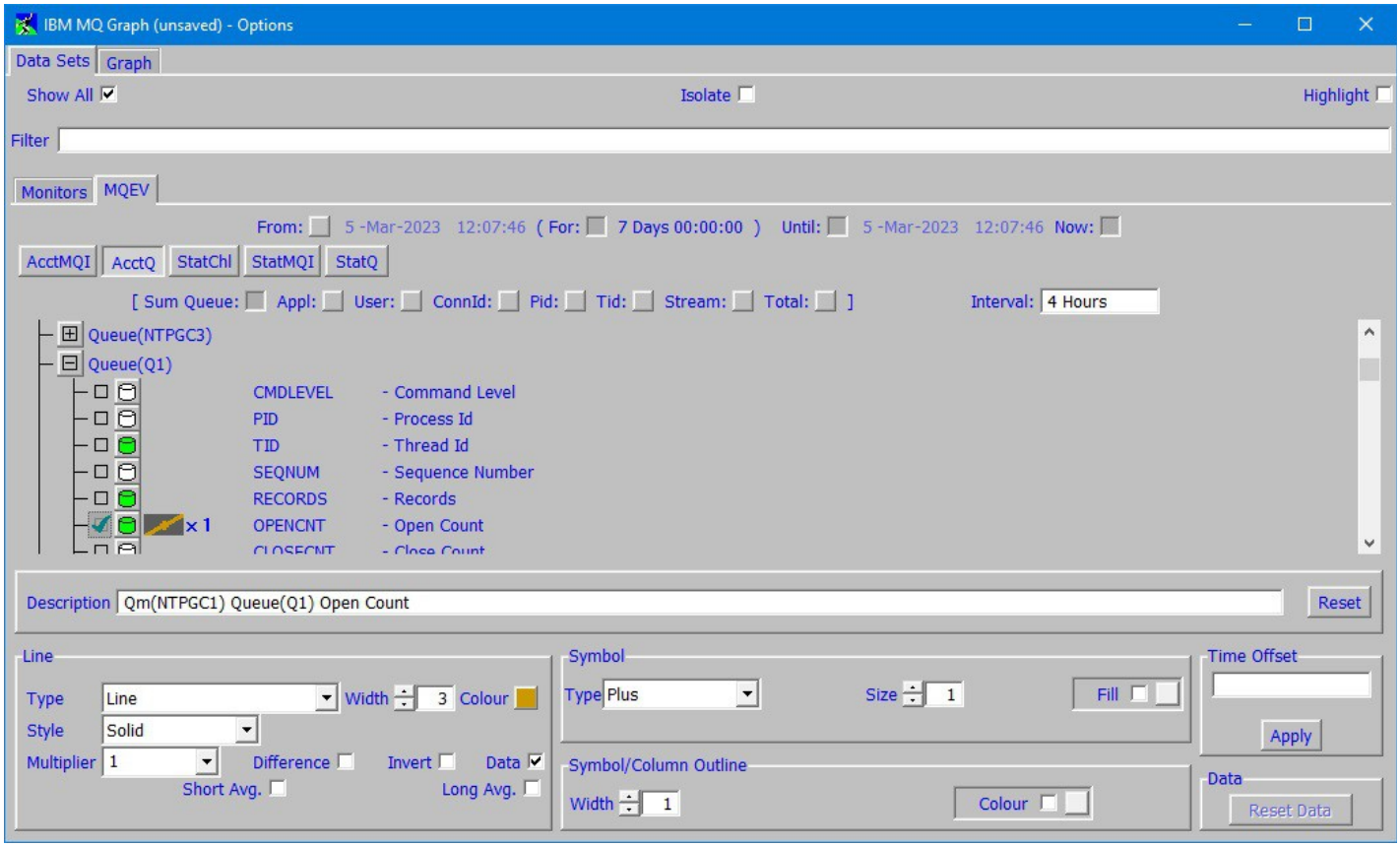
- **Show All**
  Selects whether the Data points list below shows all the available data points or just the ones currently in this graph.
- **Isolate**
  If selected then only the line selected in the Data point list is shown in the graph.
- **Highlight**
  If selected then the line selected in the Data point list is highlighted in some way.
- **Filter**
  A simple filter which controls which data sets are shown in Data set list below. Wildcards '*' and '?' may be used. This field is useful if there are many data points to choose from.

Middle Section

The middle section contains a number of tabs such as Monitors, Activity Trace, **MQEV**. The tabs allow you to see what data sets are available to be displayed on the graph. In the dialog shown above we have not specified “Show All” so MO71 will only shows those datasets which are currently selected for this graph. However, by selecting “Show All” the list of available data sets would include all the attributes which are currently being monitored.

The actual way that you interact with the tabs is a little different depending on whether you are interested in graphing the monitor data or whether it is the **MQEV** Accounting and Statistics data you are after. If it is the monitor data then it is just a question of selecting (putting a tick) next to the monitor data you want to include in the graph.

For **MQEV** data it is marginally more complicated but still pretty straight-forward. Let's consider the **MQEV** tab:



Here we have selected the 'AcctQ' tab. This is the tab for the Accounting Queue data. MQ provides five types of Accounting and Statistics data and each of these can be processed and stored by **MQEV** and we have to choose which of these we are interested in. But there are other selections we can make. Essentially work your way down the dialog:

1. **Set the timescale you are interested in**
   By default MO71 will assume you want to see the last weeks data. However, you can specify another date range if you wish. You can do this either by specifying the start and end time or by specifying just one of the end points and an elapsed time.

   If you select a time range that includes 'now' then the graph will automatically refresh itself at the MQ Accounting or Statistics interval for that Queue Manager.

2. **Choose the type of data by selecting the appropriate tab**
   You probably want to ensure that 'Show All' is selected so that all data available is shown.

3. **Choose your sum type**
   MQEV is quite flexible in how it can present the collected data. You may be interested in the data split by application name, userid, process etc. By clicking on the various sum buttons you can choose how the data is collated. You can select more than one sum option at the same time although 'Total' is only valid on it's own.

4. **Interval**
   The next field to consider is interval. By default **MO71** suggests that you want to see a data point every 4 hours. However,

you may wish to see data points every day or every hour. Bear in mind that it makes no sense to set an interval which is smaller than the interval that MQ actually generates the data. In other words **MQEV** can collate the data but it can not increase the resolution of the data it receives. Look at the Accounting and Statistics interval for your Queue Manager if you are uncertain what the collection rate is.

5. **Expand the Queue Manager you are interested in**
   Hopefully the container contains a list of unexpanded Queue Managers. If not then ensure that you have 'Show All' checked. If you still don't see a list of Queue Managers then ensure that you have configured **MQEV** for your locations. Please see 'MQEV Event Monitoring' for how to do this.

   When you expand the Queue Manager **MO71** will send a request to **MQEV** to determine what data is available for that period and with that collation. Each unique 'key' of data will be shown below the Queue Manager. The 'key' of the data will depend on the sum values shown above. For example, if you are summing by PID then each line will have a unique process id. If you are summing by Queue then each line will have a unique queue and so on. In the example above we are summing by default which is 'by Queue'. So, each key line is for a different Queue.

   You can cause **MO71** to refresh the list of key values at any time by pressing the 'refresh' icon.

6. **Expand the key line**
   So now you have the key lines you might be interested in. Expand the particular key you are interested in – in our case we have expanded queue Q1. **MO71** will then show you the fields which are available for this type of data. All you need do now is tick the left of the value line you are interested in. When you do so **MO71** will retrieve the values from **MQEV** and display them on the graph.

   The data will be automatically refreshed at appropriate intervals if the 'until' time of the time range is set to 'now. Alternatively you can press the database icon at any time to manually refresh the data.

Bottom Section

The bottom section of the dialog is concerned with setting how the data is shown on the graph.

- **Description**
  MO71 will make a description which attempts to describe the data being displayed. However, if you wish you can change this description to any text of your choosing. The **Reset** button will restore the description to the default **MO71** value.
- **Line**
  - **Type**
    This field chooses the type of line to display. Most are self-explanatory but Pie charts a worthy of further explanation, see the section below. If the <Shift> key is held down when the selection is made then the selection will apply to all lines in the graph. If the <Ctrl> key is held down when selection is made then all those lines having the same type value as the current line will also be changed to the new value.

Type	Effect
Column	Also known as a 'Histogram'
Column - Stack	Columns where each column is stacked on top of each other You can have up to 3 different stacks, A, B and C
Column - 3D	As for Column but with a 3D look
Column – 3D - Stack	As for Column - Stack but with a 3D look
Line	A line and possible symbol at data points
Pie (various)	See below
Scatter	Just symbols at the data points
Stack	Each line is 'stacked' on top of each other The display is a little like 'line' but each line with type 'stack' is plotted above the preceding 'stack' line. This display can be useful for showing the relative sizes of lines over time. The area below the line is also filled with the line colour.

  - **Width**
    The width of the line (if any).

- **Colour**
  The main colour used for this line. By default the first eight colours chosen for the lines will be those specified in the Preferences Dialogon the Graph tab. If more than eight lines are added to the graph then new colours will be generated using the first eight colours as the basis for the colours.
- **Style**
  The style of line to use (if any).
- **Multiplier**
  If you are plotting more than one line it is very possible that the two sets of values are not of the same magnitude. For example, if you were to plot ' curdepth of your transmission queue' and 'bytes sent down a channel' it is unlikely that the values are in the same numerical range. If might may more sense to divide the channel line by 1K (or more).
- **Difference**
  For non-pie charts this checkbox decides whether it is the data or the differences from the previous value which is plotted. With difference checked the plotted value is the different of the values divided by the number of seconds between the two values.

  Consider plotting the number of bytes sent down a channel. If you made a 'normal' plot you would just see a line getting higher and higher. Perhaps not very useful. However, by plotting the difference from previous value you effectively get a plot of the bytes per second rate.
- **Invert**
  Sometimes it's easier to visualise what is going on if two contrasting data points are plotted inversely to each other. Suppose we were plotting 'Msgs In' and 'Msgs Out' of Q1. By selecting 'Invert' for the 'Msgs Out' line it is more obvious what the relative pattern of data flow is.
- **Data**
  This checkbox controls whether the data line itself is displayed
- **Short Avg.**
  Checking this checkbox will tell **MO71** to display a short average line of the data line. A 'short' average is the average of the last 10 values.
- **Long Avg.**
  Checking this checkbox will tell **MO71** to display a long average line of the data line. A 'long' average is the average of the last 40 values.

➢ **Symbol**

- **Type**
  This field selects whether a symbol is added to any line types and if so, which one.
- **Size**
  This field selects the size of the symbol.
- **Fill**
  This field selects the colour used to fill the symbol, if not selected the symbol colour is used. If the symbol colour is not selected then the line colour is used.

➢ **Symbol/Column Outline**

- **Width**
  This field selects the width of the line used to draw the symbol.
  This value also controls how wide the average lines are drawn.
- **Colour**
  This field selects the colour of the line used to draw the symbol.

➢ **Time Offset**
This field allows you to move the current line horizontally. This allows you to line up two graph lines at the same position on the graph. For example, suppose you wished to compare messages down a channel with the same readings on the previous day. You could enter a time offset of '1 day' on the previous data data and the two sets of data would then be displayed at the same part of the graph making comparison easy.

➢ **Reset Data**
This button can be used to reset all the data for the current line.

14.1.2. Pie Charts

Pie charts can be very useful for showing the relative values of a group of data points. In general all data points would be of similar types. For example, a pie chart showing relative channel usage or showing the comparison of various queue depths. However, a pie chart containing a queue depth and the number of bytes sent down a channel is unlikely to be meaningful. Note that there seems to be no agreed convention of plotting a pie chart with some positive and some negative values. For this reason negative values are treated as positive values. In other words, it is merely the magnitude of the number which matters rather than its sign.

There are three types of pie chart:

1. **Pie – Total**
   This chart plots the total of all the value in the datapoint history against the totals in all the other lines. Plotting the total, rather than just the last value, adds some level of smoothing to the displayed value. The amount of smoothing is effectively controlled by how many data-points are kept for this monitor.

2. **Pie – Difference**
   This chart plots the total of all the differences in the history. Essentially it shows how much the values have changed over time. Negative differences are treated the same as positive differences. For example, a queue which has an intervals where the queue depth has increased by 10 messages will be plotted the same a queue where the depth has decreased by 10 messages.

3. **Pie – Last Value**
   This chart just plots the relative values of just the last value returned. This graph is therefore prone to wild swings as the new values are returned. However, it does show the latest 'state of play'.

A graph can only contain one type of pie chart. It does not make sense to mix and match different pie chart values on the same chart. You can, however, display any number of lines over the top of the pie chart.

You can control what is shown on the pie chart segment labels (if any) by selecting options in the 'Graph' tab described below.

14.1.3. Graph

The graph tab allows the user to set values which are global across the whole graph. The options available are:

- **Title**
  Free form text description of graph. The text entered here will appear in the title-bar of the graph window and also in the 'Windows' menu in the main window.
- **Background**
  Sets the background colour of the graph.
- **Axis**
  Sets the axis colour of the graph.
- **Label**
  Sets the axis label colour of the graph.
- **Grid**
  Sets the axis grid colour of the graph.
- **Axes in front**
  Selects whether the axes, if any, are drawn over the top or behind the data lines.
- **Axes Opaque**
  Chooses whether the axis text should be displayed with an opaque background or not.
- **Size Adjust**
  Chooses whether graph elements such as pie chart labels and line symbols should adjust their size depending on the space that is available.
- **X-Axis**
  - **Hide**
    Selects whether the axis should be hidden.
  - **Ticks**
    Selects whether the tick marks should be drawn.
  - **Minor Ticks**
    Selects whether the minor tick marks should be drawn.

- **Tick labels**
  Selects whether the tick labels are drawn.
- **Margin**
  If selected a margin is added to the extremes of the data.
- **Grid**
  If selected a grid is shown at the major tick marks
- **Integral**
  The axis is scaled to the next integral amount
- **Scroll Hold**
  With this selected the program will try to keep the currently displayed range on the display as long as possible as new data is added to the data points. The exception to this is if the scroll position is at the far right in which case the latest data is always shown.
  If this is not selected then the data will scroll past from right to left as data is added.
- **Auto Range**
  By default MO71 will try to choose a reasonable range to display the data. However, if you wish you can deselect this option, manually set your desired time range and press the 'Apply Range' button. At any time you can depress this button and have MO71 auto-range the data for you.
- **Range**
  This parameter allows the user to set how much of the data is shown on the screen. Essentially it controls the amount of zoom. For example, setting a value of 1 min will show less data than 1 hour. The units are Weeks, Days, Hours, Minutes, Seconds. You can use abbreviations if you prefer. For example '1w3d8h' would be '1 week 3 days 8 hours'

  Note that you must press the 'Apply Range' button to force the new range value to be taken into account. The actual value used is then displayed in the field.

  The time range shown can also be adjusted by clicking on the graph itself and dragging the mouse the required amount. Holding the <*Shift*> key while dragging will only change the X direction. Holding the <*Ctrl*> key while dragging will only change the Y direction.

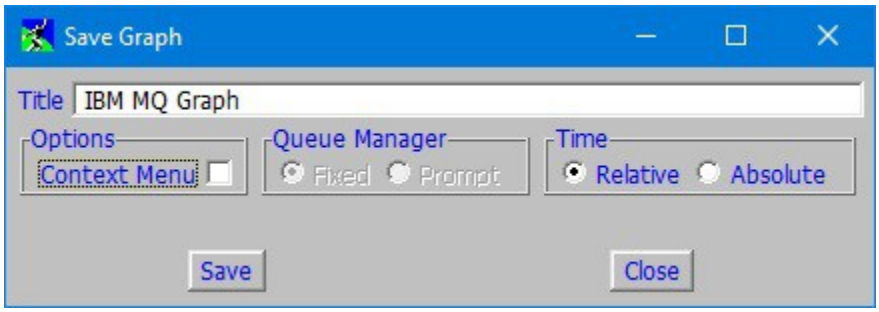
- **Y-Axis**
  Many of the Y-Axis values are the same as the X-Axis values. Except for:
  - **Show zero**
    This guarantees that the zero point is show somewhere on the Y scale.
  - **Scale**
    You can choose between a linear, log10 or log16 based Y axis scale. The advantage of using a log scale is that it is easier to see percentage fluctuations over a wide range. So, for example for a queue depths you can see the queue depth changing from 2 to 3 as easily as you can see it changing from 2,000 to 3,000.
  - **Range**
    Like the X-Axis but this time it is just a number, there is no concept of time on the Y-Axis.

    The range of items shown can also be adjusted by clicking on the graph itself and dragging the mouse the required amount. Holding the <*Shift*> key while dragging will only change the X direction. Holding the <*Ctrl*> key while dragging will only change the Y direction.

- **Columns**
  The options control how columns are displayed
  - **Width Percent**
    This parameter states how much of the available space should be devoted to the actual column themselves. By default this has a value of 90% which allows for just a small gap between the columns.
  - **Overlap Percent**
    If you are displaying multiple lines on the same graph as columns then by default the columns will overlap slightly. You can either increase or decrease the amount of overlap with this option.
  - **Depth Percent**
    This value is used when displaying 3D columns and control how much depth effect should be used.
- **Pie Charts**

- **Labels**
  This parameter controls whether any label is show in each pie segment
- **Description**
  Controls whether the text description should be included in the label
- **Value**
  Controls whether the count of items for this key is included in the label
- **Total**
  Controls whether the total count of all items is included in the label. This option is only actioned if 'V**alue**' is also set.
- **Percentage**
  Controls whether the percentage amount of this key is included in the label
- **Circle**
  Controls whether the pie chart should be displayed as a circle or using as much of the available space as possible. Note that non-circular pie charts can give the wrong ratio percentage since the eye can perceive area as more important than swept angle.
- **Legend Percentage**
  This options controls whether the segment percentage should be included in the legend.
- **Threshold**
  This is the 'Other' threshold value. The percentage value below which the data segment should not be shown in it's own right but as a cumulative part of the 'Other' segment. Pie charts work best when you only have a few segments shown and it is very easy for data to get lost in the noise of large numbers of segments. The 'Other' segment is a way of avoiding this issue by effectively hiding lots of thin segments. Note that the Other threshold is only active if :-
  - there are more than ten segments
  - the other segment consists of at least four segments

- **Save Graph**
  Some graphs, such as Monitor and **MQEV** graphs, can be saved. However, this option will not be shown for say, Activity Trace, graphs, since it is not really applicable. If Save Graph is selected then the following dialog is shown.



The Save Graph has a number of options.

- **Title**
  This is essentially what this graph will be known as. Graphs as saved a predefined dialogs and the title will be stored as the predefined description and should be unique. To start a saved graph then the user need only double click on the created predefined graph dialog definition.
- **Context Menu**
  This option controls whether this graph should be available in the context menu off the Queue Manager command menu.
- **Queue Manager**
  By default the saved definition will be associated with the Queue Manager that was used to create the graph in the first place. However, if the graph consists only of **MQEV** graph lines then the option to save this as a Prompt dialog will be offered. This means that when the graph is started the user is prompted for which queue manager the graph should be instantiated for[29].

[29]This does not happy when the graph is selected from the context menu since the Queue Manager choice has already been made.

- **Time**
  By default the time range of the graph is considered relative. For example, an **MQEV** graph showing the last 7 days of activity will still show the last 7 days when started a month later. However, if you wish you can specify that the graph should be saved using absolute times. This is useful if a particular time range is required.

- **Reset to Graph Defaults**
  When pressed MO71 will apply most of the graph settings from the preference dialog.

- **Set Graph Defaults**
  When pressed MO71 will lock-in the various graph settings so that they are used by default in future graphs.

More discussion of the features and examples of graphing can be found on our blog here.

Chapter 15. MQEV Event Monitoring

MQEV is a program which will receive, store and process MQ event messages and MQ accounting and statistics messages. It greatly alleviates a number of the disadvantages of trying to process these output messages from your queue manager, and allows you to simply and efficiently search, consolidate, summarise and total the data gathered. For a complete description you should read the **MQEV** manual which can be downloaded from http://www.mqgem.com/mqev.html

In order to view and administer the MQEV program, you can use either an command line tool such as **MQSCX** (also from MQGem Software), a GUI tool like **MO71**, or if you prefer, you can write your own tool to interface with **MQEV**. This chapter will detail how to use **MO71** to interact with **MQEV**.

15.1. Licensing

In order to use MO71 to interact with the **MQEV** product, you only require an **MQEV** licence. If a licence file is bought you will be sent an ***mqgem.lic*** file. All you need to do is place this licence file in the same directory as the ***mqmonntp.exe*** program. It is not necessary to end the program when installing a new licence file.

If you already have an **MO71** licence for administrating and configuring your IBM MQ queue managers, you can combine your existing **MO71** license file and a new **MQEV** licence file together in the same file as detailed in *2.2 Multiple licences* on page 6.

15.2. Administering MQEV

There are a number of commands that allow you to set up how MQEV operates. If you use menu categories, these will be in the MQEV Events category:

- MQ Event List...
- MQ Event Processor List ...
- MQ Event Alert List …
- MQ Event Emitter List...
- MQ Event Queue List …
- MQ Event Queue Manager List …
- MQ Event Stream List …
- MQ Event Stream Status List …
- MQ Event Processor
- MQ Event Alert …
- MQ Event Emitter...
- MQ Event Queue …
- MQ Event Queue Manager …
- MQ Event Stream …
- MQ Event Stream Status ...

As with many of the objects you are familiar with using through MO71, there are list dialogs and individual dialogs for each type (including the MQ Event Processor event though there can be only one object of that type, so that you can use it in a multi-queue manager list). Setting up MQEV will include ensuring that the event queues (and if necessary accounting and statistics queues) on your queue manager are being read by the MQEV program. Full details of how to set up MQEV to process event messages and accounting and statistics data can be found in the MQEV manual.

Please refer to the MQEV manual for details about what each of these commands do.

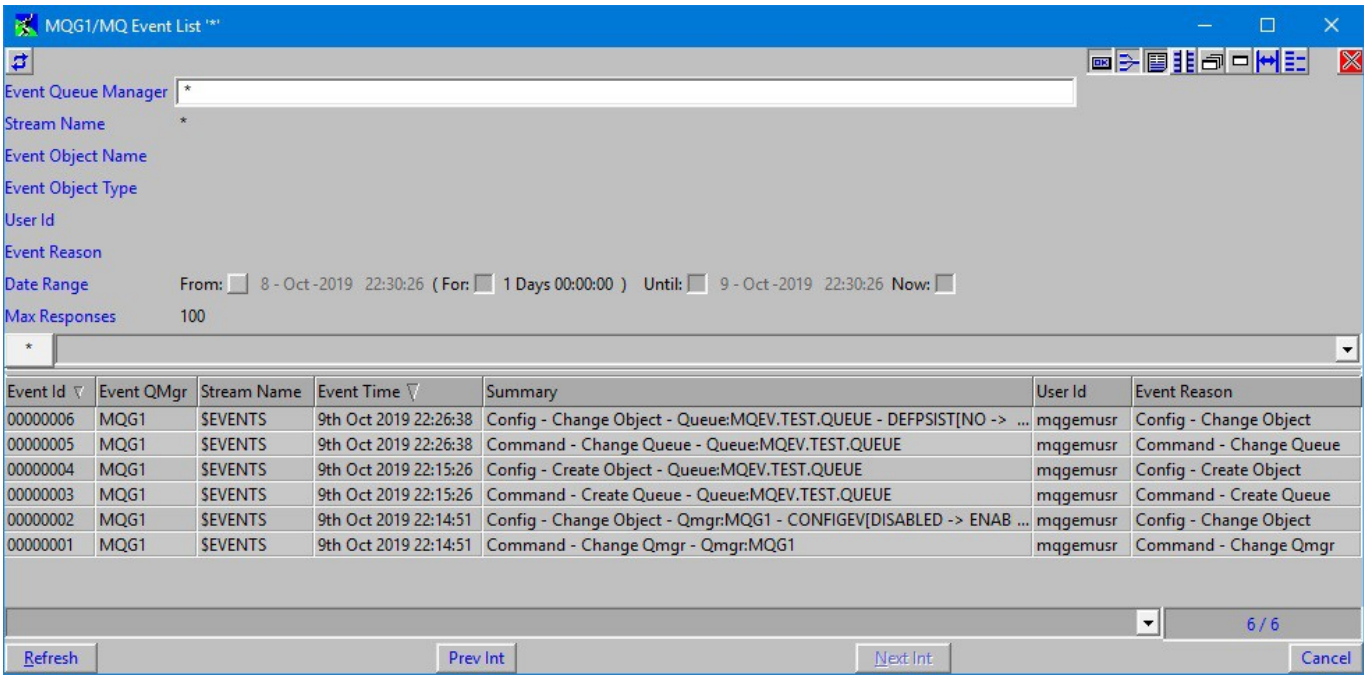
15.3. Viewing MQEV data

Once MQEV is configured and running, you can view, search and filter MQ Event and Accounting and Statistics data using MO71. If you use menu categories, you will find the MQ Event data in the **MQEV Events** category and the others in the **MQEV Acct & Stat** category.

- MQ Event List … (in MQEV Events category)
- MQ Accounting MQI List …
- MQ Accounting Queue List …
- MQ Statistics Channel List …
- MQ Statistics MQI List …
- MQ Statistics Queue List ...

These list dialogs will display the data collected by MQEV, either event data or accounting and statistics data. These list dialogs have some new features over and above what you may be familiar with already with MO71 list dialogs. Alternatively you can directly graph the MQEV Accounting and Statistics data – please see 'Graphing' on page 114 for more information.

Here is an example MQ Event List dialog.



15.3.1. Multiple dialog fields

Almost all MO71 list dialogs have at least one or two fields at the top of the list dialog to allow you to filter the contents of the list. Think of a queue list, you can add a queue name pattern, and choose a queue type to reduce the number of queues listed in the dialog. These new list dialogs are the same, but have quite a number of filter fields because there are a number of different ways you might need to change the contents of the list. Each of the six MQEV data list dialogs will have a different set of these fields because the data is very different.

15.3.2. Date Range selection

Each of the data lists has a Date Range selection control which is a type of control in MO71 that allows you to limit the information you are shown by specifying a range of dates/times you are interested in. You can do this in different ways. By specifying both ends of the date range or specifying just one end and an elapsed length of time. For example, if you look at the dialog you will see that by default it will show you events in the last 24 hours. This is indicated by the 'Now' button being selected – which means that events up until 'now' will be selected. The 'For' field is also selected with a value of 1 day. So, we have a selection of “a days worth of events up until now”. Each time the 'Refresh' button is pressed the greyed out values will be changed to indicate the effective value used on the command.

Of course all of these fields can be changed. By selecting or deselecting the buttons you can enable the data fields themselves and set whatever value you wish. To change a value just select the required field, you will see a bar above and below the field indicating it has focus. It is then merely a question of using the mouse wheel, clicking on the bars or using the up and down arrow keys to select the value you want. Pressing right and left arrow keys will move through the date elements. You can also set these fields using the filter function 'set' which is useful when you want to create predefined dialogs of these lists.

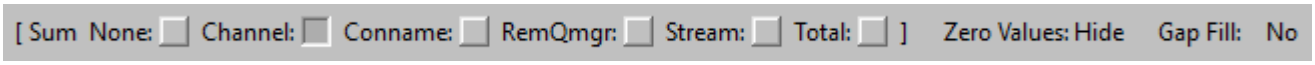
These values, known as spinners, also have a couple of keyboard shortcuts. As we mentioned before, pressing up or down will increase or decrease the value by 1. However, if the shift key is held down at the time then the value will increase or decrease by 10. Note that it is the represented value, not the displayed value, which is changed. So, for example, doing this on the date field will move the date forward or backward by 10 days rather than just increment the displayed number by 10. The other shortcut is that pressing <space> will reset the number, depending on the spinner type, to the following value :-

- **Day** Set the value to the first day of the month
- **Month** Set the value to the first month of year
- **Year** Set the value to the current year
- **All else** Set the value to zero

15.3.3. Summing data and other selection controls

The other unusual control that you will see on the accounting and statistics list dialogs contains a number of selection buttons for summing fields, as well as the choices for “Zero Values” and “Gap Fill”. When using accounting and statistics, records for the same entity can be summed together to show the totals for the time period you are interested in. For example, for queue accounting, you can sum together all the records using the same queue from the same application name. Alternatively you can see each individual record that MQ creates (a sum value of “None”), or you can add all the records together (a sum value of “Total”).

While the fields that can be used in summing vary across the five different list dialogs, the control operates in the same way on each. You can select “None” on it's own, or “Total” on it's own, or several of the others together. Selecting “None” or “Total” will unselect all the other sum buttons. Selecting any of the others will unselect “None” and “Total”.



In addition this control also contains spinner selectors for the “Zero Values” and “Gap Fill” options. Read the command reference in the MQEV manual to learn more about the purpose of those command attributes.

As with the date range selection, these controls can also be configured using the filter function 'set'.

15.4. Graphing your MQEV data

Of course rather than displaying your data in a list you may wish to see the data in a graph. This can be very useful for seeing trends. There are two ways of doing this:

1. **You can bring up the MO71 graph and add the MQEV fields directly.**
   Please see the chapter on 'Graphing' to see how this is done.
2. **Export your collected list as a CSV file**
   Like any MO71 dialog the data shown can be exported into a CSV file. Once you have the data in a CSV file you can then use your favourite graphing tool to graph the data.

15.5. Menu visibility

The menu options related to using **MQEV** will only be visible if **MO71** can detect **MQEV** usage on your queue manager. This is determined by the presence of a queue named MQGEM.MQEV.COMMAND.QUEUE, without which you could not interact with the **MQEV** program. If you use queue manager locations with the “Download Objects” option unchecked (see *24.3.4 Options* on page 216), then it will be unable to determine this automatically, and you can instead open the Location Dialog for your queue manager, go to the “Options” tab and manually tick the “MQEV Commands” option (see above referenced page for details).

As with the majority of **MO71** menu items, you can also tailor which are visible. This is described in Chapter *25 Usage Tailoring* on page 235 and applies to the previously discussed menu items just as with any others.

Chapter 16. Direct Event Monitoring

This chapter deals with a feature where **MO71** can directly capture event messages. This is not making use of the **MQEV** product. If you are using the **MQEV** product, please read the previous chapter.

Earlier chapters dealt with monitoring Queue Managers as a whole and detecting whether they were reachable. However, even though the Queue Manager is available and working it does not necessarily imply that all is well within it. What we need is some way to monitor the individual objects of the Queue manager, most notably Channels and Queues. To facilitate this MQ Queue Managers have the notion of event messages. Event messages are specially formatted messages which are written to 'known' queues when particular key events occur within the Queue Manager. For example, an event message can be generated when the depth of a queue reaches a certain predetermined level. Why, I hear you ask, rather than MQ just raising a message doesn't the Queue Manager just fix the problem? The point here is that different installations may want to perform different actions for the same event; the same event for different objects may well be handled differently; and for a number of these situations it is not at all clear what 'the right' thing to do is. For example, in the case earlier where a queue has reached a certain depth, is the right thing to do just increase the maximum depth of the queue; should one inhibit the queue for puts to prevent any further messages from being put to it; or should one start another instance of the application which is supposed to be servicing this queue. The answer may well depend on what the queue is and what the installation policy is. So it is an advantage that MQ does not enforce a certain policy for these situations and instead generates an event message which provides the most flexible method of informing an application that something has happened.

The downside, of course, is that something has to be written to process these event messages and do something with them. The other problem is that each user wants to monitor different events and do different things when they are triggered. . The solution therefore is to allow **MO71** to be flexible in both what it monitors and what it can be configured to do when receiving an event. In the end ultimate flexibility is achieved by allowing the user to invoke a system command or batch file when an event is triggered. This means that users can literally do anything they wish when an event occurs.

Perhaps the best way of describing monitoring is to do an example.

16.1. Event Example

Let's take the case above of a queue reaching a predefined limit. First we determine what the MQ event message is that will be generated and which queue it will be written to. To do this we look at the *'Event Monitoring'* book and find the chapter on queue depth events. We see that there is an event *'QueueDepthHigh'* which is written to the ***SYSTEM.ADMIN.PERFM.EVENT*** queue. Note that we need to switch on performance events for the Queue Manager and the High Depth Event for this queue. So, now we should have the Queue Manager generating this message, all we have to do is configure MO71 to look for it.

From the main menu select *Action-Event* this will bring up a dialog with quite a number of fields which we'll ignore for now. For the moment just concentrate on the fields we must fill in.

16.1.1. Queue Manager Name

Clearly you place in here the name of the Queue Manager, which must be a location already defined to MO71, i.e. you have a location for this Queue Manager on the main window.

16.1.2. Event Queue Name

For now let's just put the name of the queue we're monitoring **'SYSTEM.ADMIN.PERFM.EVENT'**

16.1.3. Type

We now need to tell MO71 what event we're looking for. Press 'F4′ to drop down the list. Select 'Queue High'.

The rest of the fields are mainly concerned with what we want to happen when we see one of these messages. Let's keep things simple and just write a message to the console. Tab down to the 'Console Priority' field and change 'None' to, say, 'Medium'.

Now press the create button and check the application says an event has been created. The message may well say it has created *2* events. We'll see the reason for this soon.

Now go back to the main window but this time select 'Action-Event List'. This will show a familiar looking list window, pressing refresh will show the event we've just defined. Notice though that it only shows the Queue Name and not the actual event type. By changing the 'Display Type' to 'Detail' and pressing refresh we see the event type we created. We also see another event, called 'Default'. This explains why the message said 2 events created but what is it for? The default event is there because the Queue High Event we're looking for isn't the only message which may be put to this event Queue. If MO71 reads a message off the queue and it

is not a Queue High Event what should it do with the message? This default event allows you to specify that behaviour. But for now, in time honoured tradition, let's ignore it.

The other thing we notice on our event list display is that the status is stopped. What does this mean? Well it means that **MO71** is not at this moment reading this queue. Any messages put to the event queue will not be noticed. So, we need to start the event, select either of the events for this queue, right-click and press start. You should see the status change from Stopped to Running. If you don't there should be a 'status reason' field stating why it didn't work.

Now the exciting bit, we get to see whether it actually works. We need to cause a Queue High Event. Choose a queue which has a fairly low value for 'Max Q Depth' and 'Depth High Limit' and put messages to it. We can use **MO71** itself to put a set of test messages to the queue. Just select *'Action-Put Message'* from the main menu. Now, put a bunch of messages to the queue and keep watching the event list display.

You should see the count of events processed change to 1 as **MO71** receives and processes the message.

Now, if you remember this far back, when we defined the event we changed the console priority to 'Medium'. What this said was that when one of these messages was received MO71 should write a message to the console. So let's go see.....

From the main menu select *'View-View Console'*. You will probably see an info message telling you the time MO71 started and now we also see *a 'Queue High'* message stating that whatever queue name you chose has reached its *Queue High* limit.

Now let's see what happens when the queue depth becomes low again. Use **MO71** to browse the queue. Select 'Message Selection-Apply to all messages'. The 'Delete' button now changes to 'Delete All'. Press the 'Delete All' button. Now it we take a look at the console you see we still have a record there saying Queue High even though we know it's empty. There are two things we forgot to do; firstly if we've enabled Queue High events for the Queue it probably makes sense to also enable Queue low events. Secondly, in our event list we need to add an entry to look for Queue Low. So, go and do that now.

Right, hopefully now you have a event list which shows three entries, Queue High, Queue Low and Default. Let's try the same exercise. Use the 'Action-Put Message' to put a bunch of messages to the queue. You should see the event list for Queue High now read 2. Let's take a quick look at the console. That's strange; we only see one record for Queue High. Why is this? The reason is the 'Console Type' we specified when we defined the event. We set 'cumulative'. What this says is that the console should only display the latest entry if there are two events of the same type, for the same object from the same Queue Manager. If you like you can change this to 'History' which will write a new entry to the console every time it happens.

Anyway, now we want to go and delete the messages on the queue. So do the same trick of 'Delete-All' from the browse window.

The first thing to notice is that in the event list we now correctly have a count of one next to Queue Low event. The second thing to notice in the console window rather than getting a console entry saying 'Queue Low' event instead the 'Queue High' event has gone. Why is this? Well for pretty much the same reason we didn't get two Queue High events. Since we've asked for a cumulative console we're asking for it to display the sum of the events that have occurred. Since a Queue Low cancels a Queue High event it seems logical not to display anything. Most events do not have an obvious opposite but there are a few such as Channel Started/Channel Stopped and Channel not Activated/Channel Activated that operate in the same way.

By now you should understand the basic principals of event monitoring so let's go through all the fields you can specify on an event object.

16.2. Event Fields

The following fields are available on an event dialog.

16.2.1. Queue Manager Name

In its simplest form this is just the Queue Manager name for which this event applies to but it can take wildcards. '*' implies one or more characters. '?' implies a single character. This can be useful for defining an event on all Queue Managers starting with the string 'NT*' for example.

16.2.2. Location

This field is largely for convenience but it can useful in some circumstances where you have more than one location defined for the same Queue Manager.

16.2.3. Type

This is the event type the event is monitoring for. You cannot define a 'default' event nor can you delete the default event unless it is the only event for a particular event queue.

16.2.4. Filter

There may be occasions when you may want different actions depending on the actual contents of the event message. The most likely of these would be the object name that generated the event. For example, suppose you wanted to only write console messages for channels starting with the string 'NT' you could specify a filter of **'channel == NT*'** For a discussion of what can be specified in filter please see “Filtering” on page 28.

16.2.5. Filter Priority

This field can be used to force the correct ordering of events. Suppose you have two events defined for catching the Queue High event. You may have a filter on one of them for just queues called 'ABC'. The second event may have no filter as a catch-all mechanism. For this to work you need to ensure that the filtered event is processed first. To achieve this define the filtered event with a filter priority which is higher than the general event i.e. priority 2 filters are processed before priority 1.

To enable you to make easy additions later it may be a good idea to define your filters with priority such as 10,20,30 and so on so that you can insert another filter between them if required.

16.2.6. Discard

This value essentially switches off the event definition. If Discard is set to 'yes' the event is essentially disabled. Any other events which match the event, including the default filter, will still be processed.

16.2.7. Command

Here you can type the name of a command to be issued if this event is driven. For example, it might run a script which calls the operators pager.

The following inserts are available:

- %m Queue Manager Name
- %l Location Name
- %o Object Name
- %r Reason Code
- %R Reason Code String e.g. ”Queue Full”

16.2.8. Log Line

The format of each line in the log file.

The following inserts are available:

- %m Queue Manager Name
- %l Location Name
- %o Object Name
- %t Time 11.14.12 format
- %d Date in 2003-11-23 format
- %r Reason Code
- %R Reason Code String e.g. “Queue Full”
- %([\n] [\t] [\m] [*] [MQSC Name])
  where
  - \n prints a new line between each attribute
  - \t precedes the attribute with its name (or title)
  - \m precedes the attribute with its MQSC name
  - * prints all attributes of the message
  - MQSC name prints only the given attribute

16.2.9. Requeue

One of the problems with processing event messages is that there may be occasions where you want more than one program to monitor events. The simplest way of achieving this is to daisy chain the monitors. Have one monitor processing the 'real' event queue and when it's finished it then puts the message to another queue for another program to then process. This field allows you to

specify where MO71 should put the message once it has finished processing it. If no destination is specified the message will be discarded.

16.2.10. Log File

If required a hard copy record of the events which have been caught can be written to file. This can be useful as an audit trail or perhaps to post process and see how often certain events are being generated. Enter the full path of the file name to write to. The file name can contain character inserts which are described in Chapter 24.8 File Name Inserts on page 228.

16.2.11. Log Event

This field is currently not supported.

16.2.12. Auto Start

As its name suggests it allows for the event monitor to reading the queue as soon as MO71 is started. If set to **'No'** the event must be started manually. Note that starting any event for the same queue will start all of the events defined for that queue. It is not possible to only start one.

16.2.13. Console Priority

This field tells MO71 whether the event should be logged to the console. A value of 'None' implies that no console entry will be made.

16.2.14. Console Type

Some events have an equivalent but opposite event. For these events it can make sense to allow them to cancel one another out so that only the negative situation is ever presented on the console. The value for this is 'cumulative'. If however you want every instance of this event to be logged to the console you can specify 'History' so that a history of all of these events is maintained in the console.

16.3. Operational Considerations

In order to monitor event messages MO71 must have a connection to the Queue Manager and open the event queue. It is written to spend its whole time sitting in an MQGET. Consequently the more event queues which need to be monitored the more connections are required to the Queue Manager. Different queues can be useful if you have one type of monitoring application monitoring one set of events and another monitoring a different set. However, if you have a program, such as MO71, which can deal with different events from different queues it is far more efficient to merge the event queues. The simplest way of doing this is to redefine the event queues as alias queues pointing to a central event queue. A simple MQSC script to do this would be :-

```
DELETE QLOCAL(SYSTEM.ADMIN.PERFM.EVENT)   PURGE
DELETE QLOCAL(SYSTEM.ADMIN.QMGR.EVENT)    PURGE
DELETE QLOCAL(SYSTEM.ADMIN.CHANNEL.EVENT) PURGE


DEFINE QLOCAL(GENERAL.EVENT)


DEFINE QALIAS(SYSTEM.ADMIN.PERFM.EVENT)   TARGQ(GENERAL.EVENT) REPLACE
DEFINE QALIAS(SYSTEM.ADMIN.QMGR.EVENT)    TARGQ(GENERAL.EVENT) REPLACE
DEFINE QALIAS(SYSTEM.ADMIN.CHANNEL.EVENT) TARGQ(GENERAL.EVENT) REPLACE
```

Once this script is run all events will be pointed at the general event queue 'GENERAL.EVENT' and only a single connection to the Queue Manager is required. This mechanism could even be extended to collect events from remote queue managers to a single, central event queue, by defining the event queues as remote queues.

16.4. User Events

So far we have talked solely about events that have been generated by IBM MQ. However, there is another type of event, called a User Event. A User Event are events which can be used to communicate directly to MO71. The principal reason you might want to do this is to put a message to (or delete a message from) the console.

A User Event, as you might expect, is a very similar format to normal MQ Events in that it is a PCF message with a format type of "MQEVENT ". There are two such user events, MQG_MO71_ADD_EVENT(50000) which will add an entry to the console and MQG_MO71_RMV_EVENT(50001) which will remove one.

The event itself can contain up to five fields, these are as follows:

Field	Constant	Type	Add Event	Remove Event
Console text The text which should be displayed on the console[30]	MQG_MO71_EVENT_TEXT (50002)	MQCFST	Yes	Yes
Console type Free Format text which should give a short text description of the event.	MQG_MO71_EVENT_TYPE (50005)	MQCFST	Optional	Optional
Priority The priority of the console entry where : 0 - NONE 10 - INFO 20 - LOW 30 - MEDIUM 40 - HIGH If not specified it is taken from the MO71 event definition.	MQG_MO71_EVENT_PRIORITY (50003)	MQCFIN	Optional	No
Object An option object name associated with the entry	MQG_MO71_EVENT_OBJECT (50004)	MQCFST	Optional	Optional
Category A free-format string which can be used to categorize the event.	MQG_MO71_EVENT_CATGORY (50006)	MQCFST	Optional	Optional

16.4.1. Operation

In order to have **MO71** process the user event then clearly it must be configured to read the event off the appropriate event queue. You may choose to keep user events separate from normal IBM MQ Events by putting them to another queue such as USEREVENTQ or MO71EVENTQ.

The event queues themselves can be generated from any program as long as it has the authority to put to the designated event queue. In the MO71 zip file is a program, called **MO71Events**, which will generate the events which can be called from the command line. This would make it easy, for example, to invoke the program from a scripting program such as **MQSCX**.

The program is simple to operate and help can be obtained by the command **mo71event -?**

Add an event with a command such as:

```
mo71event -m QM1 -e USEREVENTQ -o MYOBJECT -p HIGH -a "Is not being processed"
```

Provided **MO71** is configured to process the USEREVENTQ this will result in a new entry on the console. You can remove this entry by the following command:

[30]MO71 Version 9.3.3 introduced the concept of 'text' on the console so user events were changed to take advantaged of that. This does potentially have some impact on the use of an earlier version of MO71Event. Please see guidance on migrating from versions previous versions in Migration from previous versions for further information.

```
mo71event -m QM1 -e USEREVENTQ -o MYOBJECT -r "Is not being processed"
```

The event text is free format but if it starts with an exclamation mark (!) it has a special meaning which is that this definition overrides all other console definitions with the same Type, Object Name and Category.

To remove an entry the type, text, category and object name must all match the current entry. As mentioned before a text string starting with an exclamation mark (!) will match with any text string[31]. Of course console entries can also be delete directly from within **MO71** itself.

Using a combination of these fields it is clear that it is possible to add many events for the same object with different text or the same text for many different objects. The use of the category can be useful in separating the event entries both for avoiding clashed between reporting entities but all for filtering the resultant display in **MO71.**

[31]Note that the exclamation mark in any text string will not be shown in MO71.

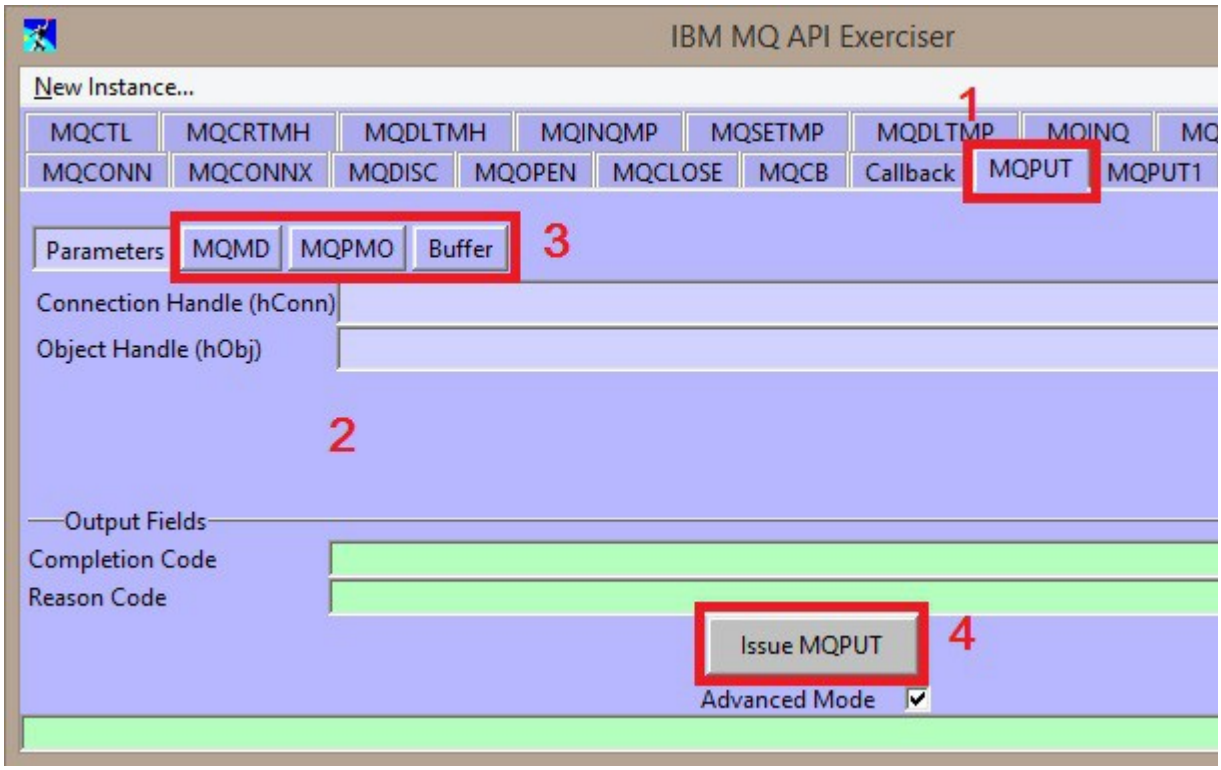
Chapter 17. API Exerciser

The API Exerciser is provided to allow the user to make API calls through a graphical interface making testing of specific options and details, and inspection of reason codes generated, quick and easy prior to writing the code for an application. Clearly it is not an appropriate tool to write an application for a production system, but is excellent for trying things out in a development or sandbox arena.

You can start the API Exerciser from the main **MO71** window, via the Action menu, or by pressing F9.

17.1. Main Layout

Each MQ API verb is provided on a tab.



1. Switch to the tab for the verb you need
2. Fill in the appropriate fields within that tab
3. If there are structures associated with the verb, there will be a section for “Parameters” and a separate section for each structure. Click on each of the structures and fill in appropriate fields within the structure.
4. Press the button to issue the verb.

There are a number of points that are common on each tab, regardless of which verb is it used for.

Each tab will have a button labelled, “Issue MQ*verb*” which you press when you are ready for the verb to be issued using all the information you have provided in the various fields within the tab.

This button, along with the Output Fields for the verb (for example Completion Code and Reason Code) are shown at the bottom of the window whichever structure you are looking at.

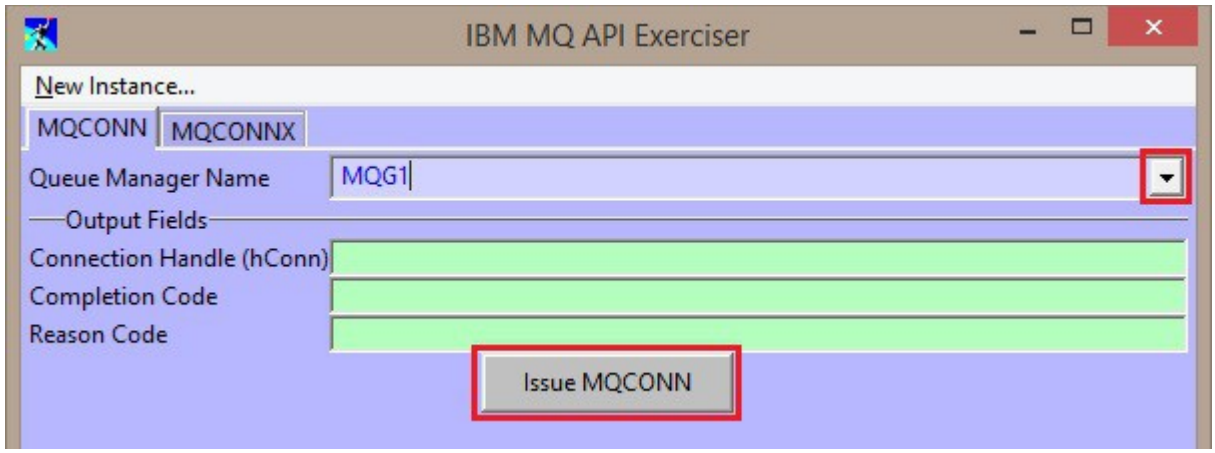
There is an Advanced Mode check box at the bottom of the window. When this is checked, all verbs and options are available to you to try even if they don’t make sense. More details later.

There is a status bar at the very bottom of the window where messages will be written to indicate any obvious errors detected prior to the user pressing the “Issue MQ*verb*” button. For example, unknown structure version numbers will be indicated here.

17.2. Walk-through

As an example, we'll step through a few of the verbs to illustrate how the exerciser is used. We will assume that Advanced Mode is not checked throughout this walk-through.

17.2.1. MQCONN



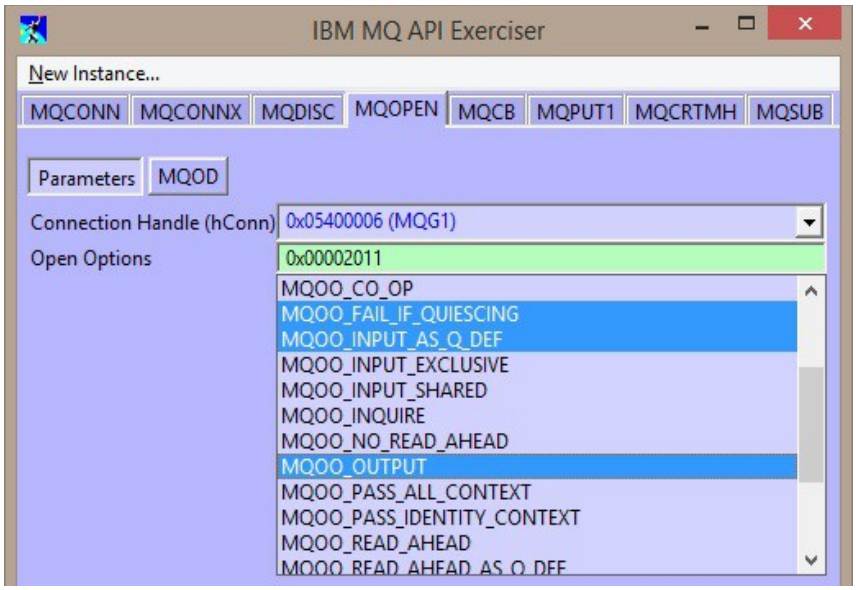
Starting on the MQCONN tab (which along with MQCONNX is the only one available to begin with), choose a queue manager name from the drop-down which contains all the queue managers defined to MO71. Having chosen your queue manager name, press the "Issue MQCONN" button and you will see the output fields are filled in, with a completion code and reason code. If you have successfully connected to the queue manager, a connection handle will also be returned to you. You will see that there are a number of additional tabs now visible.

The MQCONN tab is using the information stored in **MO71** about the location to connect to it. This means that a local or client connection will be used based on what is in the configuration. If you want to control exactly how the connection is made you can alternatively use the MQCONNX tab. More on that later.

You can double click on the queue manager name field when it is populated with a queue manager name and **MO71** will display the queue manager attributes dialog.

17.2.2. MQOPEN

Now we switch to the MQOPEN tab. We can see that the connection handle that was returned to us on the MQCONN call is already chosen for us on the MQOPEN tab. If you make several connections to queue managers, this drop-down box allows you to choose the connection handle for the queue manager you want. The API Exerciser will label your connection handle with the name of the queue manager it is for, to aid you. You will see this name in parentheses after the connection handle hexadecimal value.

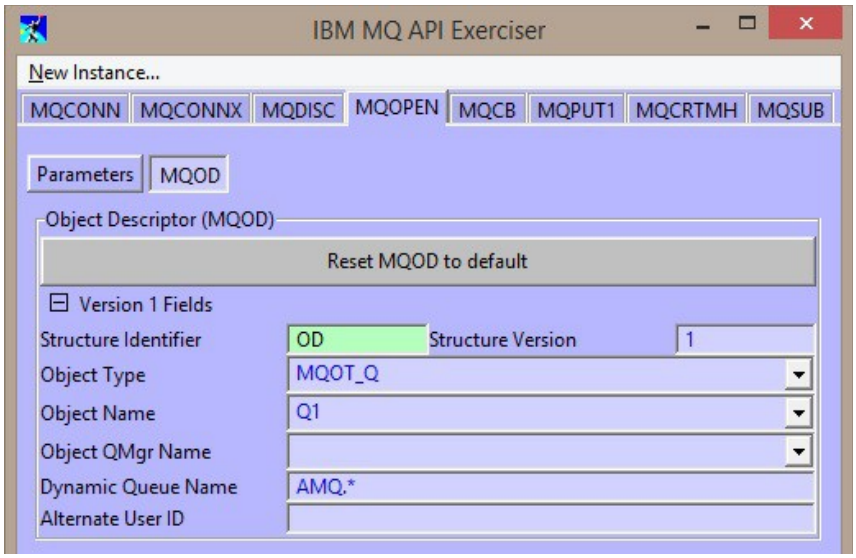


Select the open options you wish to use, holding down the <CTRL> button in order to select more than one. You can see the total number being added up for you in the field above the list of options. Keep an eye on the status line at the bottom of the window. If the API Exerciser detects a combination that is going to fail with MQRC_OPTIONS_ERROR it will provide information there.

Before clicking on the "Issue MQOPEN" button, we will add some information in the Object Descriptor.

17.2.3. Object Descriptor (MQOD)

Now click on the MQOD button to provide fields in the Object Descriptor structure. This structure comes pre-filled in, in the API Exerciser, with the fields that would be in the C-structure MQOD_DEFAULT if you were writing a C program. In the simplest case, all you need to do here is select an Object Name from the drop-down list and press the “Issue MQOPEN” button.



There are of course more fields that you can optionally use, and longer versions of the Object Descriptor which you can view by editing the Structure Version entry field.

If you make lots of changes to the Object Descriptor and then want to start again, you can press the “Reset MQOD to default” button which will reset the fields to the values from the MQOD_DEFAULT C-structure again.

You can double click on the Object Name field and MO71 will display the object details dialog. If you have typed in the name of a queue that doesn’t exist (rather than selecting it from the drop-down list) this will bring up an empty dialog allowing you to define the queue through **MO71**. If you choose an Object QMgr Name as well, and it is a different queue manager to the one you are connected to (i.e. you are opening a fully qualified remote queue), the queue attributes dialog displayed will show you the local queue definition on that remote queue manager (so long as it is known to MO71).

At the time of writing, the fields in a version 2 Object Descriptor are not supported by the API Exerciser. Those in versions 1, 3 and 4 of the Object Descriptor are supported.

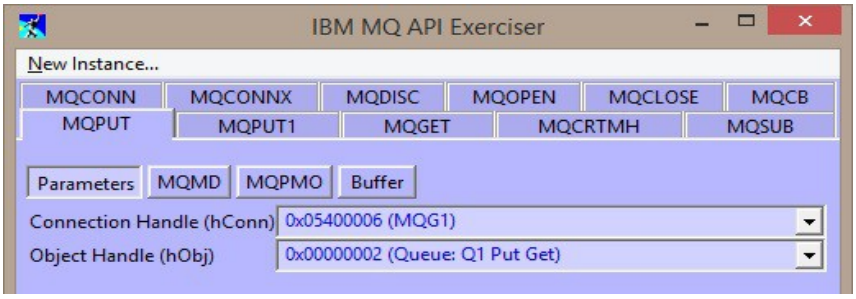
17.2.4. Using other connection handles

If you have made a number of MQCONN calls, the Connection Handle drop down box will contain all the hConns. The most recently created hConn will be pre-selected for you, but you can choose any other as required.

17.2.5. Opening other object types

Of course you can issue an MQOPEN call on objects other than queues. When doing this, I suggest first filling in the Object Descriptor with the Object Type and Object Name, then switching back to the Parameters section where only those options appropriate for the Object Type you have selected will be shown. This will make it easier to select the options you need.

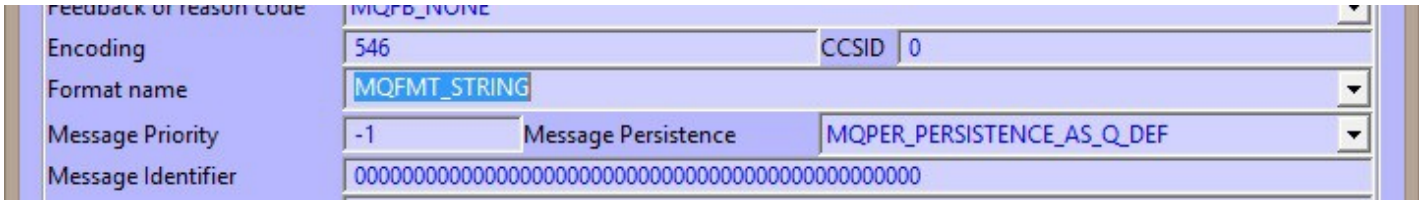
17.2.6. MQPUT



If the object we opened in the previous step was a queue and we opened it with the MQOO_OUTPUT option, the MQPUT tab will now be available to use. Switching to that tab we see the object handle that was returned to us on the MQOPEN call is already chosen for us on the MQPUT tab. If you open several queues (or topics) for output, this drop-down box allows you to choose the object handle for the object you want. The API Exerciser will label your object handle with the name of the object it is for, and the options it was opened with, to aid you. You will see these details in parentheses after the object handle hexadecimal value.

17.2.7. Message Descriptor (MQMD)

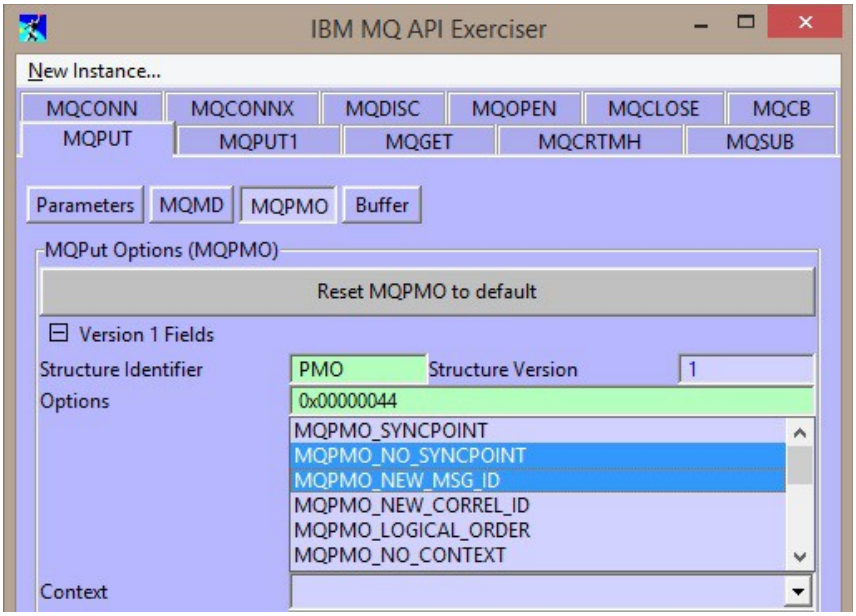
Now click on the MQMD button to provide fields in the Message Descriptor structure. This structure comes pre-filled in, in the API Exerciser, with the fields that would be in the C-structure MQMD_DEFAULT if you were writing a C program. We will make one simple change. We will choose MQFMT_STRING in the Format name drop down.



There are of course lots of fields that you can optionally use, and a second version of the Message Descriptor which you can view by editing the Structure Version entry field.

If you make lots of changes to the Message Descriptor and then want to start again, you can press the “Reset MQMD to default” button which will reset the fields to the values from the MQMD_DEFAULT C-structure again.

17.2.8. Put Message Options (MQPMO)



Now click on the MQPMO button to provide fields in the Put Message Options structure.

Select the put message options you wish to use, holding down the <CTRL> button in order to select more than one. You can see the total number being added up for you in the field above the list of options. Keep an eye on the status line at the bottom of the window. If the API Exerciser detects a combination that is going to fail with MQRC_OPTIONS_ERROR it will provide information there.

At the time of writing, the fields in a version 2 Put Message Options are not supported by the API Exerciser. Those in versions 1 and 3 of the Put Message Options are supported.

Before clicking on the “Issue MQPUT” button, we will fill in our message buffer.

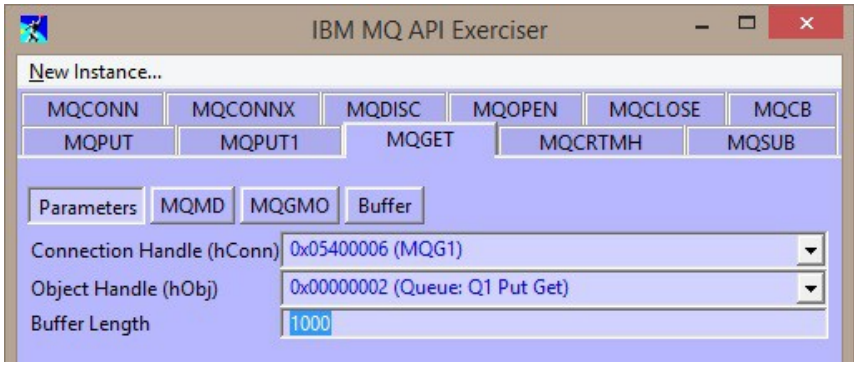
17.2.9. Message Buffer

Now click on the Buffer button and type your message text in the Buffer entry field. The Buffer Length field will be automatically filled in with the length of the text you type, although you may edit it to be longer or shorter if you wish.

Press the “Issue MQPUT” button, and check out the various output fields that are now filled in. There are output fields in your Message Descriptor (MQMD) and Put Message Options (MQPMO) structures when the verb is issued successfully.

17.2.10. MQGET

If the object we opened in the earlier step was a queue and we opened it with one of the MQOO_INPUT_* options, the MQGET tab will now be available to use. Switching to that tab we see the object handle that was returned to us on the MQOPEN call is already chosen for us on the MQGET tab. If you open several queues for input, this drop-down box allows you to choose the object handle for the object you want. The API Exerciser will label your object handle with the name of the queue it is for, and the options it was opened with to aid you. You will see these details in parentheses after the object handle hexadecimal value.



Fill in a value in the Buffer Length field.

Before clicking on the “Issue MQGET” button, we will add some information in the Message Descriptor and the Get Message Options.

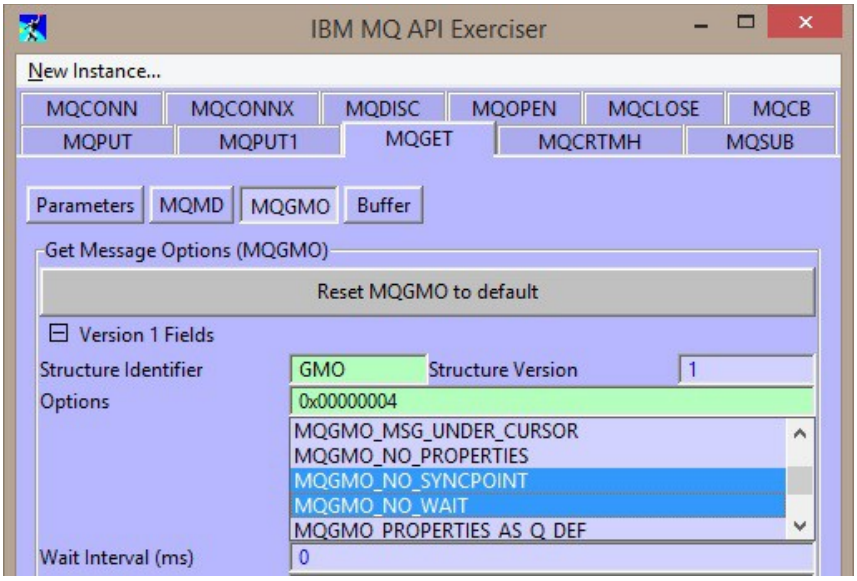
17.2.11. Message Descriptor (MQMD)

Now click on the MQMD button to provide fields in the Message Descriptor structure. This structure comes pre-filled in, in the API Exerciser, with the fields that would be in the C-structure MQMD_DEFAULT if you were writing a C program. The default Message Descriptor will do for our purposes this time. If this isn’t the first MQGET you have issued, you might want to press the “Reset MQMD to default” button which will reset the fields to the values from the MQMD_DEFAULT C-structure again.

17.2.12. Get Message Options (MQGMO)

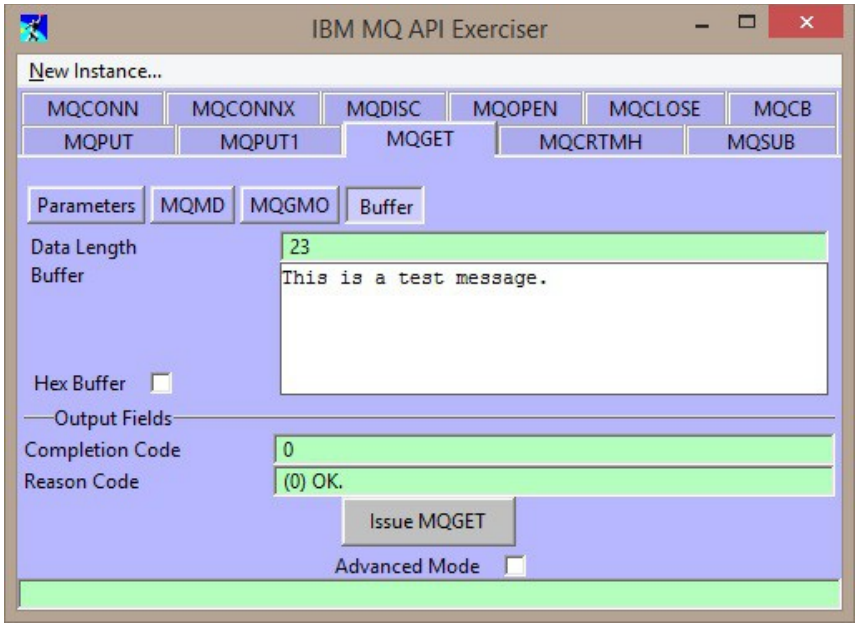
Now click on the MQGMO button to provide fields in the Get Message Options structure.

Select the get message options you wish to use, holding down the <CTRL> button in order to select more than one. You can see the total number being added up for you in the field above the list of options. Keep an eye on the status line at the bottom of the window. If the API Exerciser detects a combination that is going to fail with MQRC_OPTIONS_ERROR it will provide information there.



17.2.13. Message Buffer

Now click on the Buffer button and chose whether you want to display your message as a Hex Buffer or not. Press the “Issue MQGET” button, and view your message contents and check out the output fields that are now filled in, in your Message Descriptor (MQMD) and Get Message Options (MQGMO) structures when the verb is issued successfully.



17.3. Walkthrough with multiple instances

The API Exerciser can be used with multiple instances of the Exerciser window running at the same time. Either by starting one from the Action menu multiple times, or by selecting New Instance from menu bar on the first instance started. As an example of this, we’ll step through an example of subscribing in one window and publishing in another window. This example will also illustrate the use of message handles and message properties.

Start by connecting to your queue manager in each window.

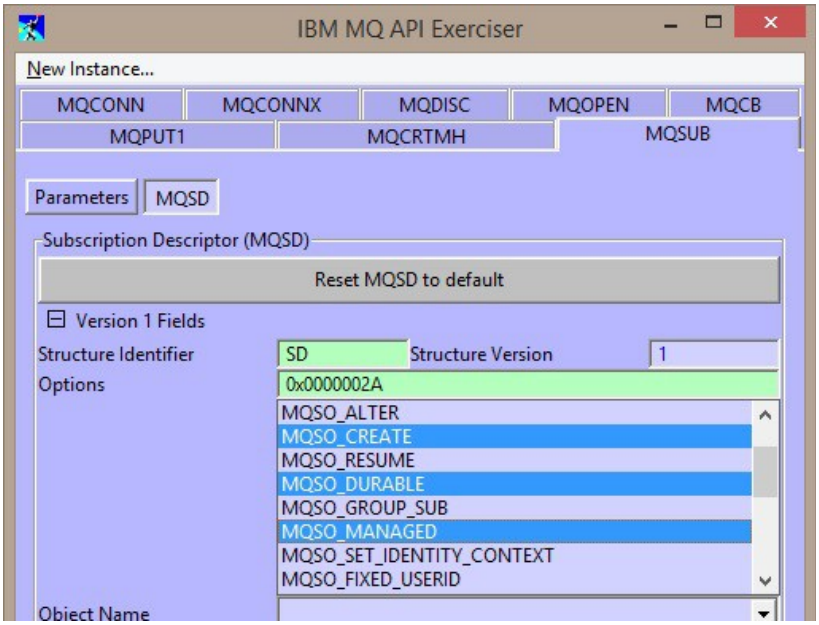
17.3.1. MQSUB

Now we switch to the MQSUB tab. We can see that the connection handle that was returned to us on the MQCONN call is already chosen for us on the MQSUB tab. We are going to select the MQSO_MANAGED option in a moment, so we don’t need to select anything in the Object Handle field. Before clicking on the “Issue MQSUB” button, we will add some information in the Subscription Descriptor.

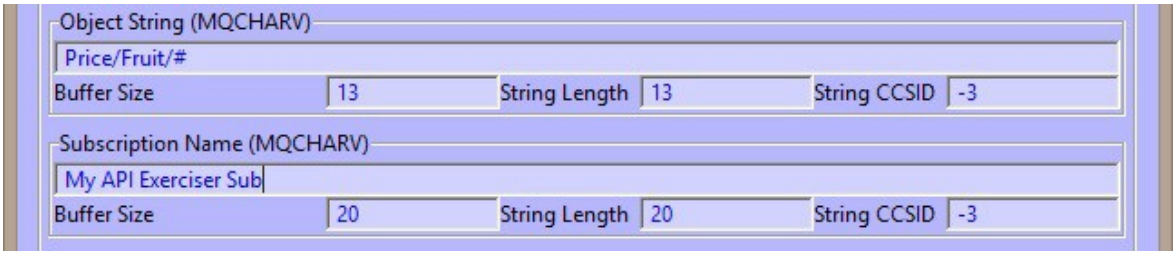
17.3.2. Subscription Descriptor (MQSD)

Select the subscription options you wish to use, holding down the <CTRL> button in order to select more than one. You can see the total number being added up for you in the field above the list of options. Keep an eye on the status line at the bottom of the window.

If the API Exerciser detects a combination that is going to fail with MQRC_OPTIONS_ERROR it will provide information there.



Fill in a topic string to subscribe on in the Object String field and, since we have chosen the MQSO_DURABLE option, also fill in a Subscription name.



Press the “Issue MQSUB” button, and check out the output fields that are now filled in, in your Subscription Descriptor (MQSD) structure when the verb is issued successfully. You will also see at the bottom of the window that this verb has returned you two handles, an Object Handle and a Subscription Handle.

17.3.3. Using MQCHARV data types

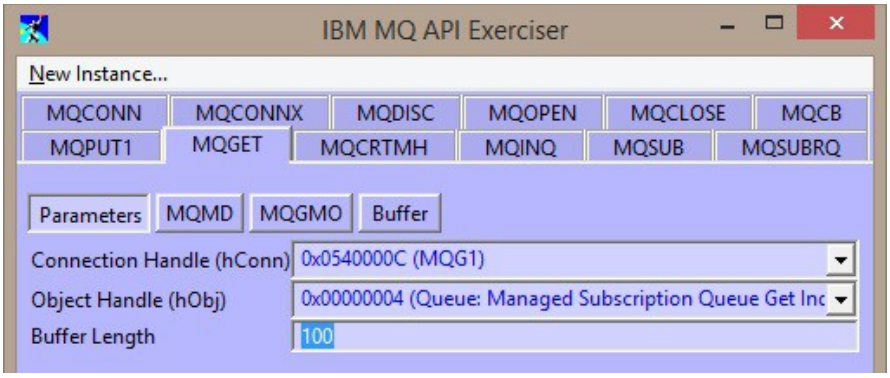
A number of the fields in the MQSD structure (and in other structures) have a data type of MQCHARV. These are variable length structures. The String Length field will be automatically filled in with the length of the text you type, although you may edit it to be longer or shorter if you wish. If the MQCHARV field in question is an output field, you should provide a non-zero value in the Buffer Size field in order to be returned the contents on output of the verb.

17.3.4. Creating a Message handle

In order to use message property verbs, you first need a message handle. Switch to the MQCRTMH tab and press the “Issue MQCRTMH” button to obtain a message handle.

17.3.5. MQGET from a handle returned from MQSUB

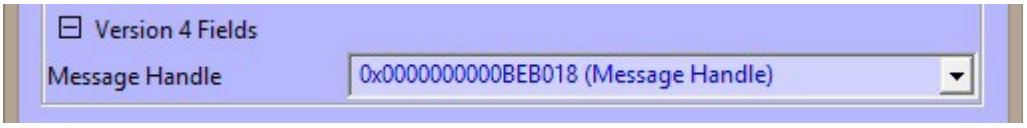
Now switch to the MQGET tab. We see that one of the handles that were returned to us on the MQSUB call is already chosen for us on the MQGET tab.



On the Parameters sub-tab, provide a Buffer Length.

If this isn't your first MQGET call then ensure that the MQMD is reset by pressing the “Reset MQMD to default” at the top of the MQMD sub-tab.

In the MQGMO sub-tab, select Options MQGMO_WAIT and fill in a Wait Interval of 60000 ms. To provide the message handle you created earlier to be used on this MQGET, you must change the MQGMO version to 4. If you scroll down to see the version 4 fields you will see that the Message Handle field has been automatically populated with the handle you created. If you didn't want to use it (but still wanted to have a version 4 MQGMO) you could select the special value of MQMH_NONE from the drop-down.



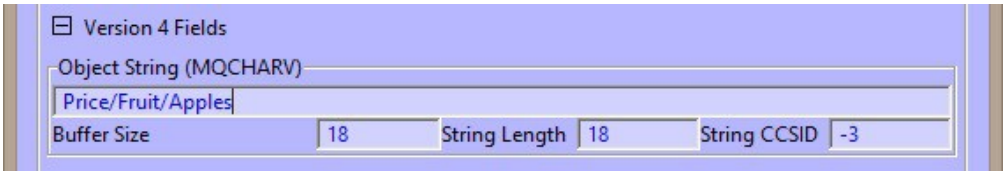
Now press the “Issue MQGET” button. We will come back to this window later.

We are now going to use our second window to publish to this topic. We already saw an example of MQOPEN and MQPUT, so now we shall use MQPUT1.

17.3.6. MQPUT1

On the Parameters tab the Connection Handle we were returned from the MQCONN in this window is pre-filled in for us. Note that you cannot use Connection Handle from the other instance of the API Exerciser. Think of each instance as a separate application.

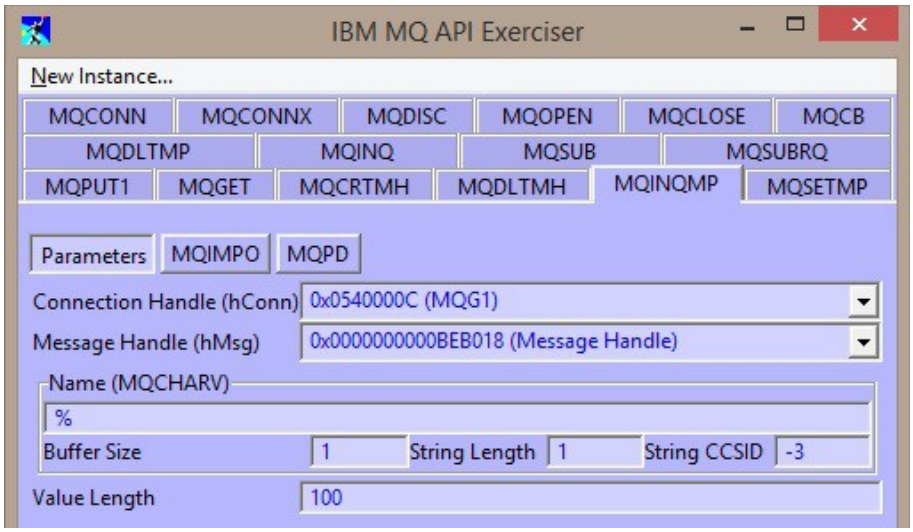
The MQPUT1 verb has three structures and you will see a button for each one, MQOD, MQMD, MQPMO and Buffer. You have seen all of these in the earlier walkthrough. Fill in an Object String in the MQOD structure that will match the subscription you just made (note my example used a wildcard #). You will need to change the structure version number to 4 in order to see these fields and change the Object Type to MQOT_TOPIC.



Provide any attributes you want in the MQMD (for example Format set to MQFMT_STRING) and MQPMO structures and fill in a message in the Buffer. Now press the “Issue MQPUT1” button.

So long as your wait interval of 60 seconds has not already elapsed on the MQGET, you should see the message immediately appear in the other window, if it has already elapsed you’ll have to press the “Issue MQGET” button again.

17.3.7. MQINQMP



Published messages will always contain at least one message property (unless you turn these off using property control attributes on subscriptions and queues). We will look at the property on our received message now. Switch to the MQINQMP tab.

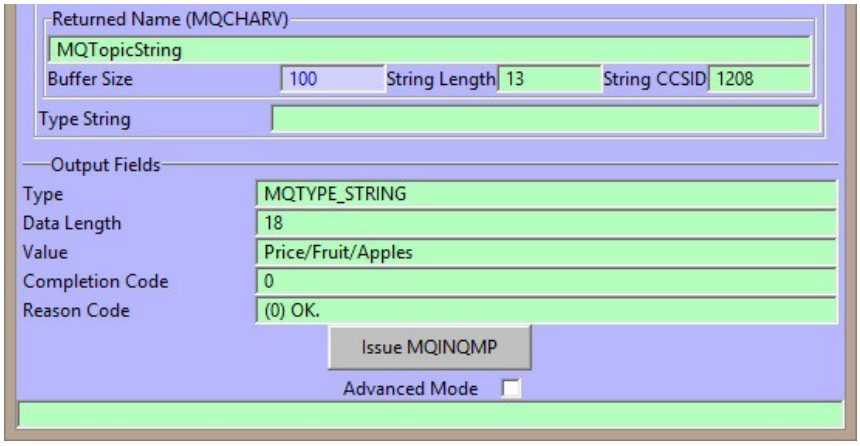
In the Parameters, provide a property name of %. This single character is the wildcard for property names. Also provide a length for the buffer which will receive the value of the property.

You'll note that the status bar contains a warning, because you have supplied a wild-carded property name, you probably also want to provide a buffer to receive the actual name of the property. This buffer is supplied in the MQIMPO structure.

Switch to the MQIMPO sub-tab and fill in a value for the Buffer Size of the Returned Name (MQCHARV) structure.

Now press the “Issue MQINQMP” button. In the Output Fields, you'll see the type and value of the inquired message property, and filled in in your Returned Name buffer just above the Output Fields you'll see the actual name of the property.

To inquire the next message property (if there is one), select the Option MQIMPO_INQ_NEXT and press the “Issue MQINQMP” button again.



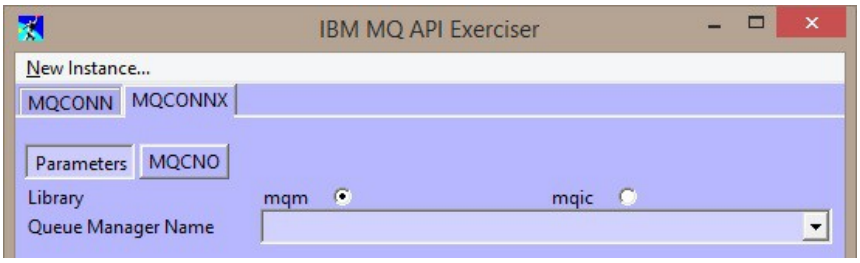
17.3.8. MQCLOSE

On the subscriber API Exerciser instance, now switch to the MQCLOSE tab. There are two choices in the drop down for the Object Handle, chose each one in turn, selecting MQCO_REMOVE_SUB for the subscription handle one, and press the “Issue MQCLOSE” button for each one.

17.4. MQCONNX

So far, you've only made use of the MQCONN verb which takes all the configuration from the MO71 location dialog. You can alternatively use the MQCONNX tab and provide all these options directly. Additionally you can choose to make the connection using the mqm library or the mqic (client) library. We'll step through using all the various options and tabs in the following example.

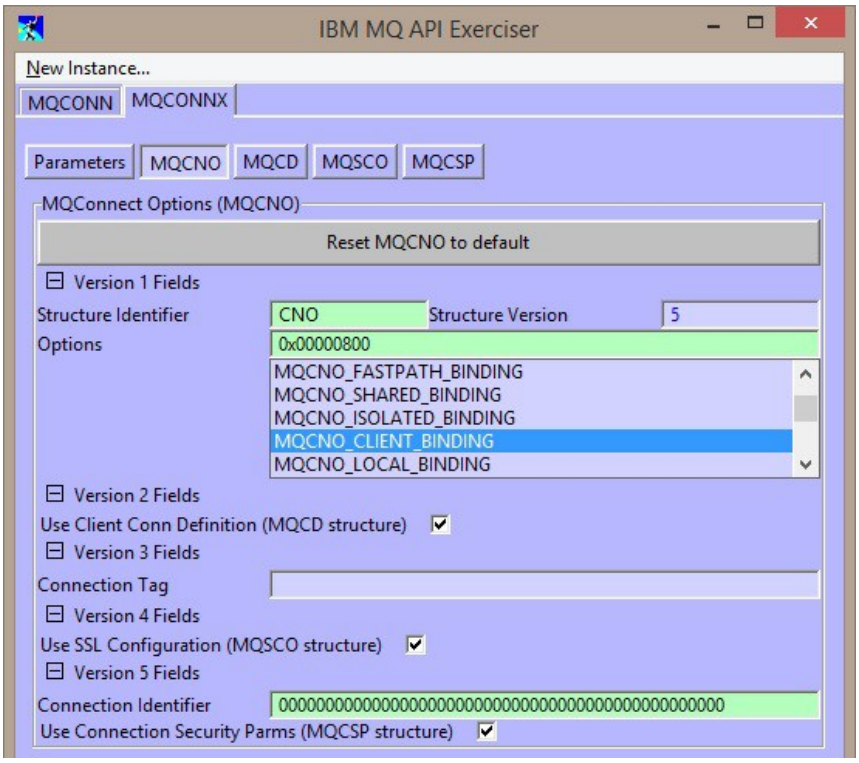
Switch to the MQCONNX tab. In the Parameters sub-tab you have the choice of library. This will be dependant on what you have installed on your system and will only offer you the use of the ones that are available. In our example we're going to make a client connection using the mqm library (this assumes you are using MQ at V7 or later!). For a client connection, we don't have to provide a queue manager name, we can leave it blank or fill it in if you want to.



Now switch to the MQCNO sub-tab. To start with you only see a version 1 MQCNO. We will need a higher version number to be able to use the various additional structures for specifying client connection details.

Change the version to at least 5 and also select the Option MQCNO_CLIENT_BINDING.

There are various optional structures hung off the MQCNO. In order to indicate that you want to use them (rather than passing a null pointer or zero offset to the MQCONNX verb) you must check the box against each one. When you do, the sub-tab for that structure will become visible. For this walk-through we'll use all three!



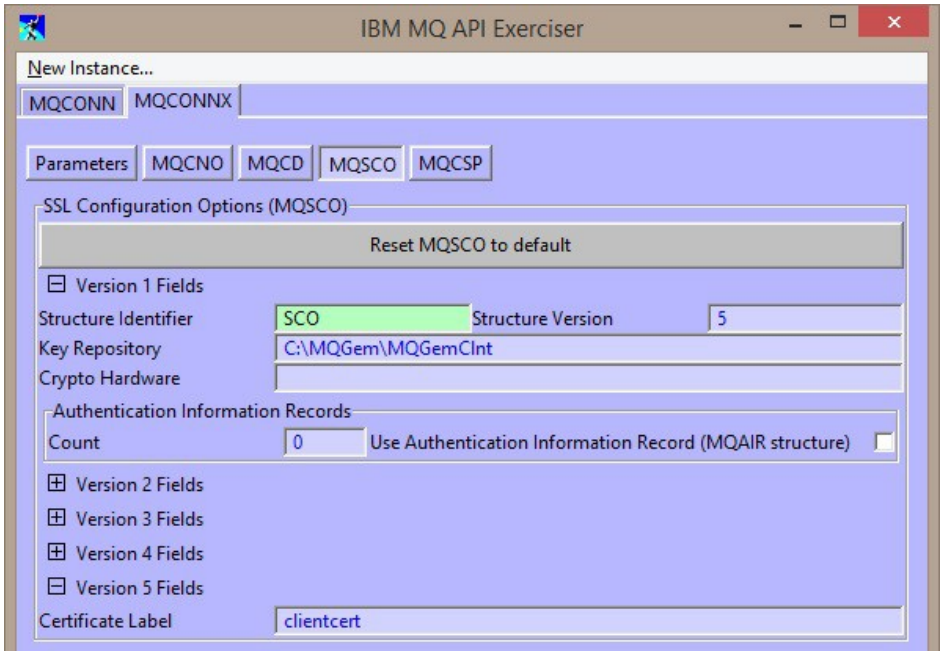
17.4.1. Channel Definition (MQCD)

In the channel definition you'll need to, at a minimum, provide a Channel name and a Connection Name (which is found in the version 2 section). This structure comes pre-filled in, in the API Exerciser, with the fields that would be in the C-structure MQCD_CLIENT_CONN_DEFAULT if you were writing a C program. This default structure (unlike all the others in the MQI) starts off with a version set to 6. I'm going to make an SSL connection (in order to demonstrate the various other structures on the MQCONNX) and so I'll need a version number of at least 7 in order to also be able to select an SSL Cipher Spec.

17.4.2. SSL Configuration Options (MQSCO)

When making an SSL/TLS connection, there are a number of options you might need in the MQSCO structure. I'm going to set the version to 5 in order to provide a Certificate Label, as well as filling in my Key Repository location.

In the SCO structure, you also have the opportunity to provide Authentication Information Records (MQAIR). In the MQI you can have a number of these. In the API Exerciser this number is limited to 3. As with the other structures in the MQCNO, if you want to use these you must check the box to indicate this, and then also fill in the count for how many of these structures you need. The appropriate number of tabs for these structures will then become visible.

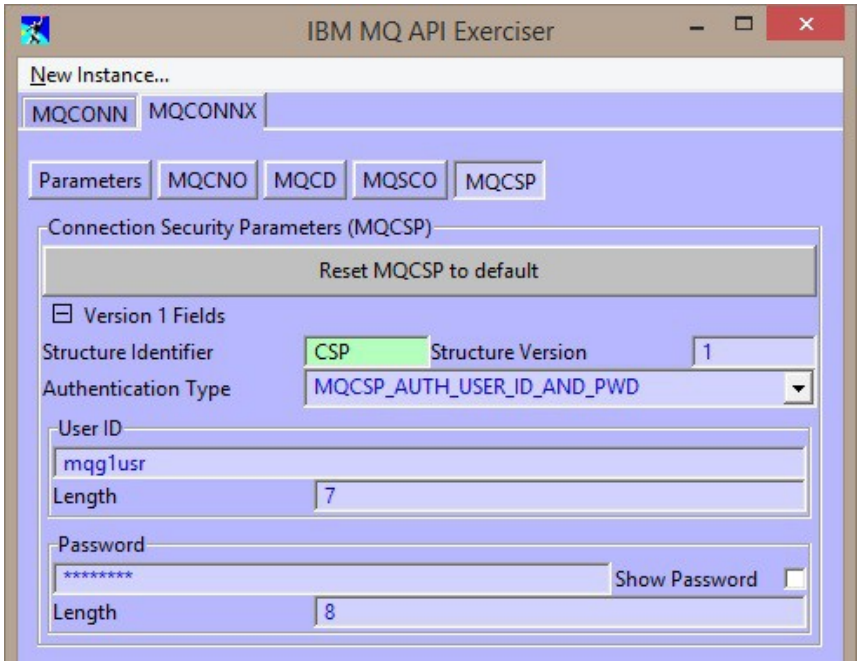


17.4.3. Connection Security Parameters (MQCSP)

Last, but by no means least, we're going to provide a user id and password to authenticate to the queue manager.

On the MQCSP sub-tab we select an Authentication Type of MQCSP_AUTH_USER_ID_AND_PWD (because otherwise the queue manager will ignore the contents of the MQCSP structure - you may have security exits that operate differently). Then we fill in the two fields, User ID and Password.

The associated length fields are automatically populated as you type. You can change them to be longer or shorter if you wish. The password field is hidden as you type by default, but you can show/hide it with the check box (nothing to do with the MQI just a handy feature of the API Exerciser UI).



Once you've filled in all the attributes you need in the various optional structures that are part of the MQCONNX verb, you can press the “Issue MQCONNX” button to see if it worked. Remember that the Reason Code and the error log will provide more information if it failed. MQCONNX is a complicated verb and can take some getting used to.

17.5. Asynchronous Consume

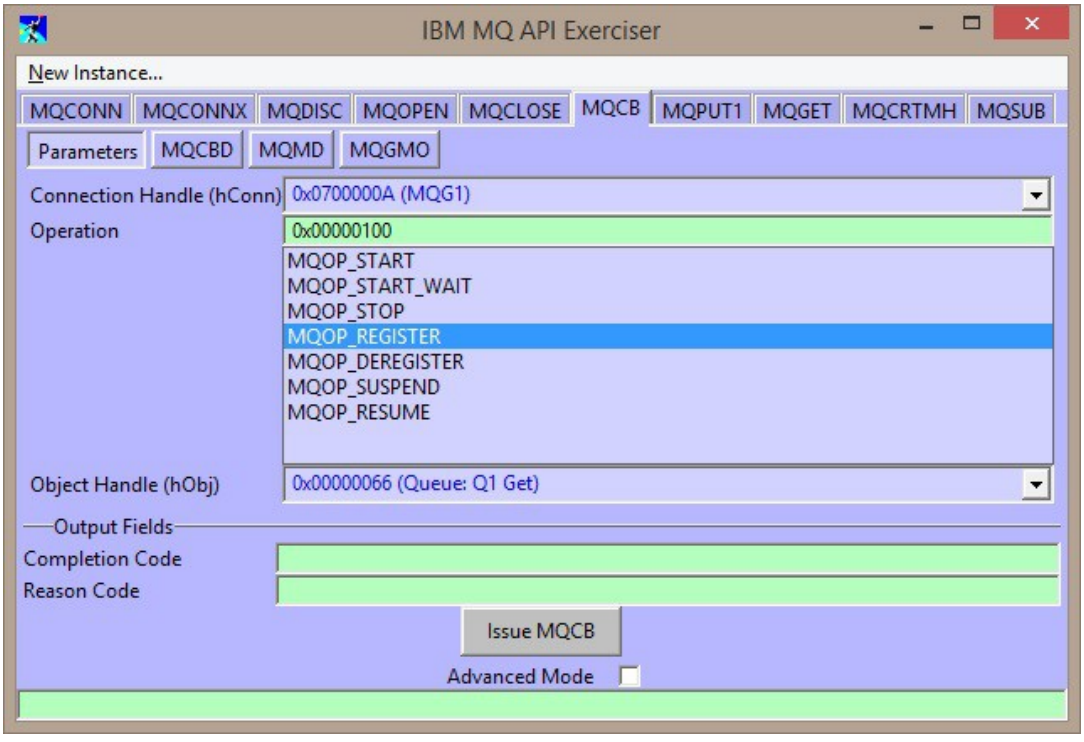
Asynchronous consume is probably one of hardest areas in the MQI to get your head around. At it's heart the concepts are simple. Rather than tying up a thread sitting and waiting for the arrival of a message, as we do with an MQGET call, we would rather MQ would call us when a message arrives. This simple idea can make programming much easier when using the MQI. The advantages of using Asynchronous consume over an MQGET call are as follows:

1. No need to tie up a thread in the application to wait for a message
2. It is simple to consume messages from multiple queues
3. No need for the application to 'guess' at the required message size to receive the next message
4. The message buffer is not allocated until required by an actual message
5. The application doesn't need to cope with the awkward situation of a message growing through DBCS conversion.
6. Very easy to stop asynchronous consumer, where as stopping an MQGET call is very awkward
7. Can register an event handler to be told about asynchronous events

So, there are many advantages. However, despite the concept being simple there are many subtle nuances in the MQI which can make programming a little tricky. The key limitation is that you can only issue one MQI call on one hConn at a time. The lines are blurred a little with verbs such as MQCTL where it is possible to issue some MQCTL calls while other MQI calls are being performed in an Asynchronous consumer callback function.

Having an API exerciser where one can try out a few of these combinations is very handy. However, as you might expect, using the API exerciser is also slightly more complicated than what we've seen before. The reason being, of course, rather than a simple 'call and return' model we now have multiple things happening at the same time.

The thing to do, of course, is to try a worked example. So, let's see whether we can receive a message via an asynchronous consumer. As before issue an MQCONN (or MQCONNX) call and then issue an MQOPEN for a queue where you open the queue using MQOO_INPUT_SHARED. Now, instead of calling MQGET we're going to register an Asynchronous consumer using the MQCB tab.



So, select 'MQOP_REGISTER' from the list of operations and choose the object handle we opened earlier. Now, on the MQCBD tab we need to choose the callback function we want to register. In a 'real' program this would be the address of your function, or the name of an entry point, that is going to process the message. MO71 provides a Callback Function pointer for you, so select that. Now, hit the 'Issue MQCB' button. If all goes well you should get a zero reason code.

So, we've registered a consumer. All we need to do now is actually start message consumption. The verb to do that is MQCTL. So, go to the MQCTL tab and issue the call with operation 'MQOP_START'. Again, we would expect a zero reason code.

What happens now depends on whether the queue we opened was empty or not. If it was empty we need to put a message to the queue to give our consumer something to consume. So, use the *'Action-Put Message'* dialog to put a test message to the queue we are listening on.

Voila! Another exerciser window! We now have a new Exerciser window with a title of 'IBM MQ API Exerciser - Call-Back Function'. So, why did that happen ? Well, this is telling us that the callback function was invoked and programmatically we are still inside it. The dialog is allowing us to view all the parameters passed to the callback and is waiting for us to hit the button to return from the callback. So, take a look at the parameters and see what you can find.

Hopefully the first thing you noticed was that the 'Buffer' contains our message. Hopefully you also noticed that the hConn passed to the callback function is the same as the one in our first exerciser dialog. This perhaps was to be expected but it's good to see it confirmed. So, what will happen if we try to use the hConn in the first dialog ? Try it.....go and issue an MQOPEN in the first dialog, without returning from the callback function. What did you find ? Probably that the call looks like it hangs right ? So, what is happening here ? MQ is waiting for the callback function to return to decide whether it should allow your MQOPEN or not. So, press the 'Return from callback' button. You will now see the first dialog display a failure reason of 'Hconn Async Active'. In other words, you can't issue an MQI call right now because that hConn is currently 'started' and being used by asynchronous consume. In fact there are very few MQI calls you can now issue.

By contrast when the callback function is invoked you can issue pretty much any verb. Let's try it, put another test message on to our test queue to get the callback function invoked. You can see the new test message but notice that this second instance of the API exerciser has the full set of MQI verbs available to it. So, try one. Open another queue or put another test message to a queue. You can see it all works fine. You can even register other asynchronous callbacks and issue starts, suspends and stops. One notable exception is MQDISC. Note that unless you are in Advanced mode the MQDISC tab isn't even shown in the Call-Back Function dialog. If you do try to issue it then you will be told MQRC_ENVIRONMENT_ERROR. This is a restriction in the MQI to prevent applications from confusing themselves.

So, anyway, there you have it. You can now try out the various different combinations of asynchronous consume and issue MQI verbs from the 'main thread' and 'consume thread' and see just how they interact. Hopefully you will find this a very useful aid in understanding, what is in MQI terms, a complicated area.

17.6. The MQI Verb set

At the time of writing the API Exerciser does not support the full MQI Verb set. The table below shows which verbs are currently supported by the API Exerciser and which are not yet supported. Comments about support for specific versions of API structures are noted earlier in the chapter as each verb is discussed.

MQI Verbs provided in the API Exerciser		MQI Verbs not yet provided in the API Exerciser
MQCONN	MQSTAT	MQBEGIN
MQCONNX	MQSUB	MQSET
MQDISC	MQSUBRQ	MQBUFMH
MQOPEN	MQCRTMH	MQMHBUF
MQCLOSE	MQDLTMH	
MQPUT	MQINQMP	
MQPUT1	MQSETMP	
MQGET	MQDLTMP	
MQCMIT	MQCB	
MQBACK	MQCTL	
MQINQ		

17.7. Advanced Mode

When Advanced Mode is checked, all verbs and options are available to you to try even if they don't make sense. For example, you can attempt to issue an MQOPEN before having issued an MQCONN. This may be helpful when exercising verbs to get particular reason codes. When unchecked, the exerciser attempts to discourage you from doing anything that will obviously bring about an error. For example, the MQOPEN tab is not visible until you have issued an MQCONN verb.

We will detail the differences when running in Advanced Mode in this section.

17.7.1. MQCONN

In addition to choosing a queue manager name from the drop-down which contains all the queue managers defined to MO71, you can alternatively type in a queue manager name and if it is unknown to MO71, an attempt will first be made to connect to it using local bindings, and if that fails, a client connection will be attempted. If the queue manager name selected from the list is a "via" location, this pattern to attempt to connect will also be used. All MQCONNX style details, for example client channel definition or connection security parameters (MQCSP) are configured in MO71. Alternatively you can use the MQCONNX tab and provide these fields directly.

17.7.2. Connection Handle fields

In MQOPEN, and indeed all verbs which have a connection handle as an input field, in addition to being able to chose the connection handles create by previous uses of the MQCONN verbs you have issued, you can also choose, "0xFFFFFFFF (MQHC_UNUSABLE_HCONN)" or type in any number yourself. These are of course only useful for testing error conditions!

17.7.3. Options fields

In advanced mode, all possible options for a particular verb are shown, even when some of them are not appropriate, for example, they will cause MQRC_OPTION_NOT_VALID_FOR_TYPE if used. Additionally, you may type in an options numeric total in the entry field above the list box instead of selecting the options you want from the list. This will highlight in the list below the field which options are chosen from the number you typed in.

17.8. Troubleshooting

17.8.1. Why are my object name drop-down lists empty?

In order to populate the drop-down list with relevant queue (or other object) names, the API Exerciser first needs to know the queue manager name. If no connection handle is selected, or the MQHC_UNUSABLE_HCONN is chosen, then there is no context within which to populate these drop-down lists.

17.8.2. Why is the object handle I just opened not in the drop-down list?

There are two main reasons for this.

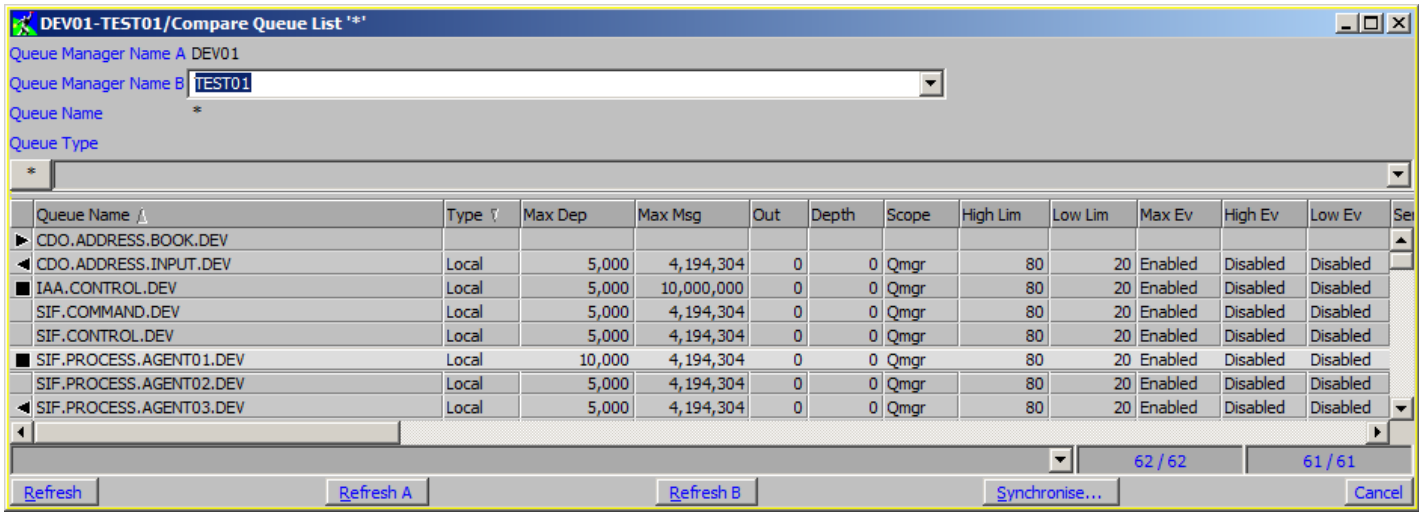
1. If the tab you are on has no connection handle selected or has a different connection handle selected, then the object handle will not be in the drop-down list until the correct connection handle is first selected.
2. If the object handle was made without MQOO_OUTPUT, it will not show up in the MQPUT object handle drop-down list since it is not valid for use with that verb. Using advanced mode will allow all object handles for this connection – even those with incorrect options - to show in the drop-down list.

17.8.3. Why is the queue I just defined not in the drop-down list?

The drop-down lists of object names are populated from MO71's database of information about the queue manager, rather than going to the queue manager and requesting a list of queue names at the point when the drop-down list needs them. This means that you will need to tell MO71 to refresh it's database of object names if you define any new queues, in order for them to show up in the list (remember that you can also just type in the queue name too). In order to refresh this database, go to the main MO71 window, select your queue manager from the list, and press File->Refresh Information (or <CTRL>+<R>).

Chapter 18. Comparing Queue Manager Objects

By selecting the *'Action-Compare Definitions'* main menu item from the main window a dialog is displayed which allows the objects from two different Queue Managers to be compared. This dialog operates in a similar way to the normal object dialogs and most of the normal dialog operations are available. However, there are a few key differences.



The first difference is that the dialog itself is not permanently tied to a particular Queue Manager. Instead there are two Queue Managers involved and these must be entered at the top of the dialog. The first Queue Manager is referred to as Queue Manager 'A' and the second one as Queue Manager 'B'. Nothing will be displayed until a response has been received from both Queue Managers. Note that the action *'Refresh'* will refresh **both** Queue Managers, actions are provided to refresh the data for just one Queue Manager as well.

Another difference is that the first column is an indicator of where the object is defined and whether the definition differs between the two Queue Managers. The icons are:-

- Left pointing triangle — Object is only defined on Queue Manager A
- Right pointing triangle — Object is only defined on Queue Manager B
- Black Square — Object is defined on both Queue Managers and they are different
- Black Diamond — Object is defined on both Queue Managers and they are different and the difference is immutable
- <Nothing> — Object is defined on both Queue Managers and they are identical

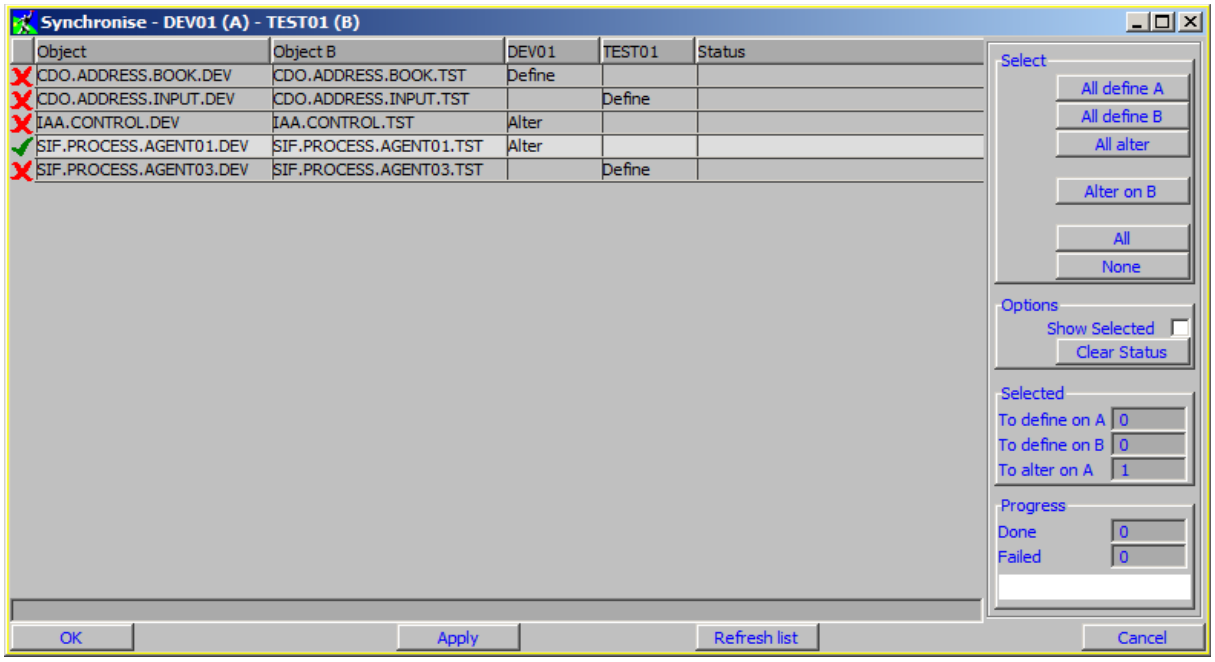
The list output itself is essentially two lists presented side by side. The lists are ordered so that the data for the same object name in the two Queue Managers are aligned horizontally. As with the other dialog lists you can change the fields displayed using the *'Alter list...'* menu option and the lists can be split into two display areas using the *'Split list'* menu option. However, there are a few new menu options which are particular to the compare dialogs.

- **Highlight difference fields**
  This is only applicable to those objects which are defined on both Queue Managers but are different. The fields which match between the two definitions are displayed using the list lowlight colour making the different fields more obvious.
- **Display only different fields**
  This option is useful to reduce the amount of information shown. When selected only those columns which contain different data between the two objects are displayed. Note that the columns displayed will probably vary greatly depending on how many objects are displayed. To further reduce the columns it may well be worth using the object name/type dialog field, entering a filter or using the *'Display objects'* menu item.
- **Display objects**
  This menu option provides a quick and simple way of displaying only a subset of the data based on which of the four categories the object falls in. Just selecting a menu item will deselect all the other items. So for example, clicking on *'Qmgr A defined only'* will only show objects which are defined on Queue Manager A. However, suppose you want to see these plus the objects defined only on Queue Manager B; by holding down the *<Ctrl>* button when selecting the menu option *'Qmgr B defined only'*, this option will be added to the current selection. In other words, the *<Ctrl>* button allows the selection status of the menu item to be toggled without affecting the selection status of the other object menu items in a similar way to selection in list boxes.

- **Use Name Matching**
  If this option is checked then MO71 will check whether the two Queue Managers being compared have defined object patterns. If they do then these patterns will be used to match an object on one Queue Manager with an object on another. For example suppose you have patterns of *.DEV and *.TST then a queue called Q1.DEV will be matched with Q1.TST on the other queue manager. In order to use name matching the format of the object name patterns of the two Queue Managers must match. This means that they must have the same number of list elements and the wildcards in the string must match. So, *.TST and *.DEV will match but *.*.TST and *.DEV does not since the first string has more wildcards.
  If this option is not checked then objects will be compared by name only.
  For a description of name patterns and their specification please see Local Object Name Pattern Specification on page 77.

18.1. Synchronise

One of the most powerful features of the compare dialog is that it allows you to synchronise the definitions of the two Queue Managers. By selecting the *'Synchronise'* menu option a dialog is presented listing the object differences.



At first glance this appears like a complicated dialog but it is relatively straight-forward. The selected objects are the only ones which will be changed. In the example only one queue will be updated, **SIF.PROCESS.AGENT01.DEV**. **IAA.CONTROL.DEV** exists on both Queue Managers so the dialog shows the action needed is to alter the definition on **DEV01** to match the definition on **TEST01** and so on. In this instance we have used name matching and the dialog shows us what the objects names would be on the two Queue Managers. If object matching is not used then the 'Object B' column will not be present since the object name used will be the same on both Queue Managers.

For objects defined on both Queue Managers clearly one could alter the definition of either to match the other. The synchronise dialog will always assume that you wish to alter Queue Manager A to match Queue Manager B. However, by pressing the *'Alter on B'* button the dialog will then alter the definitions on Queue Manager B to match the definitions on Queue Manager A. It is not possible to alter some objects on A and some on B in a single operation. If you wish to do this then you would have to select the first set of objects, make your changes, then select you different set of objects, change where the alter happens and then make the second set of changes.

The *'Selected'* group of fields shows the number of changes which will be made when the *'Apply'* or *'OK'* buttons are pressed. By selecting the *'Show Selected'* checkbox the list will offer further confirmation by only showing those objects which will be changed. Once the user is happy with the selection they should press either the *'Apply'* or *'OK'* button.

Updates are made and a progress indicator shows how far through the process the dialog is. In addition, the status column next to each object will display the result of the operation. Note that it is possible for an update to fail for a variety of reasons. For example, a queue may be defined when the initial object list is obtained but by the time the synchronisation command is issued someone else has independently deleted the object. Clearly the alter command will fail if there's no base object to alter.

18.2. Queues

Temporary queues are not compared since, by definition, these queues are only temporary. Naturally the model queues, which define the temporary queues in the first place, can be compared and synchronised.

18.3. Channel Authentication

Synchronising Channel Authentication objects are handled slightly differently to other MQ objects. For the other objects an 'Alter' will make the objects identical. However, for Channel Authentication the effect of an 'Alter' will only add the definitions from one object to another. For security purposes the entries, for example in the Block User list, will not be removed.

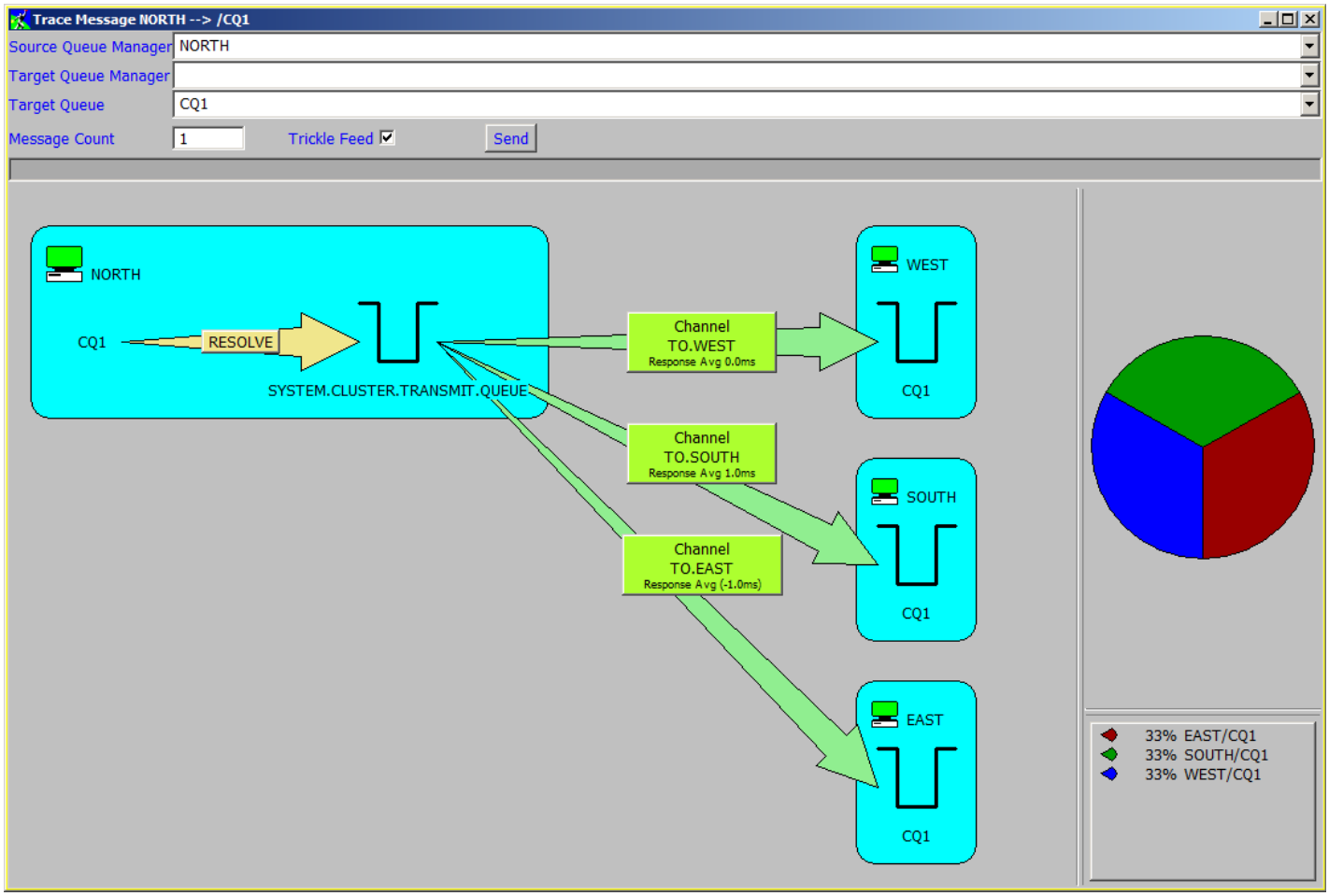
18.4. Subscriptions

When comparing subscriptions only subscriptions which contain a 'subname' may be compared and, for obvious reasons, only durable subscriptions may be synchronised.

Chapter 19. Trace Message

The IBM MQ product has the ability to allow an application to send a message through the system and ask for reports to be sent back to a reply queue to notify the application of the route the message took. MO71 uses this behaviour to display, in a digram, the path a message took through the MQ network. In addition MO71 will measure the response time for these replies and indicate these times on the diagram which can give an indication of the current messaging performance.

By selecting *Action-Trace Message...* on the main window and typing in a few values in the given fields you can easily generate a diagram such as the one below.



This diagram shows a number of messages being put from the NORTH Queue Manager to the queue 'CQ1'. CQ1 is a cluster queue which is hosted on Queue Managers, SOUTH, WEST and EAST. As a consequence, as a number of messages are sent, MO71 will get responses from the different queue managers and reflect these responses in the diagram.

The trace message feature can be useful for a number of reasons:

- Show whether a queue resolution is happening correctly
- Show what path(s) a message may travel through the MQ network
- Show the percentage split of messages to multiple destinations in, for example, an MQ cluster
- Given an indication of the response time sending to various locations in the MQ network
- Given an indication of the current cost of a Channel link

The operation of the dialog is very simple. Each time the user presses the 'Send' button MO71 will send one or more messages to the target queue and report on the path the message took. The user need not be concerned about affecting the operation of any running applications since no messages will actually appear on the application queues. To show a diagram such as the one above clearly MO71 needs to send multiple messages. To make this simpler the dialog allows the user to specify the number of messages

to send. By default it is just a single message but setting it to a larger value allows the diagram to be created with just a single click of the 'Send' button.

In addition to the 'Message Count' the user can also specify whether the messages should be 'Trickled'. The choice about whether to use this option or not depends on why you are using the trace message dialog. If you are only interested in the path that messages take then there is no real reason to trickle messages. However, if you are interested in the time it takes for the messages to be sent and returned from the target queues then it is not a good idea to flood the transmission queue with messages.

Consider an example when you send 50 messages. If you just put 50 messages on the transmission queue as fast as possible then the processing of the last of those messages will be significantly delayed by the previous 49 messages. Trickling the messages will put the messages at only 10 messages per second which allows the channel to process each message before the next message arrives on the transmission queue. So, as a general rule, if you are interesting in the response time ensure that the trickle message option is selected. If you are only interested in the message path then switching off the trickle option will speed up processing.

The dialog provides a number of different options. These are selectable by pressing the context mouse button over the diagram.

The options are described below.

19.1. Commands

When viewing the diagram display type, you may right click inside the box that represents a queue manager and the Commands menu will allow you to run any command against that queue manager.

19.2. Send

This menu option does the same job as the “Send” button.

19.3. Open <object type>

When viewing the diagram display type, you may right click on any of the labels which reference the objects being displayed and, depending on the object type, choose one of the following menu options to open an individual object dialog to view the object.

Menu name	Meaning
Open Queue	View the queue definition
Open Queue Manager	View the queue manager object
Open Sending Channel	View the channel definition on the sending queue manager
Open Sending Cluster QMgr	View the cluster queue manager on the sending queue manager – used when the sending channel is a cluster-sender, and so there is less likely to be a channel object to view.
Open Receiving Channel	View the channel definition on the receiving queue manager

19.4. Display Type

There are actually four different formats of data.

Display Type	Meaning
Diagram	This shows a diagram of the style shown above
Nodes	This display type shows a linked node display showing each trace action reported.
Raw	This shows a textual display of the trace message response records
Consolidated	This shows a consolidated view of the raw data. In other words duplicate records are consolidated with a count.

19.5. Display Options

A number of display options are available to change the format of the display.

Display Option	Meaning
Queue Manager Icon [32]	This option controls whether the Queue Manager icon is displayed
Response Times [32]	This option controls whether the message response times are shown in the diagram. Any displayed time is the time taken from the original MQPUT of the message to receiving the trace message report. The response time values shown are: • Average Response Time • Last Response Time • Maximum Response Time • Minimum Response Time All timings are in milliseconds. Note that if channels are not running when you first issue the trace requests then you could get an unrepresentative large response time for the first message as the channels are started. For this reason is it sensible to check that you don't have too large a spread between your minimum and maximum values. If required you can issue '*Delete All Data'* to then get a new set of data.
Channel Response Times [32]	This option controls whether the time across the channel link is reported. Note that the time should not be treated literally but used as an indication of the performance across this link. It is calculated by subtracting the response time for the sending channel GET call from the receiving channel PUT call. Note that these response times include the time for the return trip messages. Consequently the channel response time value will also include the return channel cost. There are also many other factors which can affect response time such as transmission queue depth, current message profiles and general system load. However, although it should not be taken literally, it can be a useful indication of how long messages are taking to cross back and forth between these two Queue Managers particularly if the response times are large.
Counts [32]	This option controls whether the message counts are shown in the diagram.
Use small font [32]	By default the additional information such as the message count and response times are displayed in a smaller font than the queue name. However, by deselecting this option these values are shown in a full size font.
Route in key	This option controls whether the delivering Queue Manager is also shown in the pie chart key. This can be useful if there are different routes which arrive at the same queue.

In addition to the above display options the colours of the display can also be changed using the Colour Dialog described on page 229.

19.6. Show Pie Chart

If a message trace results in messages being put to different queues then it can be useful to display a pie chart to show the percentage split. The most likely cause of messages going to different destinations is clearly when injecting messages into an MQ cluster.

As you can see from the screen-shot the pie chart will only show you the percentage of messages sent to each destination. If one or more of the messages do not make it to a target destination then the graph will show a 'No Response' segment. Note that that this could indicate a number of potential problems such as the return path for the report messages is not defined, channels are not running, trace message is switched off or the user does not have sufficient authority.

[32]Only applies to Display Type Diagram

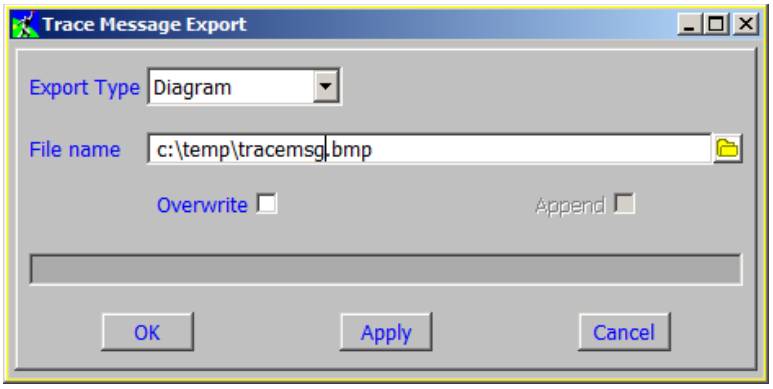
19.7. Pie Chart Labels

A number of options are available to change how your pie chart looks.

Pie Chart Labels	Meaning
Show Labels	Labels will be positioned on the pie-chart drawing in addition to the legend below the pie-chart.
Key	Show the key value, usually the target queue manager and queue name.
Value	Show the number that makes up the pie-chart segment, that is the number of messages that were targeted to this queue.
Total	Show the total number of messages for the whole pie chart. This is only shown if “Value” is also selected. You will see it shown as “Value / Total”
Percentage	Show the percentage that this pie-chart segment represents of the whole pie-chart.

19.8. Export

Selecting this option will display a simple dialog such as the one below:



This dialog allows to export the Trace Message data either as a BMP file or as a text file depending on the export type you choose. This is useful if you want to save the data or would like to import the diagram into a document.

There are two options:

- **Overwrite**
  To avoid you accidentally writing over an existing file the export will check that the file does not exist before the export is done. If the file does exists then a confirmation message will be shown asking whether MO71 should overwrite the file. If you are sure that you want the file to be overwritten then you can check the overwrite option.
- **Append**
  This option is only applicable for a text file export which applies to the 'Raw' and 'Consolidated' data. If checked then any exported data will be appended to the end of any existing file.

19.9. Copy text to Clipboard

When viewing the diagram display type, you may right click on any of the text labels and choose this menu option to copy the text to the clipboard.

19.10. Delete

Any collected data will be deleted if the dialog is closed. However, it can be useful sometimes to manually delete the collected data.

Menu name	Meaning
Delete Counts	This option will delete all of the counts and response time data but leave the basic diagram structure. This can be useful if you believe that the current set of response times have been skewed for some reason, for example because the channels were not running at the time of the first message.
Delete All Data	This option allows you to delete all of the trace data so far collected. The data will automatically be deleted if the source or target Queue Manager or Queues are changed. This option is useful if you believe that diagram should be recalculated for a target because something has changed which will affect message flow, for example changing the status of channels or a cluster weighting.

19.11. Limitations

In order to use trace message there are some configuration limitations you must conform to:

- Trace message only currently works with point-to-point messaging and not publish/subscribe.
- There must be a channel return path for responses to be received.
- This feature depends on receiving trace message reports. As a consequence all Queue Managers in the path must have ACTIVREC(MSG) set in the Queue Manager definitions.
- The user must have the authority to inquire attributes of the local transmission queues.
- All Queue Manager and Channel names must be unique.
- Loops are not supported
  The diagram is constructed by matching channel and Queue Manager names, it is not a full-path comparison. Consequently the diagram function will not cope with passing through the same node multiple times.

Chapter 20. Web Access

MO71 can be configured to listen for connections from browsers and is capable of returning the information about the configured queue managers as web pages. Currently this information is only read-only, no support is provided for issuing commands other than display commands. By default MO71 does not listen for HTTP or HTTPS connections. In order to enable the feature start the listener from the HTTP tab in the preference dialog (please see HTTP on page 210 for more information). The number of different devices which can connect concurrently is different depending on the type of licence that MO71 is running under. An Emerald licence allows just a single device to connect at any one time, where as a Diamond licence is unlimited.

The most important thing to ensure is that the 'Root Path' is set to the directory where the HTML files which came with MO71 are located. MO71 is capable of delivering virtually any files and can therefore also be used as a simple HTTP server.

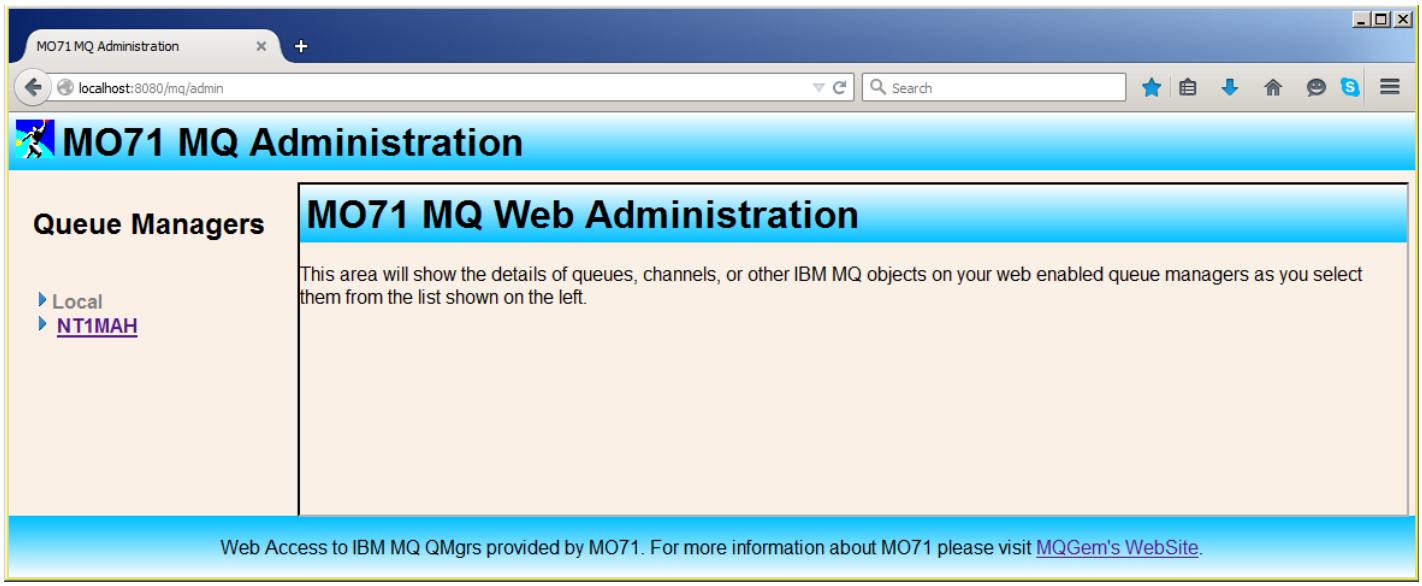
20.1. Getting Started

The first thing to do is to choose which of your Queue Manager locations you wish to be accessible from the browser. For those locations open the location dialog and select the **'HTTP Allowed'** in the Options tab. For one of the Queue Managers you can also select the **'HTTP Default'** option. Being the HTTP Default location means that any HTTP request which doesn't explicitly state the queue manager name will get data from this location. You can also choose whether browsing of queues is supported for this Queue Manage by setting the **'HTTP Allow Browse'** option.

Once the Queue Managers have been modified you should now start the HTTP or HTTPS Listener. This is done in the HTTP tab of the Preferences Dialog described on page 210. Note that you can configure the HTTP listener to start as soon as MO71 itself is started.

Once the listener is started start your browser and enter the URL http://<Host Address>/**mq/admin**

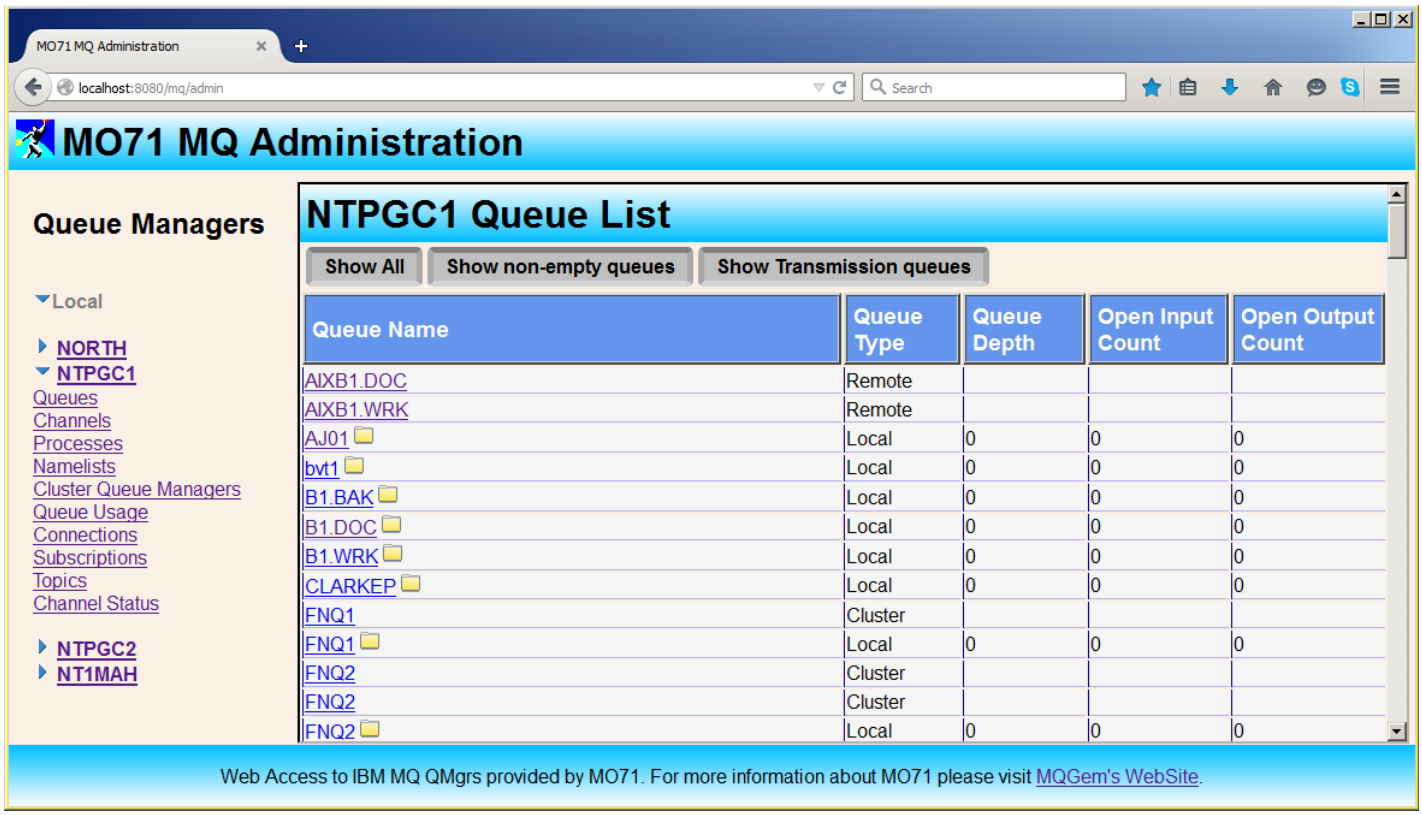
Depending on which Queue Managers you have selected your browser should now look something like this:



As you will see later the HTTP output is highly tailor-able however this is default output. The output format is governed by the content of the file **mqadminindex.html** in your HTML root path. By editing this file you can add your own links, add installation logos or provide your own pre-defined data queries. You can, of course, also change the fonts, font sizes, colours and screen layout by modifying the style sheet. The style sheet used by default is in <HTML-root-path>/styles/simple.css

This default layout will show all the Queue Managers, which have been chosen to be accessible by the browser, in a pane on the left-hand of the screen. On the right-hand side a panel will show the selected content. When you first access the page the content panel will show the content of **frameinit.html**. However, if we select links in the left-hand pane we will see the content of this pane changing.

For example, if you expand a Queue Manager twisty and select the Queues link you should see an output something like this:



The links in the left-hand pane are constructed using the following form:

http://<Host Address>/**mq/admin/**<Object Type> '/' [<Queue Manager Name>] ['/' <Object Name>] ['?' <Dialog Parameters>]

As mentioned before the Queue Manager part of the URL is optional but if not specified there must be a location which is identified as the HTTP default location.

The object type indicates the type of information which should be returned in the page, the table below shows all the available types and gives an example of the full URL syntax. For list displays the columns shown will be the ones selected in the default list for that object type. These can be set from the main window by selecting the *View-Set Default Lists-...* menu. Additional columns can be displayed if required by using the ***showcol()*** filter function.

As you can see from this example it also very easy to include buttons to modify the displayed data. By default in the queues display, buttons are added which will show just queues that contain messages and transmission queues. However, adding your own buttons which could show groups of queues or queues with certain characteristics would be very easy. For the queues display all that is needed is to modify the **queues.html** file.

By default a list of objects uses **objects.html** and a singe object uses **object.html**. However, each of the objects and object lists can have an equivalent HTML file that you can create and/or modify, as shown by **queues.html**. In this way you can add your own links and/or logos and integrate the MQ data in to your own Web administration pages. For a list of the files please see HTML Files on page 158.

If you are requesting the data to be returned in XML format then the equivalent files with an XML filetype should be used. For example to modify the response to an XML queues request then the file **queues.xml** should be edited.

20.2. Examples URLs

Examples of the URLs are given in the next two tables for each type of display. Most of the links should work against your Queue Manager but some won't because they reference an object name that you won't have in your definitions. Note also that the examples assume that MO71 is configured to listen on the default HTTP port of 80. If a different port is selected then all URLs will need to be modified accordingly. If you decide to use HTTPS then the links should also be modified to use that specific protocol, i.e. `https://...` instead of `http://...`

MQ Commands

These are the URLs for MQ commands.

Display Type	Example URL
AMQP Channel Status	http://localhost/mq/admin/AMQPchs//NTPGC1.NTPGC2
AMQP Channel Status List	http://localhost/mq/admin/AMQPchses
Application List	http://localhost/mq/admin/Apps
Auth Info	http://localhost/mq/admin/Authinfo//SYSTEM.DEFAULT.AUTHINFO.CRLLDAP
Auth Info List	http://localhost/mq/admin/Authinfos
Authorisation Record List	http://localhost/mq/admin/Authrecs
CF Struct	http://localhost/mq/admin/CFStruct//APPLICATION1
CF Struct List	http://localhost/mq/admin/CFStructs
CF Struct Status	http://localhost/mq/admin/CFStatus//APPLICATION1
CF Struct Status List	http://localhost/mq/admin/CFStatuses
Channel	http://localhost/mq/admin/Channel//SYSTEM.DEF.SVRCONN
Channel List	http://localhost/mq/admin/Channels
Channel Auth List	http://localhost/mq/admin/Chlauths
Channel Status	http://localhost/mq/admin/Chstatus//NTPGC1.NTPGC2
Channel Status List	http://localhost/mq/admin/Chstatuses
Cluster Queue Manager	http://localhost/mq/admin/Clusqmgr//NTPGC2
Cluster Queue Managers	http://localhost/mq/admin/Clusqmgrs
Connection	http://localhost/mq/admin/Conn//414D51434E5450474331202020202020205E28FF4D20015301
Connections	http://localhost/mq/admin/Conns
Console	http://localhost/mq/admin/Console//0
Console List	http://localhost/mq/admin/Consoles
Message	http://localhost/mq/admin/Msg//SYSTEM.ADMIN.QMGR.EVENT?pos=0
Message List	http://localhost/mq/admin/Msgs//SYSTEM.ADMIN.QMGR.EVENT
MQTT Channel	http://localhost/mq/admin/MQTTChl/TEST
MQTT Channel List	http://localhost/mq/admin/MQTTChls
MQTT Channel Status	http://localhost/mq/admin/MQTTChs//TEST
MQTT Channel Status List	http://localhost/mq/admin/MQTTChses
Namelist	http://localhost/mq/admin/Namelist//SYSTEM.DEFAULT.NAMELIST
Namelist List	http://localhost/mq/admin/Namelists
Process	http://localhost/mq/admin/Process//SYSTEM.DEFAULT.PROCESS
Process List	http://localhost/mq/admin/Processes
Queue	http://localhost/mq/admin/Queue//SYSTEM.DEFAULT.LOCAL.QUEUE
Queue List	http://localhost/mq/admin/Queues
Queue Manager	http://localhost/mq/admin/Qmgr
Queue Manager List	http://localhost/mq/admin/Qmgrs
Queue Status	http://localhost/mq/admin/Qstatus//SYSTEM.ADMIN.COMMAND.QUEUE
Queue Usage List	http://localhost/mq/admin/Qusage
SMDS	http://localhost/mq/admin/SMDS//MQG1?cfstruct=APPLICATION1
SMDS List	http://localhost/mq/admin/SMDSs//*?cfstruct=APPLICATION1
SMDS Connection	http://localhost/mq/admin/SMDSConn//MQG1?cfstruct=APPLICATION1
SMDS Connection List	http://localhost/mq/admin/SMDSConns//*?cfstruct=APPLICATION1
Storage Class	http://localhost/mq/admin/StgClass//DEFAULT
Storage Class List	http://localhost/mq/admin/StgClasses
Subscription	http://localhost/mq/admin/Sub//bob
Subscription List	http://localhost/mq/admin/Subs
Topic	http://localhost/mq/admin/Topic//SYSTEM.BASE.TOPIC
Topics	http://localhost/mq/admin/Topics
Topic Status	http://localhost/mq/admin/topstatuss///Price/Fruit
Topic Status List	http://localhost/mq/admin/topstatuss

20.2.1. MQEV Commands

The following URLs can be used can be used for any Queue Manager with an associated MQEV processor. For more information about MQEV please see “Chapter 15. MQEV Event Monitoring” on page 123.

Display Type	Example URL
MQEV Processor	http://localhost/mq/admin/ev
MQEV Processor List	http://localhost/mq/admin/evs
MQEV Stream	http://localhost/mq/admin/evstream//$EVENTS?type=events
MQEV Stream List	http://localhost/mq/admin/evstreams
MQEV Stream Status	http://localhost/mq/admin/evstrmst//$EVENTS?evqmgr=NTPGC1;evtype=events
MQEV Stream Status List	http://localhost/mq/admin/evstrmsts
MQEV Event Queue	http://localhost/mq/admin/evqueue//SYSTEM.ADMIN.QMGR.EVENT
MQEV Event Queue List	http://localhost/mq/admin/evqueues
MQEV Queue Manager	http://localhost/mq/admin/evqmgr//NTPGC1
MQEV Queue Manager List	http://localhost/mq/admin/evqmgrs
MQEV Alert	http://localhost/mq/admin/evalert//1
MQEV Alert List	http://localhost/mq/admin/evalerts
MQEV Event	http://localhost/mq/admin/event//00003994?evqmgr=NTPGC1;evstream=$EVENTS
MQEV Events List	http://localhost/mq/admin/events
MQEV Queue Statistics	http://localhost/mq/admin/evstatqs
MQEV MQI Statistics	http://localhost/mq/admin/evstatmqis
MQEV Channel Statistics	http://localhost/mq/admin/evstatchls
MQEV Queue Accounting	http://localhost/mq/admin/evacctqs
MQEV MQI Accounting	http://localhost/mq/admin/evacctmqis

20.3. Dialog Fields

It is possible to qualify the URL by specifying the values of the dialog fields.

- http://localhost/mq/admin/Queues/?qtype=model
  Will show the list of model queues on the default HTTP queue manager.
- http://localhost/mq/admin/Channels/?type=svrconn
  Will show all the SVRCONN channels.

In addition there are other parameter values you can specify which modify the data retrieved or how it is displayed. These are equivalent to setting the same option in the message dialog. The list of parameter fields you can add are:

Parameter	Meaning
filter	to set the value of the filter field on list dalogs
pos	to set the required position on the queue when browsing
xmlauto	display xml formatted if recognised
xmlshort	display xml in short form
hex	display message data in hex
formatted	display message in formatted form
mqmd	display the message descriptor
nomqmd	display the message without the message descriptor
low	display low level of detail
med	display medium level of detail
high	display high level of detail
offset	display offsets next to any hex data
msgrange	display a particular range of messages when browsing
maxmsgsize	to set the maximum message size which can be browsed
search	to set the search string for the message browse
message	to return the full message on browse regardless of size
htmlfile	This parameter allows you explicitly state the HTML file template which should be used as the basis for the response. The file should be specified without the file type. For example, htmlfile(msghi). If the file is not found MO71 will resort to using the best template if can find for the type of response.

Essentially any dialog field value can be set by sending the name of the field and an appropriate value.

20.3.1. Examples

- http://localhost/mq/admin/Queues/?filter=”cur > 0”
  Will show only queues with messages on them.
- http://localhost/mq/admin/Msg//SYSTEM.DEFAULT.LOCAL.QUEUE?pos=1;mqmd
  Will show the first message on the queue along with the message descriptor.
- http://localhost/mq/admin/Msg//SYSTEM.DEFAULT.LOCAL.QUEUE?pos=1;mqmd;high
  Will show the same information but with ‘high’ detail.
- https://localhost/mq/admin/topstatuss/?type=sub
  Will show topic status information about subscribers. Equivalent to `DISPLAY TPSTATUS('#') TYPE(SUB)`
- http://localhost/mq/admin/cfstatuses/?type=smds
  Will show CF Structure Status of type SMDS. Equivalent to `DISPLAY CFSTATUS(*) TYPE(SMDS)`

20.4. HTML Files

Although the program itself generates HTML constructs to report the data, it embeds this data inside HTML files stored on disk. This means that by altering the contents of the HTML files you can achieve a significant amount of customisation about the way the data is reported. For example, colour schemes, layout, logos and additional links can be added. You may also explicitly state the HTML file you wish MO71 to use in the link URL using the htmlfile keyword.

The HTML files supplied with MO71 are:-

- **index.html**
  This is the page that would be delivered for the URL http://localhost
- **mqadminindex.html**
  This is the page that would be delivered for the URL http://localhost/mq/admin
- **mqadminnav.html**
  This is the main navigation pane if you choose to use the iframes main index.
- **frameinit.html**
  This html file contains the initial content that is shown in the data frame.
- **object.html**[33]
  This is the default html template which is used whenever a single object is returned and there isn't an html file defined for that particular object type.
- **objects.html**[33]
  This is the default html template which is used whenever a list of objects are returned and there isn't an html file defined for that particular object type.
- **msg.html**
  This is the html template which is used whenever a single message is displayed.
- **<object>.html**[33]
  These templates can be used to show a different look and feel for each object type. For example, by having a **channels.html** the display just for the channel list can be changed.
- **errors.html**
  This template will be displayed for general errors, for example queue manager not available.
- **error<error number>.html**
  By having html files with these names you can report back your own error messages for HTTP return codes. For example, a file **error400.html** will be returned if the inbound URL is badly formed.
- **error_no_browse.html**
  This file is sent if the user specified a browse URL and the Queue Manager does not allow browsing. If this file does not exist then MO71 will send a standard 404 response.

[33]Note that the same file name but with a filetype of XML should be used if the request is for an XML response.

20.4.1. HTML Inserts

In addition to being able to modify the html files themselves you can also use inserts in the html files. These are:-

- **%[content [msglinkparms=(<*value*>)]**
  The point at which the actual content of the page should be inserted. For example, in an object html file it's where the table containing the object definition should be placed. The tables are added with a specific CSS class name of **mo71listtable** and **mo71objecttable** depending on whether it is a list or a single object being returned. These class names means that the look and feel of the table data can be more easily changed. For example, suppose we wish the links in the tables to be displayed using the foreground colours specified in the filter rather than the default link coloured configured in the browser. We could achieve this by putting the following line in the style sheet.

  ```
  .mo71listtable a {color:inherit;}
  ```

  The optional **msglinkparms** parameter allows you to specify the parameters which should be used for any message browsing links. This is useful, for example, if you would like to control whether the link would should show a formatted message, the message descriptor or the message detail. Some examples would be:

Example	Meaning
%[content msglinkparms=(hex)]	Show messages in hex
%[content msglinkparms=(high;mqmd)]	Show message in high detail, including the message descriptor
%[content msglinkparms=(filter="htmlfile("msgshi")")]	Useful for the 'queues.html' file this parameter controls which HTML file should be used to generate the response.

- **%[curr]**
  This insert is replaced by the URL of the current message minus any clashing fields that may follow.
  For example, you may code something like **High Detail**. In this case the insert is replaced with the current URL minus any parameters which may clash with the property 'high'. So, suppose the current URL is something like http://localhost:8080/mq/admin/msg/MQ800/Q1?;mqmd;low;hex;pos=10 then the resulting URL would be http://localhost:8080/mq/admin/msg/MQ800/Q1?;mqmd;hex;pos=10;high. Note that all the parameters are maintained except the low' parameter which would clash with the 'high' parameter in the template.
  This insert should only be used in the msg.html template.

- **%[currpos]**
  This insert is replaced by the URL of the current message minus all fields except the message position.
  This insert should only be used in the msg.html template.

- **%[errormsg]**
  This insert is replaced by the error message. This should be used in the error.html template.

- **%[include <filename>]**
  The text will be replaced with the identified include file. This is useful if you want the same html to be included in many different html templates, for example to include a logo or stylesheet. Include files can, themselves, include other files.

- **%[loc]**
  This insert is replaced by the location name of the queue manager.

- **%[next]** and **%[prev]**
  These inserts are replaced by the URL of the next or previous message. This should only be used in the **msg.html** template.

- **%[objtype]**
  This insert is replaced by the name of the object type. For example 'queue'.

- **%[qm]**
  This insert is replaced by the name of the queue manager.

20.4.1.1. %[qmlist] HTML Insert

This insert has the following form. It consists of essentially four positional sections separated by ':' characters.

%[qmlist [<qm name mask>] [':' <Options>] [':' <Contents>] [':' <Parameters>]]

This insert will be replaced by a list of queue manager object links with twisties which contain links to the various queue manager objects. Use of this insert requires HTML5 support in your browse for `<summary>` and `<details>` tags[34]. See the contents of **mqadminindex.html** for examples. The insert can be configured to suit a particular installation.

Section 1 [<qm name mask>]

The first section specifies a filter string which controls which Queue Managers should be included. If not specified then all HTTP queue managers will be shown. Wildcards can be used where '*' matches zero or more characters. '?' matches any single character. By default the string is matched against the Queue Manager name however, in the next section you can specify that the filter should be matched against the location name.

Here are some examples:

qmlist[] example	Meaning
%[qmlist]	show all queue managers selected for HTTP access
%[qmlist %[qm]]	show just the queue manager being displayed**.** This is useful to show additional links to other objects for this queue manager.
%[qmlist *WIN*]	show all queue managers with 'WIN' in their name.

Section 2 [':' <Options>]

The second section specifies overall options which control to display of the %[qmlist] output. You can specify none or all of the options in a list separated by commas or spaces.

Option	Meaning	Parameter Group
qm	Display the QM name This is the default is no display parameters are specified	Display
loc	Display the location value	Display
grp	Use the QM group structure	Display
floc	The first section filter should be matched against location names	Filter
fnet	The first section filter should be matched against network names	Filter
fqm	The first section filter should be matched against Queue Manager names. This is the default if no filter parameters are specified	Filter

Here are some examples:

qmlist[] example	Meaning
%[qmlist : loc]	show all queue managers and use the location name
%[qmlist : qm, loc]	show all queue managers and display both Queue Manager name and location.
%[qmlist *WIN* : grp]	show Queue Managers with 'WIN' in their name and include group hierarchy.
%[qmlist *(Dev)* : floc]	show all Queue Managers with '(Dev)' as part of their location definition

[34]Or alternatively, the use of simple Javascript which enables expanding areas, an example of which is included in the old version of the template. See 27.2.1. Twisties not working 243 if you can't make use of HTML 5.

Section 3 [':' <Contents>]

The third section specifies what links are contained in the twisty for each location. Again these are specified in a list which can be comma or space separated. The values can be:

Identifier	Meaning
acm	MQEV MQI Accounting
acq	MQEV Queue Accounging
ai	Auth Info Record
all	All objects
allev	All MQEV objects
allmq	All MQ objects
app	Applications
ar	Authorisation Record
cf	CF Struct
cfs	CF Struc Status (Type Summary)
cfsb	CF Struc Status (Type Backup)
cfsc	CF Struc Status (Type Connect)
cfss	CF Struc Status (Type SMDS)
cha	Channel Auth Records
chl	Channels
chs	Channel Status
chsa	AMQP Channel Status
ci	Comm Info Records
con	Connection Applications
cona	Connection All information
conh	Connection Handles
cqm	Cluster Queue Managers
def	Default list
eva	MQEV Alerts
evm	MQEV Queue Managers
evp	MQEV Processor

Identifier	Meaning
evq	MQEV Queues
evs	MQEV Streams
evst	MQEV Stream Status
evt	MQEV Events
file[=filename]	Provide a file of specific links to be used
l	Listeners
ls	Listener Status
nl	Namelist
prc	Processes
pub	Publishers
q	Queues
qs	Queue Handle Status
qsq	Queue Usage
sbs	Subscription Status
stc	MQEV Channel Stats
stg	Storage Classes
stm	MQEV MQI Stats
stq	MQEV Queue Stats
sub	Subscriptions
sv	Services
svs	Service Status
top	Topics
tps	Topics Status
tpsp	Topic Status (Type Pubs)
tpss	Topic Status (Type Subs)

SMDS and SMDSCONN commands are not part of the above table because those commands cannot list all objects. The commands must be qualified by a CFSTRUCT name. Instead SMDSCONN is linked from a CFSTATUS display.

The identifier '**file**' allows you to provide a snippet of HTML to be inserted into the twisty for the queue manager. The filename is optional, and if omitted, a file called ***qmlist.html*** will be used. Otherwise, you can specify the name of an HTML file without it's extension. The '**file**' identifier can only be used once in the list, and any additional occurrences of it will be ignored.

If the third section is not specified a default set of links will be shown. You can choose the default list of links using the identifier '**def**' and then add other identifiers after it in the list.

For example we could specify:

qmlist[] example	Meaning
%[qmlist :: q,chl]	show queues and channels
%[qmlist : grp: q,qs]	show queue and queue status and use group hierarchy
%[qmlist *WIN* : grp,qm,loc : chl,chs]	show just Queue Managers with 'WIN' in their name, include group hierarchy but only show channels and channel status links.
%[qmlist MQG*: : def,ar,file=mqglinks]	Show just Queue Managers with names starting MQG, and populate the twisty with the defaults links, the AuthRecs link and the links in ***mqglinks.html***.

In the above final example, ***mqglinks.html*** might look something like this:-

```
<a href="/mq/admin/authrecs/%[qm]?objtype=qmgr" target="dataframe">QMgr Auth Recs</a><br>
<a href="/mq/admin/authrecs/%[qm]?objtype=queue" target="dataframe">Queue Auth Recs</a><br>
```

Section 4 [':' <Parameters>]

The fourth, and final, section is reserved for additional parameters. Currently only one parameter, **linkhtml**, is supported. This parameter allows you to specify additional HTML text which will be included in the generated links.

qmlist[] example	Meaning
%[qmlist ::: linkhtml=(target="_blank")]	The link will be displayed in a new browser tab
%[qmlist ::: linkhtml=(target="mainpanel")]	Show the content in the 'mainpanel' frame
%[qmlist ::: linkhtml=(class="highlight")]	Add a CSS class name to allow the style of the link to be set.

20.5. Using HTTPS

Using the MO71 listener with the HTTP protocol is as simple as selecting the “Enable HTTP” radio button and pressing the “Start” button. If you instead want to use the HTTPS protocol, there are a few more things you need to configure which will be covered in this section.

To use HTTPS, MO71 will need a certificate to identify the web server. Browsers will be connecting to the web server, so it is essential that the certificate includes a ***Subject Alternative Name*** extension that matches the webpage URL. If it doesn't browsers will claim the website is unsafe, and you'll have to click on various advanced buttons in order to bypass the warning.

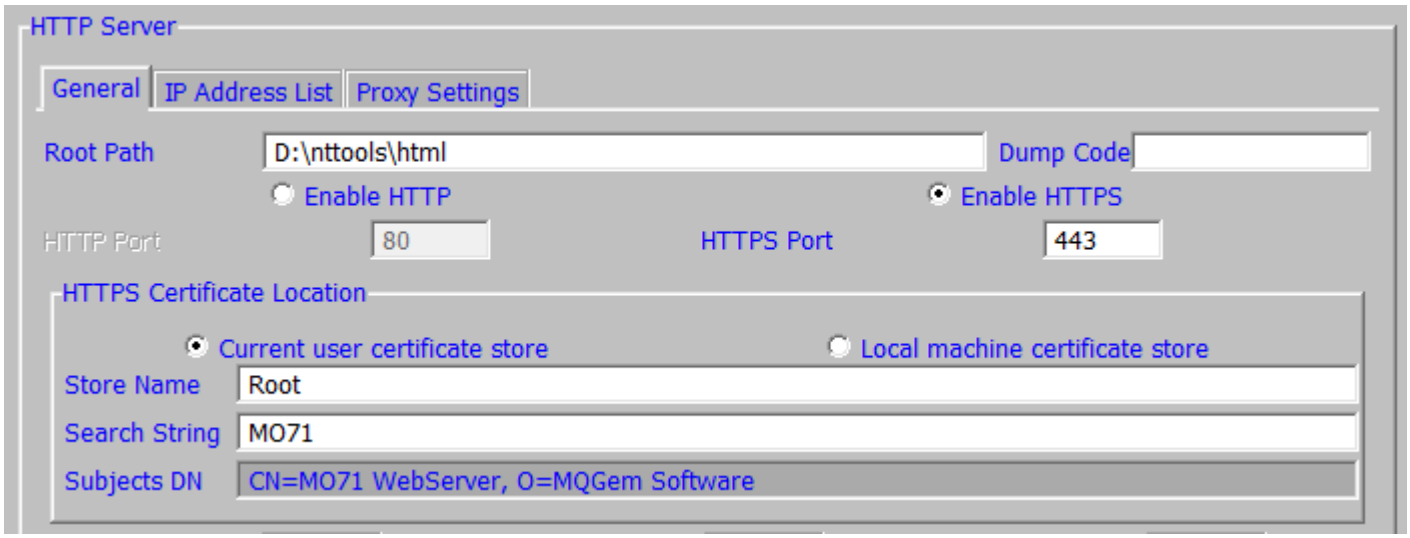
MO71 expects the certificate to be used for the web server to either be in a:-

- **Local machine certificate store**
  This type of certificate store is local to the computer and is global to all users on the computer. Handy if you're running MO71 as a background service.
- **Current user certificate store**
  This type of certificate store is local to a user account on the computer.

If you wish to read more about the various different locations of Windows certificate stores, see https://docs.microsoft.com/en-us/windows/win32/seccrypto/system-store-locations

Regardless of whether you choose a certificate store that is machine wide or restricted to your user account, you must also provide MO71 with the name of certificate store where the certificate is located. Predefined certificate store names include, ***MY***, ***Root***, ***Trust*** and ***CA***.

To locate the specific certificate to be used to identify MO71's web server, provide a portion of the Subjects DN in the 'Search String' field, enough to uniquely identify the certificate in the store you have nominated. Upon successfully finding the certificate, MO71 will display the Subjects DN of the certificate it found in the 'Subjects DN' field.



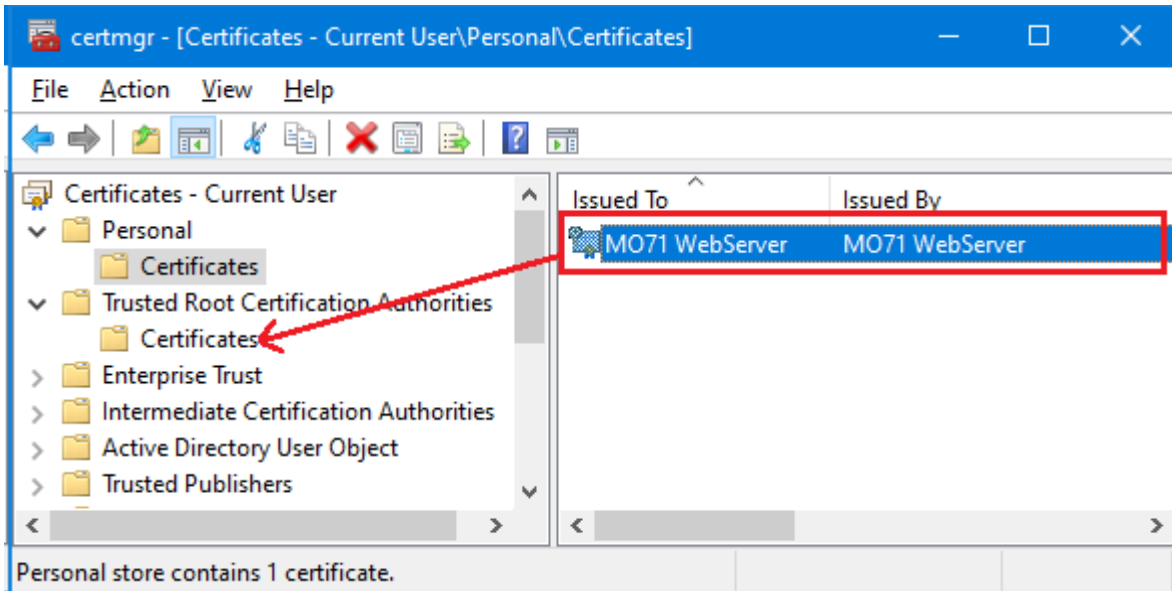
MO71 does not require you to configure which TLS protocols or ciphers suites are enabled in its web server, but instead runs with the settings configured in your Windows O/S, negotiating the best cipher available based on those enabled on your system. It is not expected that you need to change these, but more details can be found in the Advanced Configuration section below should you need to.

20.5.1. Trying it out with a self-signed certificate

If you want to try this out with a simple self-signed certificate, the following Powershell command will create a certificate and place it in your Current User certificate store called “My” store. The `-DnsName` parameter specifies one or more DNS names to put into the ***Subject Alternative Name*** extension of the certificate. What you supply here should be the same as the way you are addressing the webpage, or the earlier mentioned complaints will be made by your browser.

```
New-SelfSignedCertificate -Subject "CN=MO71 WebServer,O=MQGem Software"
-DnsName "localhost" -CertStoreLocation "cert:\CurrentUser\My"
```

Since this is a self-signed certificate however, your browser will still complain it cannot trust it. So you need to move it to the Root store. Use the ***certmgr*** tool, find the certificate in the "Personal" Certificates folder (aka "My"), and drag it to the "Trusted Root Certification Authorities" Certificates folder (aka "Root").



Now configure your MO71 HTTPS preferences, just as the earlier screen-shot showed, specifically:-

- Select the "Current user certificate store" radio button
- Type in "Root" in the "Store Name"
- Type in "MO71" in the "Search String"
- Now press the "Start" button to get the HTTPS listener running. Check for any messages along the bottom of the preferences dialog. If it worked it will say "Listener Started". If it failed, it will provide some guidance about what failed.
- Now navigate your browser to https://localhost/mq/admin

This is just a simple way of trying it out on your own machine, using localhost as your URL. It is expected that you will have browsers from other machines connecting into this web server and so a certificate with a more appropriate ***Subject Alternative Name*** will be required, and one that is trusted by other machines rather than a self-signed one.

20.5.2. Advanced Configuration

Which TLS protocols are available (and enabled or disabled by default) varies by Windows version. A summary of these can be viewed here: https://docs.microsoft.com/en-us/windows/win32/secauthn/protocols-in-tls-ssl--schannel-ssp-

If you need to understand which protocols are enabled on your machine, you must view the settings in the registry in the path `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols`

By default there will be no subkeys defined, and so the above link will show what is enabled. You can explicitly enable or disable the various protocols by adding subkeys as shown here: https://docs.microsoft.com/en-us/windows-server/security/tls/tls-registry-settings

Which Cipher Suites are used, and in which priority order, varies by Windows version. A summary of these can be viewed here: https://docs.microsoft.com/en-us/windows/win32/secauthn/cipher-suites-in-schannel

If you need to change the priority order of cipher suites, please follow the Microsoft document which describes how here: https://docs.microsoft.com/en-us/windows-server/security/tls/manage-tls

20.6. Limiting what is displayed

MO71 has the concept of Usage Tailoring which allows you to configure which commands are available to the user. For example, you could decide that users can display Queues but not Processes or Channels. The Usage Tailoring section describes how you can make this configuration either globally or on a per Queue Manager basis. If the available commands are restricted via Usage Tailoring then those restrictions will also apply to the Web Access. This will affect two aspects of the web interface:

- Users will not be presented with unavailable links when using qmlist[]
- A 404 error will be generated if the user explicitly attempts to use a link which is prohibited.

20.7. Who is connected ?

By selecting *'HTTP Connections...'* from the *'Action'* menu MO71 will display a dialog of the connections from browsers. There are three types of connection displayed.

Type	Meaning
Running	The connection is connected and active
Blocked	The connection is connected but all activity is being blocked
Rejected	The connection was blocked but is now no longer connected. MO71 will maintain a list of the last 10 connections which were rejected.

The dialog will show all the currently connected devices and, possibly, the last 10 devices which were blocked and are no longer running. Connections can be blocked for two reasons. Firstly because the limit of devices has been reached for the type of licence MO71 is running under. Emerald licences allow one concurrent device, Ruby and Sapphire licences allow 5 concurrent devices. Diamond licences, of course, do not limit the number of concurrent devices. The second reason why a connection may be blocked is administratively by their IP address which is discussed is the following section.

The number of HTTP connections may be more or fewer connections displayed than you have devices running. The reason for this is that, for performance, some browsers will make multiple connections to display a single page. If that page is static, then after some internal time-out, the browser may close the connection even though the page is still displayed. So, it is perfectly possibly for a single browser displaying a list of queues, say, to have 0, 1, 2 or even 3 connections open to MO71.

There are a number of other attributes displayed for each HTTP connection such as the time the connection was made; the time of the last request; how many bytes have been transmitted in either direction and the User Agent string that identifies the type of browse in use. If HTTPS is in use, then the algorithms chosen during the TLS handshake will also be show in this list. Perhaps the most important attribute is of course the IP address of the device.

20.7.1. Limiting connections

By default MO71 allows any device to connect and display the MQ objects. However, you may well prefer to limit access to only a known subset of devices. This can be done by adding the IP address of the device to an 'allow' list. Under the HTTP tab of the “Preferences Dialog“ you can specify a list of IP address to allow to connect. So, if the list is empty then all devices can connect. However, if the list contains any entries then only devices which match those address will be allowed to connect. Any non-matching device will be returned a message stating they are blocked.

The format of the allow list address follows the normal scheme for this type of thing. Essentially in each segment of the IP address you can specify an actual number, a range using the '-' (minus sign) or a single wildcard character '*'.\

For example:

Valid Addresses	Invalid Addresses	Comment
1.2.3.4	1.2.3.4.5	At most four segments should be specified
1.2.3.*	1.2.3.4*	Wildcard characters must be the only character in a segment
1.*		Equivalent to 1.*.*.*
1		Equivalent to 1.*.*.*
1.2-8.10.45	1.2-*.10.45	Wildcard characters must be the only character in a segment
1.2.5-8.10-45	1.10-300.10.45	An IP address segment value must between 0 and 255 inclusive.

The Preferences Dialog has 3 buttons next to the allow address. 'Add' which will add the entry to the list. 'Check' which will check whether the given IP address is allowed. And 'Delete' which will remove the selected entry from the list. Because of browsers caching connections any changes to the allow list may take a minute or two to become effective.

20.8. Running MO71 as a service

The Web Access feature is available when running MO71 in normal foreground mode. However, some users prefer to isolate the Web Access from normal administrator use. In this case the MO71 which is serving the HTML pages is not tied to a particular user instance on the Windows machine. As a consequence it makes sense to run MO71 as a service to effectively run it in background mode. If you're doing so with the HTTPS protocol, consider using a certificate store that is machine wide.

MO71 itself does not know anything about Windows services. The reason for this is that there are many different utilities around which implement the job of running a standard Windows EXE as a service. Many users will have their own preferred way of doing this so it is more important that MO71 allows you to use the service utility of your choice. Examples of such utilities are:

Utility	Description
SC.EXE	The standard Windows way of creating a service that comes with windows. For more Information see: https://support.microsoft.com/en-us/kb/251192
NSSM	The rather amusingly named 'Non-Sucking Service Manager' Free to download and use. This utility has a good reputation amongst users for being easy to use. For more information see: http://nssm.cc/
SRVSTART.EXE	Allows any program to be run as a service For more information see: http://rozanski.org.uk/software
Launcher Service	Allows any program to be run as a service For more information see: http://emutastic.emulation64.com/netosoft/projects.htm
SRVANY	Supplied by Microsoft with their Windows Resource Kit For more information see:https://support.microsoft.com/en-us/kb/137890

Of course there are others and they all do essentially the same job. Here at MQGem we do not endorse any of theses utilities, but we have to say that we were impressed by 'NSSM' both in terms of it's ease-of-use and also its capabilities. However, which utility you use will, of course, be down to your personal preference.

20.8.1. Running MO71 in background mode

If you do choose to run MO71 as a service you should be aware that you are essentially running MO71 in the background. This is useful since the program is not tied to a single user (and therefore not to a single user terminal) but it does mean that the user cannot respond to prompts or messages. By default MO71 would display errors in a message box with an 'OK' button for the user to click on when they have read the message. However, when running as a service this is no-longer feasible. To get around this problem you should pass in the **-b** parameter to the mqmonntp.exe program when you run it as a service. The **-b** parameter tells MO71 to run in background mode. This has two effects:

- **Error messages written to a log file**
  Instead of displaying an error message box MO71 will write any errors to the file 'MQMON_BKGRND.LOG' in the M071 configuration file directory. It is worth periodically checking this file to ensure that MO71 is not reporting any errors.
- **Disconnecting from MQ**
  When the service is ended MO71 will attempt to disconnect any connections to MQ for up to 5 minutes. Of course it depends what service utility you use as to exactly how the service ends. Most utilities will try a number of different methods to end the program. For example, sending a WM_QUIT message to the application or terminating the process. MO71 will shut-down when it receives the WM_QUIT message. If you have the ability to configure the service utility to add a delay between the different stages then you should try to ensure that there is sufficient delay between the WM_QUIT and terminating the process to allow MQ to actually perform the disconnect. If you don't have sufficient delay then you may see error messages on your queue managers reporting that the client connection experienced a communications error. This is because the client program was terminated before it had a chance to issue the MQDISC().

Chapter 21. Predefined Dialogs

Accessible from the main menu under the '*Action'* menu the user can create predefined dialog definitions. These definitions allow the user to create a set of dialogs of various types which can be started either when the application itself starts or from the predefined dialog list dialog.

There are many advantages to predefined dialogs. It allows the user to pre-select most of the attributes, such as position, filter, and object name, of the object lists dialogs. This pre-canned definition can then be stored along with a description ready to be started at a moments notice. The simplest way to start a predefined dialog is to display the list of predefined dialogs and double click on the dialog required[35].

21.1. Definition

The attributes which can be associated with a predefined dialog are:-

- **Type**
  Each predefined dialog definition has a type. These can be :-
  - **List**
    A list is the definition of a list dialog. Starting one of these will start a list dialog.
  - **Group**
    A group is the parent object of a other predefined dialog definitions. A group can 'contain' any number of members. A predefined dialog can be a member of a group just by specifying the same group name. A group can contain any type of predefined dialog. Starting a predefined dialog Group will start all the dialogs which are members of that group. This means that a single click on a predefined dialog can start as many dialogs as you like.
  - **Graph**
    A predefined dialog graph is the definition of a graph format including how to obtain the required graph data. The definition is created by saving a graph. For more information please refer to Chapter 14 Graphing on page 114.
- **Groups**
  This field is optional, it is not required if you don't want to use groups. However, if you do wish to use groups then you can provide a comma separated list of group names. For example, "sales, queues, QM1". Each group name can contain blanks but cannot start or end with blanks. Starting a Group will start all dialogs containing any of the group names.
- **Dialog Type**
  Probably the most important attribute which is the type of dialog you want created. You can choose from almost all the **MO71** list dialogs.
- **Description**
  This field is a free format description of the dialog. This field can be used in conjunction with the **-p** parameter to start a defined subset of dialogs on application start. By including a word in the description, say MONITORMVS, the application can be started with the command, **mqmonntp –p MONITORMVS**, and all the dialogs with MONITORMVS in their description. The description can also be used as the window title, or as part of the window title, using the next attribute.
- **Title**
  Chose how the title of the window is created. The values can be:-
  - **Default**
    This value acts just like "Dialog" but may change in the future.
  - **Dialog**
    The window title will be formed from the queue manager name and the standard dialog title.
  - **Description**
    The window title will be formed from the queue manager name and the predefined dialog description field.
  - **Description Suffix**
    The window title will be formed from the queue manager name, the standard dialog title and then the predefined dialog description field.

[35]The action of double click on the predefined dialog list is settable by a preference option. The user can choose between getting the dialog definition or switching to the dialog

- **Queue Manager**
  The Queue Manager for which the dialog should be started. A special value of **<Prompt>** may be specified. If **<Prompt>** is used the application will prompt for the name of the Queue Manager every time a predefined dialog of this type is to started unless it is clear which Queue Manager should be used from the context.
- **Status**
  Shows whether a dialog of this type is currently active or not.
- **Id**
  A unique numerical id which identifies this dialog definition. The id of a definition is not guaranteed to be constant across multiple invocations of the program.
- **Filter**
  Use this field to set an initial value for the filter in the dialog.
- **Hide System Objects**
  This field defines whether the dialog should initially show the system objects or whether they should be hidden.
- **Auto Start**
  This field defines whether the dialog should be started automatically upon program start.
- **Initial Refresh**
  This field defines whether the dialog should refresh its display automatically on start. Specifying initial refresh **'yes'** displays the dialog already containing information but does not allow any selection fields to be specified.
- **Context Command**
  This field specified whether this dialog should be made available on the context menu. The context menu is the menu which is shown when you right-click on a Queue Manager or is available from the 'Commands' menu on the main window. **MO71** will use the 'Queue Manager' field to determine which Queue Managers this dialog is applicable to. The values and meaning are as follows:-

Value	Meaning
No	This predefined dialog should not be shown in the context menu. The dialog can still be started by using 'Start' in the predefined dialog list.
All	This predefined dialog should be shown in all context menus for matching Queue Managers.
Main	This predefined dialog should only be shown when the context menu is selected from a Queue Manager icon. For example, in the main window.
Qmgr	As 'Main' above but also the Queue Manager and Queue Manager List dialogs
Object	This option has two effects. First the menu is only shown if the currently displayed dialog is relevant to this predefined dialog definition. Secondly the object name that is currently selected is used as the key value for the new dialog. For example, you could define a predefined dialog type of 'Browse' with a Context Command of 'Object'. This would mean that when you were looking at a queue in some way then the context menu would contain this dialog definition. Selecting it would bring up this new dialog for the object currently selected. In this case it would allow you to browse the selected queue.
Prefix	This option is similar to the 'Object' definition above. However, rather than using the whole name as the key just the first portion of the name (up until the first punctuation character) is used. Suppose we were looking at a queue called **MQGEM.COMMAND.QUEUE**. By selecting this queue and then a predefined dialog definition defined with a context command of Object we would display this new dialog with a key of **MQGEM.***

- **Context Level**
  If the dialog is to be shown on a context menu, choose where on the context menu it should be positioned. The values can be:
  - **Normal**
    The normal position for predefined dialogs to be shown is in a submenu titled “Predefined Dialogs” off the command menu.
  - **Command**
    Choose this option to position this specific predefined dialog to be shown at the end of the Command menu.
  - **Top**
    Choose this option, to position this specific predefined dialog, to be shown on the top level context menu, rather than in a submenu. This is handy for your most commonly used dialogs. If the command menu is the top-most context menu, then this will have the same behaviour as “Command”.

- **Context Applicable**
  This field is available in Group definitions. It controls whether the context command should be shown only if all of the group members apply to the given Queue Manager or whether it should be shown if any of them apply.
- **Refresh Rate**
  If the dialog should be refreshed at regular intervals this field allows you to specify the refresh rate.
  You can enter the value in seconds or you can use a format such as [**n days**] [**n hours**] [**n minutes**][**n seconds**].
  The field will always be display in this format. Since this field is a character field it will not sort numerically in a list display. If you wish to sort the predefined dialogs in order of refresh frequency then you can use the field 'Refresh Rate (Seconds)'. Either add this field to the displayed fields and click on the top of the column or add a filter such as '**sortd(ratesec)**'
- **Refresh Align Time**
  This field allows you to specify a time during the day at which you would like a refresh. The value is only valid if the dialog has a non-zero value for Refresh Rate.

  The format of the align time is **HH**[:**MM**[:**SS**]]

  So values of 10, 10:15 or 10:20:30 are all valid.

  The time is used as a reference point for all other refreshes. So, for example, if you were refreshing hourly then specifying a value of 9:30 would request that all refreshes are done at half-past the hour. In this case the hour itself would be irrelevant and an align time of 0:30 would have the same effect.

  An align time of 'None' is shown is no align time has been specified. In this case the next refresh time is calculated purely from the time the dialog started and the Refresh Rate.

  An align time can only specified if the Refresh Rate of the dialog is equally divisible into 24 hours. If the Refresh Rate is not divisible, say 5 hours, then the align time is shown as 'Unavailable'.
- **Show Buttons, Show Fields, Show Filter**
  The settings that should be used to determine which parts of the dialog should be displayed.
  A preference option controls whether changing these settings while the dialog is running should be reflected back to the dialog definition.
- **Auto Export**
  The 'Auto Export' fields allow you to start a predefined dialog which will automatically write to an export file whenever it refreshes its data. The values specified will be preset in 'Refresh' dialog, for more information please see “Auto Refresh Export Fields” on page 21.
- **Export Type**
  Controls the format of the data which is exported.
- **Export File**
  The name of the export file. The export file format can contain character inserts which are described in chapter 24.8 File Name Inserts on page 228.
- **Export Append, TimeStamp, Always Header, Default Objects, System Objects, Default Differences,Blank Fields**
  The values control what is exported.. For a description of the fields see “Auto Refresh Export Fields” on page 21.
- **Restore**
  This field determines how the dialog should be displayed:-
  - **Normal** Normal, sizeable window
  - **Minimize** Minimized to the task-bar or window menu
  - **Maximize** Maximized
- **X Position, Y Position, Width, Height**
  The initial position and size of the dialog.
  A preference option controls whether re-sizing and moving the predefined dialog should update the values in the dialog definition. Alternatively you can start the predefined dialog, size and position it to your liking and then select 'Update Predefined Definition' from the context menu to set these values.

21.2. Menu

- **Create**
  The create menu item will take whatever fields are presented on the dialog and create a new predefined dialog entry. The predefined dialog list dialog will automatically update to show the new entry.
- **Update**
  The update menu item will take whatever fields are presented on the dialog and update the dialog entry with the id given by the id field.
- **Open**
  The open menu item will bring up the predefined dialog to allow the values to inspected or changed. This can be the double-click default action depending on a preference option.
- **Start**
  The start menu item will start a new instance of the selected dialog regardless of whether an instance of this dialog is already running.
- **Switch**
  The switch menu item will check whether there is an instance of the selected dialog already running. If there is the dialog is brought to the foreground, if there isn't a new instance of the selected dialog is created. This can be the double-click default action depending on a preference option.
- **Delete**
  The delete menu item will delete selected dialog definition.

21.3. Refresh Group

A predefined dialog definition can be a member of zero or more groups. A group is merely a comma separated list of names specified in the 'Groups' field of the definition. The main advantage of Groups is that starting a predefined dialog group will start all of the associated dialogs. Consequently predefined dialogs can be a very useful way of starting a whole sequence of dialogs that may be concerned with a particular situation, for example the dialogs associated with a particular application group or test-case.

The other advantage of Groups is the ability to refresh the entire group with one click. The refresh group context menu is available in any dialog that is associated with one or more groups. So, the predefined dialog group, a predefined dialog member and a command dialog itself can all refresh the group. In fact all the groups associated will be refreshed. It is possible, therefore, to effectively define a group or groups and refresh all those dialogs in a single click.

Selecting 'Refresh Group' from the context menu will issue the refreshes and report in the status field how many dialogs were refreshed.

More discussion on the concept of Predefined Dialogs can be found on our blog here.

Chapter 22. User Commands

User command are a simple way of augmenting the user interface with additional commands which may be useful to an MQ Administrator. The basic principle is that you can define a command to be issued and a menu value and that menu item will then be available in the context menu associated with your Queue Manager. The type of commands you might want to issue can be anything that can be run as a simple EXE, CMD or BAT file. So, for example, some suggestions might be:

Menu	Command	Description
Start Queue Manager	`strmqm %q`	Starts your Queue Manager
End Queue Manager Quiesce	`endmqm %q`	Ends your Queue Manager
End Queue Manager Immediate	`endmqm -i %q`	Ends your Queue Manager Immediately
Telnet	`putty telnet://%c`	Start a telnet session to the location's Machine
MQEdit	`mqedit -m %q`	Start an MQEdit message editing session
Start MQ Trace	`strmqtrc -m %q -t detail -t all`	Start MQ trace
End MQ Trace	`endmqtrc -m %q`	End MQ Trace
Run report	`runreport -m %q`	Run some user reporting tool
MQSCX	`mqscx -m %q`	Bring up an MQSCX session for this Queue Manager
Backup	`backupqm -m %q`	Run Queue Manager backup script
Remote Desktop	`C:\Windows\system32\mstsc.exe /v:%c /f`	Start a Remote Desktop Session
QLOAD Summary	`cmd qload -m %q -i %o -ds`	Run the QLOAD summary display for this Queue

There really is no limit to number and type of commands that you might wish to add to this list. Of course you could just run the appropriate command from the command line so you could argue that this does not do anything you couldn't do before. And that is true. However, this can save time hunting for the right command, the right parameters, the object name and syntax. It can also be a useful memory jogger and it is very useful if you wish to give a new user a preconfigured set of commands that they should issue should certain situations arise.

22.1. User Command Fields

Configuration of user commands could not really be simpler. On the main window, under the 'Action' menu item, you will see 'User Command List'. Select this and a standard list dialog will be shown. If you press 'Refresh' more than likely the list will be empty. So, let's create our first user command. Right click on the list and press 'New...'. This brings up a simple dialog with a few empty fields. The meanings of these fields are:

- **Menu Text**
  This is the text which will be displayed to the user in the context menu.

  You can specify any characters up to a maximum length of 256 characters.

- **Object Type**
  This field can be used to limit which dialogs will show the command in their context menu. A value of 'All' is used where this command should be available in all context menus. This is generally the case when all the command needs to know is the name of the Queue Manager that is involved. For example, any of the first few command examples above. Or you can define this field with a particular object type. This means that the menu will only be presented for these object types. If a user does this then it is usually used in conjunction with a command which might take an object name or object type. By using the %o and %O inserts then MO71 will insert the currently selected object into the command.

  This therefore allows you to define a command which should be run against only a particular object type and have the object name automatically added. For example, look at the **QLOAD** example above. This command uses the %o insert to pluck out the currently selected object name. Clearly this command would only make sense run against a queue so you would define the

User Command object with an object type of 'Queue' or, more likely, 'Local Queue' since only local queues can contain messages. This would then mean that the user command is only shown in the context menu of appropriate queue dialogs.

- **Command**
  This field contains the command you want **MO71** to issue. This can be an EXE program or a CMD or BAT file. The command can contain a variety of 'Command Inserts' described below.

  You can specify any characters up to a maximum length of 1000 characters.

- **Description**
  This field is a short description of what the command does as a memory jogger. If the command is confirmed before being executed then this description will shown in the confirmation window.

  You can specify any characters up to a maximum length of 128 characters.

- **Hidden**
  This field controls whether the command should be shown or be hidden as a background task. If you are doing something like starting trace you probably want this to be hidden, in the background. However, if you are starting a GUI, such as MQEdit, then you don't want the window to be hidden.

- **Confirm**
  This field controls whether MO71 should ask for confirmation before issuing the command. For example the 'endmqm' commands above would probably be defined with confirm equal to 'yes' to avoid the command being accidentally issued. If the user holds the 'Shift' key while selecting a user command from the context window then a confirmation dialog will always be shown.

- **Keep Window**
  This field controls whether the command window that has been started to run the command should stay around once the command is finished. If you are doing something like running a command that prints output to the screen, you probably want this command window to stay around. If your command text is not running a “cmd” string then this field has no effect.

- **Context Level**
  If the dialog is to be shown on a context menu, choose where on the context menu it should be positioned. The values can be:

  - **Normal**
    The normal position is for user commands to be shown is in a submenu titled “User Commands” off the command menu.

  - **Command**
    Choose this option to position this specific user command to be shown at the end of the Command menu.

  - **Top**
    Choose this option, to position this specific user command, to be shown on the top level context menu, rather than in a submenu. This is handy for your most commonly used commands. If the command menu is the top-most context menu, then this will have the same behaviour as “Command”.

- **Queue Manager Name**
  The Queue Manager Name field is one of a number of selector fields which control whether this command should be applied to a particular location. If a value is specified then the Queue Manager Name will be checked against the value to see whether there is a match. For example a value of 'WIN*' will only apply to Queue Managers starting with the letters 'WIN'.

- **Queue Manager Group**
  The Queue Manager Group field is one of a number of selector fields which control whether this command should be applied to a particular location. If a value is specified then the location will be checked to see whether it has a matching group value. For example a value of '*TEST*' will only apply to Queue Managers in groups containing the letters 'TEST'.

- **Location**
  The Location field is one of a number of selector fields which control whether this command should be applied to a particular location. If a value is specified then the location will be checked to see whether it has a location value. For example a value of '*Local*' will only apply to Queue Managers with a location field containing the letters 'Local'.

- **Network Name**
  The Network Name field is one of a number of selector fields which control whether this command should be applied to a particular location. If a value is specified then the location will be checked to see whether it has a matching network name. For example a value of '*z/OS*' will only apply to Queue Managers in groups containing the letters 'z/OS'.

Fill in the fields according to your needs and then press the 'Create' button. You will see that your newly created command object is now shown in the User Command list window. Now, you can bring up a context menu of the appropriate type and you should see

your new menu item included. Command context menus are available in many places such as IBM MQ Object and List dialogs, on the main menu, in the Network view and on the Application view dialogs.

22.2. Command Inserts

There are a number of insert sequences which can be used in the command specification. The advantage of this is that a single command definition can therefore apply to all of your Queue Managers and yet the command issued is different. An insert is simply a special character following the '%' (percent) character. The inserts available are:

Insert	Meaning
%[var]	Environment variable. For example %[temp] will be converted to the current value of the 'temp' environment variable
%%	Insert '%' character
%c	Location CLNTCONN connection name (not including any port number)
%d	Day of the month e.g. 16
%D	Day of the month e.g. Mon, Tue, Wed
%g	Location Group Name
%H	Current time hours
%l	Location Name
%m	Current Month e.g. Jan, Feb, Mar
%M	Current time minutes
%o	The object name, such as a Channel Name, of the currently selected object
%O	The object type, such as “Channel”, of the currently selected object.
%q	Queue Manager Name
%S	Current time seconds
%t	Current time in HH.MM.SS format
%y	Four digit year

Examples of some of these inserts in use are in the opening section of this chapter.

More discussion on the concept of User Commands can be found on our blog here.

Chapter 23. Activity Trace

If correctly configured IBM MQ[36] will write Activity Trace messages, which describe the current application messaging activity to a special queue, normally SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. This can be extremely useful for a variety of reasons from logging activity to debugging application errors.

These messages are in a special format know as PCF format. Each MQI operation is described in a PCF Group. The format is relatively straight-forward to process but it is not very human readable. What MO71 does is make it very easy to browse these messages and see what is going on.

23.1. Configuration

At the top level the configuration of Activity Trace is done using an MQAT.INI file in the Queue Manager data directory. This file is read by the Queue Manager to determine which applications should be traced and how much information should be included in the output. Clearly it will depend on your installation about how much data you wish to capture.

However, to give you an idea here is an example.

```
#*******************************************************************#
#* Module Name: mqat.ini                                           *#
#* Type       : WebSphere MQ queue manager configuration file      *#
#  Function   : Define the configuration of application activity   *#
#*              trace for a single queue manager.                  *#
#*******************************************************************#
AllActivityTrace:           # Global settings stanza
  Trace=On                  # Control whether tracing is on or not
  ActivityInterval=10       # Time interval between trace messages
  ActivityCount=100         # Number of operations between trace msgs
  TraceLevel=HIGH           # Amount of data traced for each operation
  TraceMessageData=100      # Amount of message data traced
  StopOnGetTraceMsg=ON      # Stop trace on get of activity trace message
  ApplName=*                # Application name (may be wildcarded)

ApplicationTrace:           # Application specific settings stanza
  ApplClass=ALL             # Application type
  ApplName=*                # Application name (may be wildcarded)
  ApplFunction=*            # Application function (may be wildcarded)
  Trace=ON                  # Activity trace switch for application
  ActivityInterval=10       # Time interval between trace messages
  ActivityCount=100         # Number of operations between trace msgs
  TraceLevel=HIGH           # Amount of data traced for each operation
  TraceMessageData=100      # Amount of message data traced
```

The basic structure of the INI file consists of one 'AllActivityTrace' stanza followed by as many 'ApplicationTrace' stanzas as required. The latter allows you specify the tracing characteristics for particular applications or groups of applications. The former stanza allows you to specify the tracing characteristics for everything else not mentioned explicitly.

In this file we are saying that we want to trace everything. This can be useful when you are initially testing the feature until you become familiar with it. At that point you can start reducing the amount you capture.

23.2. Switching Tracing on and off

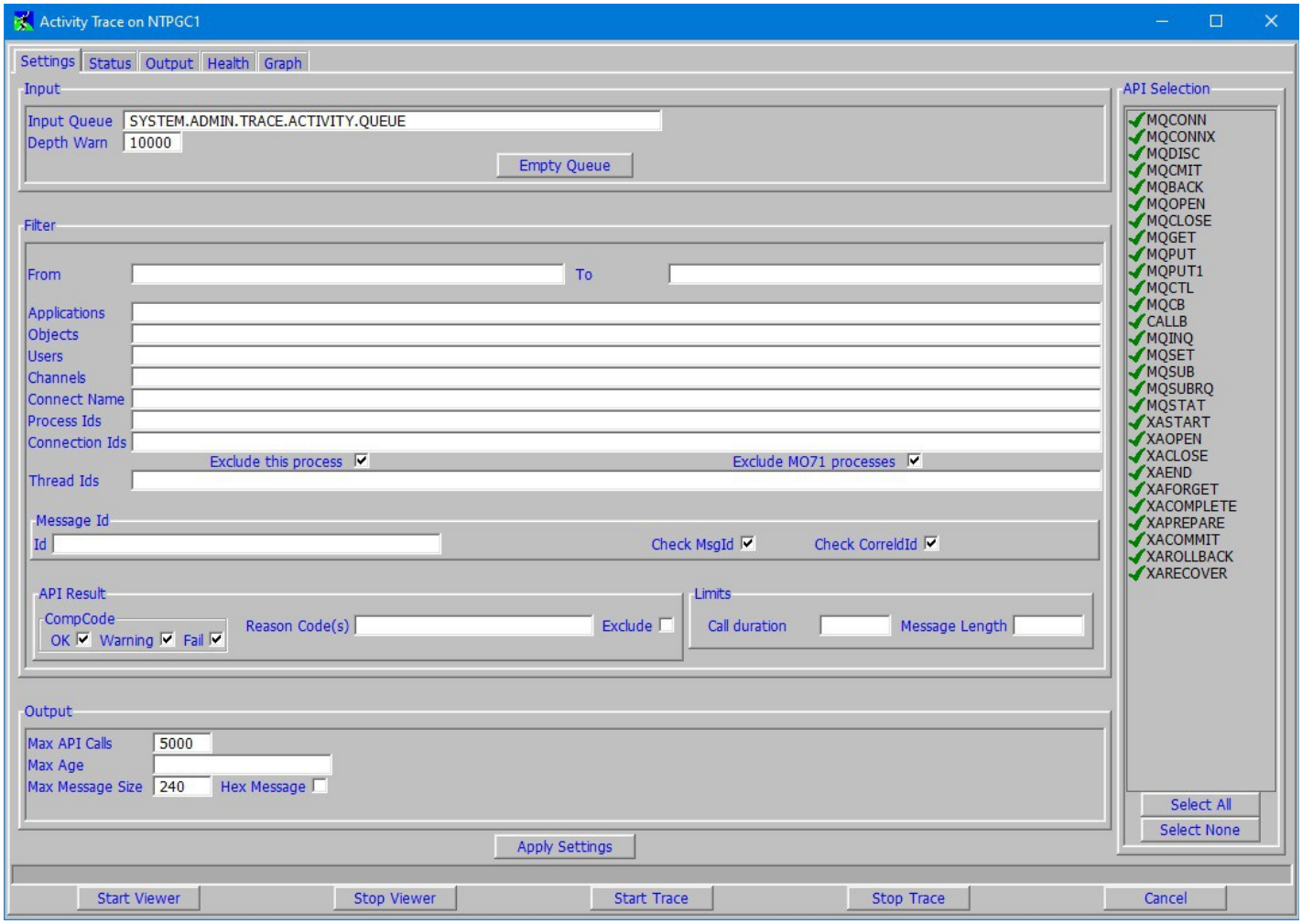
Just because you have a configured MQAT.INI file does not necessarily mean that the Queue Manager will capture the data. You have to have tracing switched on as well. This is done by changing the Queue Manager attribute, ACTVTRC. Clearly this can be changed using RUNMQSC, MQ Explorer, MQSCX or MO71 itself. To make things even easier the Activity Trace dialog also provides a simple short-cut method setting this value.

[36]Unfortunately this feature is not available on the z/OS product

23.3. Activity Trace Dialog

The IBM MQ product comes with a sample program, AMQSACT, capable of reading and displaying Activity Trace messages. However, the filtering and formatting options available with this program are very limited.

MO71 provides a window into the Activity Trace messages in a single dialog. This dialog is started by selecting the Queue Manager you are interested in and selecting *Action-Activity Trace...* from the main menu. This will present a dialog that looks like this:



At first glance this can appear a little daunting but don't worry it is very simple. The dialog has five tabs, these are:

- **Settings**
  This tab allows you to set the values on which the dialog should operate.
- **Status**
  This tab shows you the current status of the viewer and various API counts
- **Output**
  This tab is where you will see the output of the viewer and be able to interrogate the APIs that have been issued.
- **Health**
  This tab will allow you to check the health of some of the APIs
- **Graph**
  This tab will allow you to display the API usage in a graph of some sort.

The first step though is to set the setting values to the ones you need. You can see that the settings tab itself is split into four areas.

We'll discuss each of these in turn.

23.3.1. Input

The input area is concerned with reading the Activity Trace messages. There are the following fields:

- **Input Queue**
  This is the name of the queue that you want MO71 to read for the Activity Trace messages. By default MO71 will expect this queue to be SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE which is the default location for these messages. This means that you can have MO71 read the messages as they are generated. However, you could change this queue to be another location and read messages that have been previously collected or offloaded.

  MO71 will be browse the messages, they will not be destroyed.

  It is possible, and sometimes very useful, to have multiple dialogs pointing at the same queue. This allows you to have multiple views against the same data. If you do this then the messages themselves are only read from the queue once, the various dialogs share the same consumer.

- **Depth Warn**
  Clearly MO71 will need to read the entire contents of the queue. If the queue is very deep then this could take some time, especially over a client connection. To ensure that MO71 doesn't process large numbers of messages unnecessarily this field allows you set the point at which MO71 will ask for confirmation that you really do want to process these messages.

- **Empty Queue**
  Clicking this button will attempt to empty the given queue. Note that this is destructive, any messages on the queue will be lost. As such a confirmation dialog is shown asking whether you really do want to empty the queue. If you say you do then MO71 will first attempt to use the CLEAR QUEUE command and if that fails, because the queue is 'in use', then MO71 will repeatedly get the messages off the queue using MQGET.

23.3.2. Filter

The filter section of the dialog is concerned with filtering the input messages so that only the messages you are interested in are displayed in the output area. The messages can be filtered by a variety of contents. The filtering occurs locally, and does not require re-reading the trace queue and is therefore efficient. You can refine the filter fields again and again, each time pressing the 'Apply Settings' button without fear of causing huge amounts of messaging activity.
The filter fields are as follows:

- **From/To**
  These fields allow you to filter the messages based on when the API operation was issued. The full format of the date is:

  DD/MM/YYY hh:mm:ss for example 25/12/2016 09:12:34

  However, when you specify a date you need only specify as much of the time and date as you need. Some examples would be:

From	To	Meaning
9	10	From 9am to 10am on the current day
9		All APIs issued after 9am today
	23	All APIs issued before 11pm today
9:45	9:50	All APIs issued between 9:45 and 9:50 today
9:45:22	9:45:22	APIs just issued at 9:45 and 22 seconds today. Note that the times are inclusive
3/4 8		All APIs issued after 3rd April this year at 8am
9	now	All APIs issued from 9am until 'now'
now		All APIs issued from the time at which you press the 'Apply Settings' button

  When you press the 'Apply Settings' buttons the fields will change to their full resolved values so you will be able to see exactly what dates are in effect.

- **Applications**
  In this field you can specify a comma separated list of applications you are interested in. For a match, the strings you specify must just appear somewhere in the Application Tag field. It is not necessary to type in the whole Application Tag field.

- **Users**
  This field allows you to specify a list of userids to filter by. Users must be specified in full, in the correct case.
- **Channels**
  This field allows you to specify a list of channels to filter by. Channels must be specified in full, in the correct case.
- **Objects**
  This field allows you to filter based on the object name that the API is acting on. You can specify a list of wild-carded names. You can also start the name with an '!' to indicate that you want to invert the filter. All values are case-sensitive. So, for example the following could be used.

Filter	Meaning
Q1	Just show APIs for Q1
Q1,Q2,Q3	Show APIs for Q1, Q2 or Q3
Q*	Show APIs for any queue starting with a Q
!SYS*	Show all queues except system queues
SYS*,!*ADMIN*	Show system queues but no queues containing the string ADMIN.

- **Connect Name**
  This field allows you to specify a list of connection names to filter by. The connection name must be specified just as it appears in the activity trace message, usually the full IP address.
- **Process Ids**
  This field allows you to specify a list of process ids to filter by. Up to 10 process ids can be given.
- **Connection Ids**
  This field allows you to filter by Connection Id. Up to 10 Connection Ids can be specified in a comma separated list.
- **Exclude this process**
  By default MO71 will attempt to not have it's MQI activity traced since this can lead to loop conditions. This is done by using the MQCNO_ACTIVITY_TRACE_DISABLED connect option. However, this option can be disabled depending on the value of the ACTVCONO Queue Manager attribute.

  So, it is possible that MQI Operations issues by MO71 will generate Activity Trace records. Most of the time you would not be interested in these records so this option allows you to filter them out.
- **Exclude MO71 processes**
  Similar to the option above but this option will filter out all records generated by MO71, not just the current process.
- **Thread Ids**
  This field allows you to specify a list of thread ids to filter by. Up to 10 thread ids can be given.
- **Message Id**
  This option allows you to filter by the Message Id. Only API calls using the particular Message Id will be shown. You can choose whether MO71 should look for the Message Id in the Message Id and/or Correlation Id field. The Message Id field itself should be specified as a 48 character hex string.
- **API Result**
  These options allow you to filter based on the result of the API call. For example, you could ask to see only the failed API calls or an API call that returned a particular reason code or set of reason codes. If multiple reasons codes are of interest then just specify a comma separated list.
- **Exclude**
  This checkbox allows you to choose whether the reason code(s) you have specified are excluded from the output or are the only ones shown.
- **Call duration**
  If you are trying to track down a performance problem it may be useful to filter APIs that exceed a particular duration.
- **Message Length**
  Again if you are trying to track down a performance problem then it can be useful to look for API calls which are sending/receiving messages beyond a particular threshold.

23.3.3. Output

The output area is concern with what is shown to the user in the Output tab.

- **Max API Calls**
  This option allows you to specify a limit to the number of API calls which are shown in the output tab.. The maximum value which can be specified is 30,000. Note that you can still graph more than this number of APIs. However, it is still recommended, for the sake of performance, that you don't try and graph enormous numbers of API calls. Use the filter fields to reduce the API's to those you are interested in.
- **Max Age**
  This option allows you to specify the maximum age of API calls shown. The input is fairly free format. The default units are seconds but you can qualify the units if you wish. Examples could be:
  - 5 min
  - 1 hour 30 min
  - 2h5m30s

  Once the 'Apply Settings' button has been pressed the full parsed time value will be shown.

  The maximum age option can be useful for displaying a dynamic view of what is happening at the moment. For example, settings a value to, say, 20 seconds, will mean that only the very recent APIs are displayed. This can be useful for demo purposes to demonstrate the APIs a particular application issues at any particular moment.
- **Max Message Size**
  In the MQAT.ini file you can specify the maximum amount of message data which is included in the Activity Trace message. This setting field follows on from that and sets a maximum amount of message data which is displayed.
- **Hex Message**
  This checkbox specifies whether message data should be displayed in just character format or also include the hex representation.

23.3.4. API Selection

By default MO71 will show all API calls. However, it can often be useful to limit the display depending on what you are trying to diagnose. For example, to track down a security problem you may only wish to see the MQCONN(X) call.

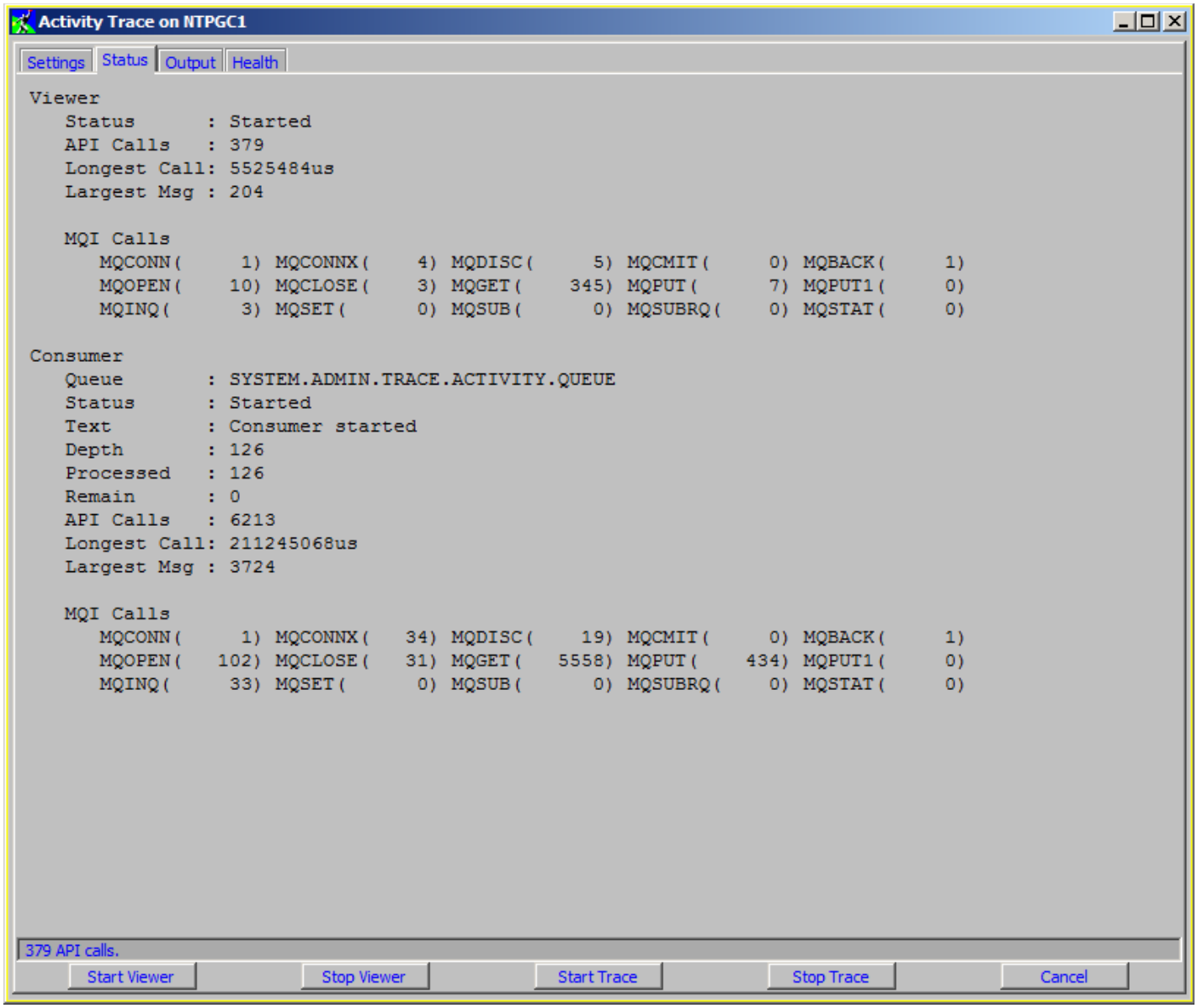
23.3.5. Actions

The dialog has a number of actions which control its operation.

- **Apply Settings**
  This button is the button you press when you have made the changes you require in the Settings tab. In fact, the Dialog won't do anything until you press this button! You can make changes to the settings at any time and pressing this button will apply the changes and the output will change accordingly.
- **Start Viewer**
  By default the viewer is 'started' which means that any new Activity Trace messages will be shown immediately. However, you can stop the viewer in which case this button is used to start it again.
- **Stop Viewer**
  Stopping the viewer can be useful if you want to prevent the data from changing underneath you.
- **Start Trace**
  Activity Trace messages are only collected if Activity Trace is actually in 'started' state. This is controlled by the value of Queue Manager attribute ACTVTRC. This button provides a quick way of issuing the ***ALTER QMGR ACTVTRC(ON)*** command.
- **Stop Trace**
  As above this button provides a quick way of issuing the ***ALTER QMGR ACTVTRC(OFF)*** command.

23.3.6. Status Tab

The status tab gives the current status of the viewer, including counts of the various APIs.



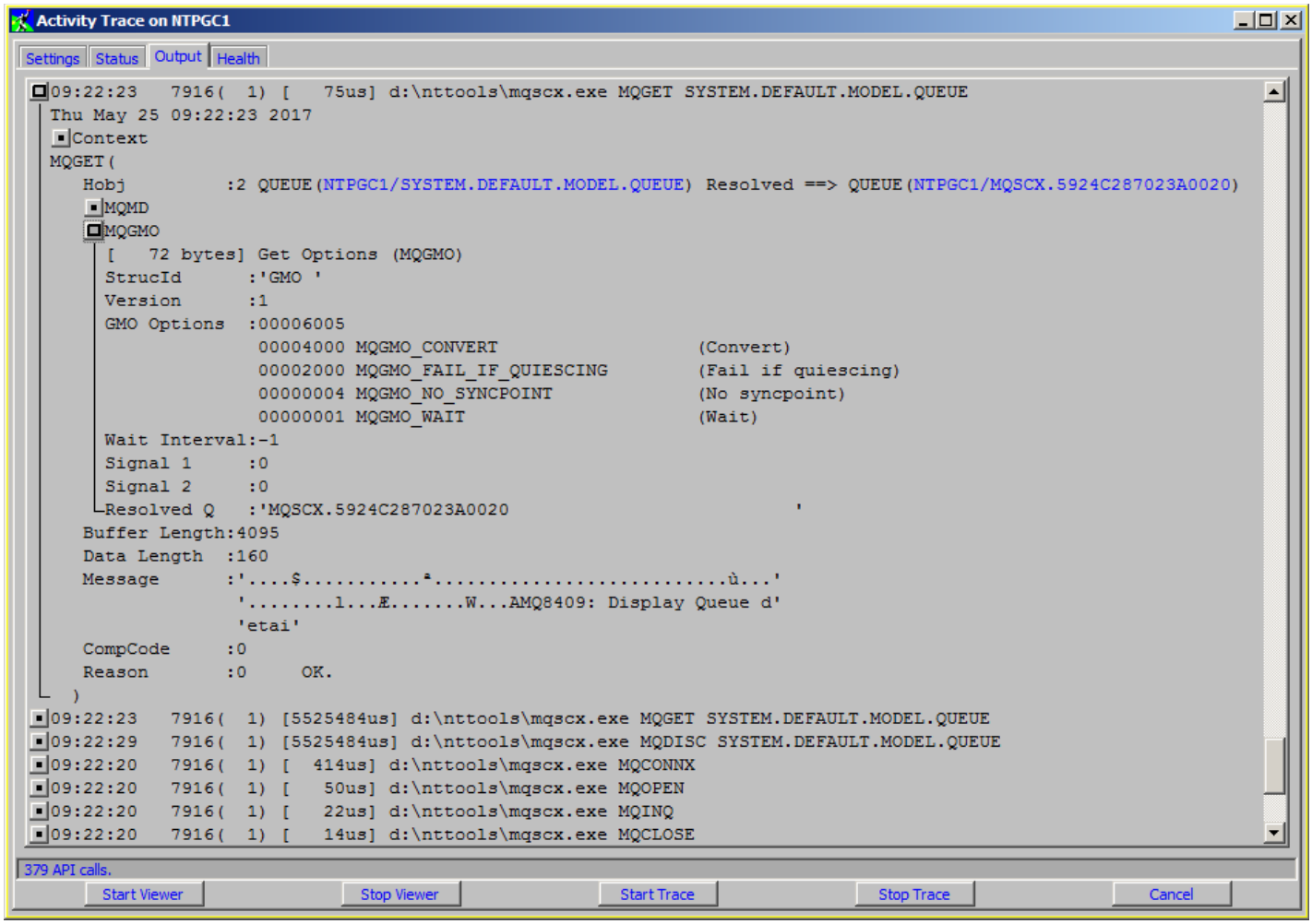
The status is split into two sections. Firstly the statistics for the viewer itself. These are the records which match the filter values on the settings tab. Secondly the consumer statistics. This represents all the messages that have been read from the queue. One way to think of it is that the viewer values are particular to this dialog only, whereas the Consumer values are global and any Activity Trace Dialog reading the same queue would have the same values.

If the dialog has a lot of Activity Trace messages to read then a progress bar will also be displayed giving an indication of when analysis will be complete.

If the dialog is not currently associated with a queue then the status window will read 'No attached consumer'. To start analysis set the setting tab fields to the values required and press the 'Apply Settings' button.

23.3.7. Output Tab

The output tab is where MO71 will display the formatted Activity Trace messages. It will display an output like the following:



Each API call is represented by a top level line. The API calls are shown in chronological order, the oldest being at the top. The top level line shows the following fields.

Time Process Id(Thread Id) API Duration Application Name API Call Object Name(if any)

If you wish to see the API is greater detail then pressing the icon on the left-hand side will expand the API[37]. Expanding the API for the first time will cause MO71 to re-access the queue to read the Trace Message in greater detail. If, for some reason, the message is no longer on the queue then MO71 will display a message to this effect.

The exact format of the expanded lines will depend on the API call in question but they all follow a similar format. The first line will be the API context. This can be expanded to show general information about the API call such as the Queue Manager it was issued on, the user id, the host name it was issued on and any channel or connection name information.

The context line is followed by the API itself. Exactly what it shown in the API lines depends on which API call it is and the level of Activity Trace detail. Bear in mind that even at high detail not every field of every API call is available. The viewer can, of course, only show the data which is included in the Activity Trace message itself.

On some APIs, those that refer to MQ objects, the object names themselves are displayed as links. This is demonstrated in the screenshot above. Pressing these links will display a standard MO71 dialog displaying the object definition.

Multiple views of the same consumer data can be shown just by starting multiple dialogs and associating them with the same input queue.

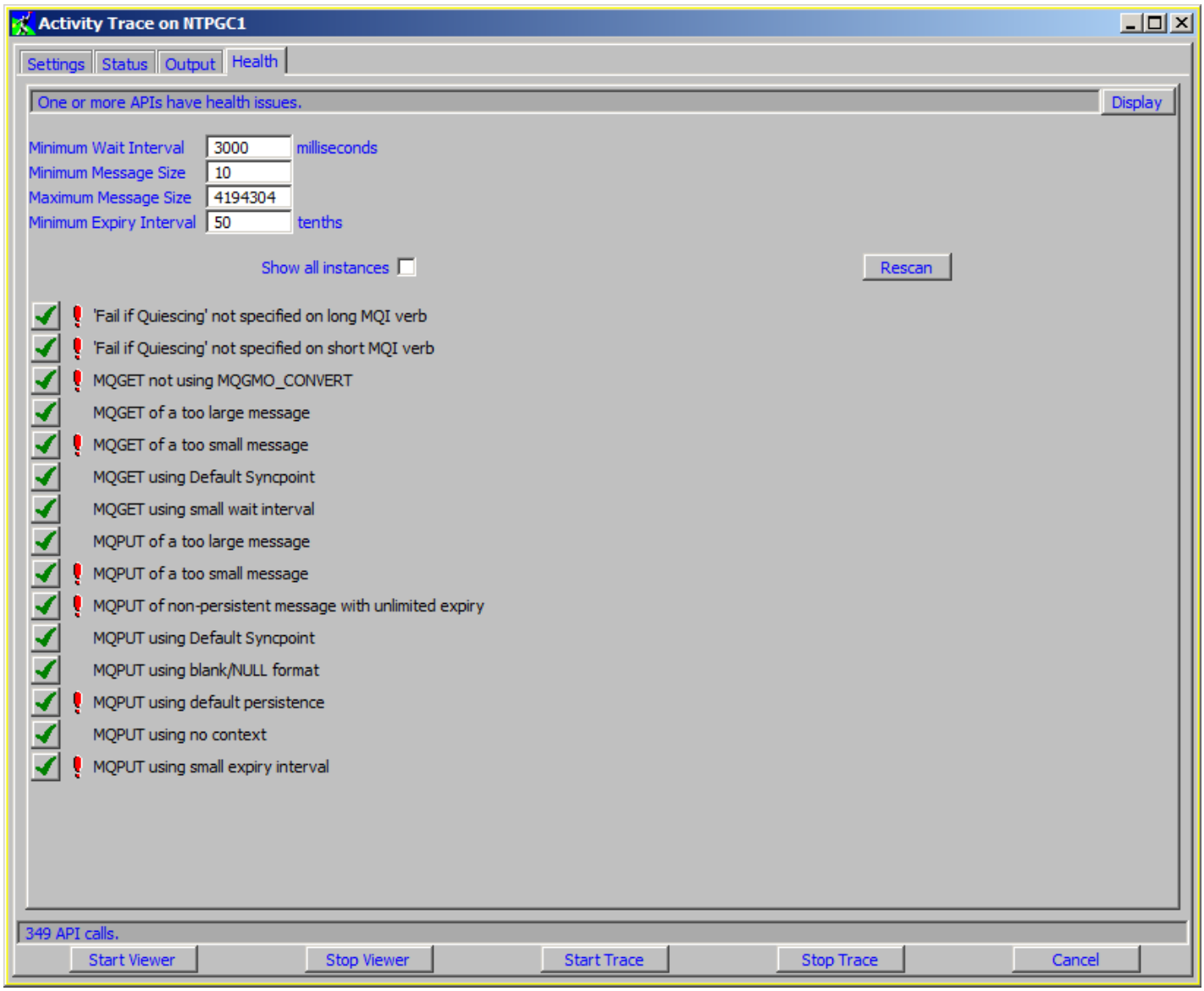
Text can be copied from the Output container, indeed any MO71 container, by swiping with the mouse and pressing Ctrl-C. This makes it easy to copy and paste fields into the settings tab filter fields to hone the display to just the messages you are interested in.

[37]For information about keyboard interaction with container windows see “Keyboard Control“ on page 226.

23.3.8. Health Tab

The Health Tab is concerned with whether the APIs being issued by the applications are reasonable. Now, clearly it is not possible for MO71 to verify that the application is correct. However, there are some MQI rules and guidelines that can be verified. For example, an MQI should always specify a SYNCPOINT option. If it doesn't then it's behaviour would be different if it ran against the Distributed and then z/OS MQ products. Another example might be message size. It is not a good idea to MQPUT very large messages to MQ nor is it a good idea to send lots of really small messages. Ideally you want to avoid the extremes. So, the Health Tab allows you to search for MQI calls which use these extreme message sizes. A final example might be 'Fail If Quiscing'. Fail if Quiescing is an option you can specify on all MQI verbs which tells MQ to fail the MQI call if the Queue Manager is trying to end. It is generally recommended to always specify this option.

The Tab itself looks like this.....



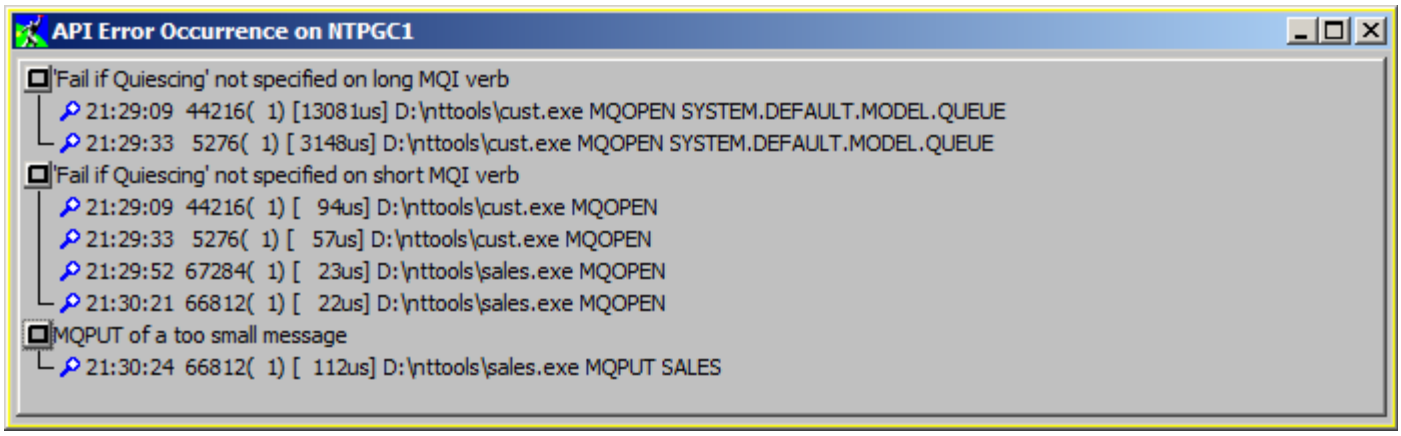
Essentially this dialog is a list of 'problems' that MO71 will scan for in the API records[38]. Some of the 'problems' need a comparison value and these values are settable at the top of the dialog. For example, we all know that putting large messages to an MQ queue is not a good idea. But exactly what is counted as large is debatable. Some would argue that anything over 1 Megabyte is large, while others would say it's closer to 4 Megabytes. Either way these fields allow you to set the limits.

[38]It is anticipated that this list will grow over time. If you have a suggestion for a check that MO71 could perform we would love to hear from you.

You can choose whether MO71 should perform the check or not by selecting the check mark next to the problem. For example, you may decide that for a certain application it is perfectly normal for it to put really small messages and that you are not interested in being told that that is happening.

Immediately following the check mark MO71 will output an exclamation mark ('!') if it detects one or more APIs which seem to have this problem. Note that the detection is based on the filtered API call, not on the entire list of API call read from the queue. So, for example, by filtering by a particular application name you can check whether any of these issues exist in just that application.

Of course knowing that there may be problems somewhere in your application is not that helpful unless we can look at the API call itself. That is where the 'Display' button comes in. If we press the 'Display' button we get a dialog such as the following:



This dialog lists the locations where each problem was found. By default only the first instance of each problem, per process and thread, is shown. After all, if an application issues an MQGET call without specifying a SYNCPOINT option then it is likely that each of the calls it makes will lack this option. However, if you really do want to see each occurrence of the problem then you can check the 'Show all instances' check box and then press the 'Rescan' button.

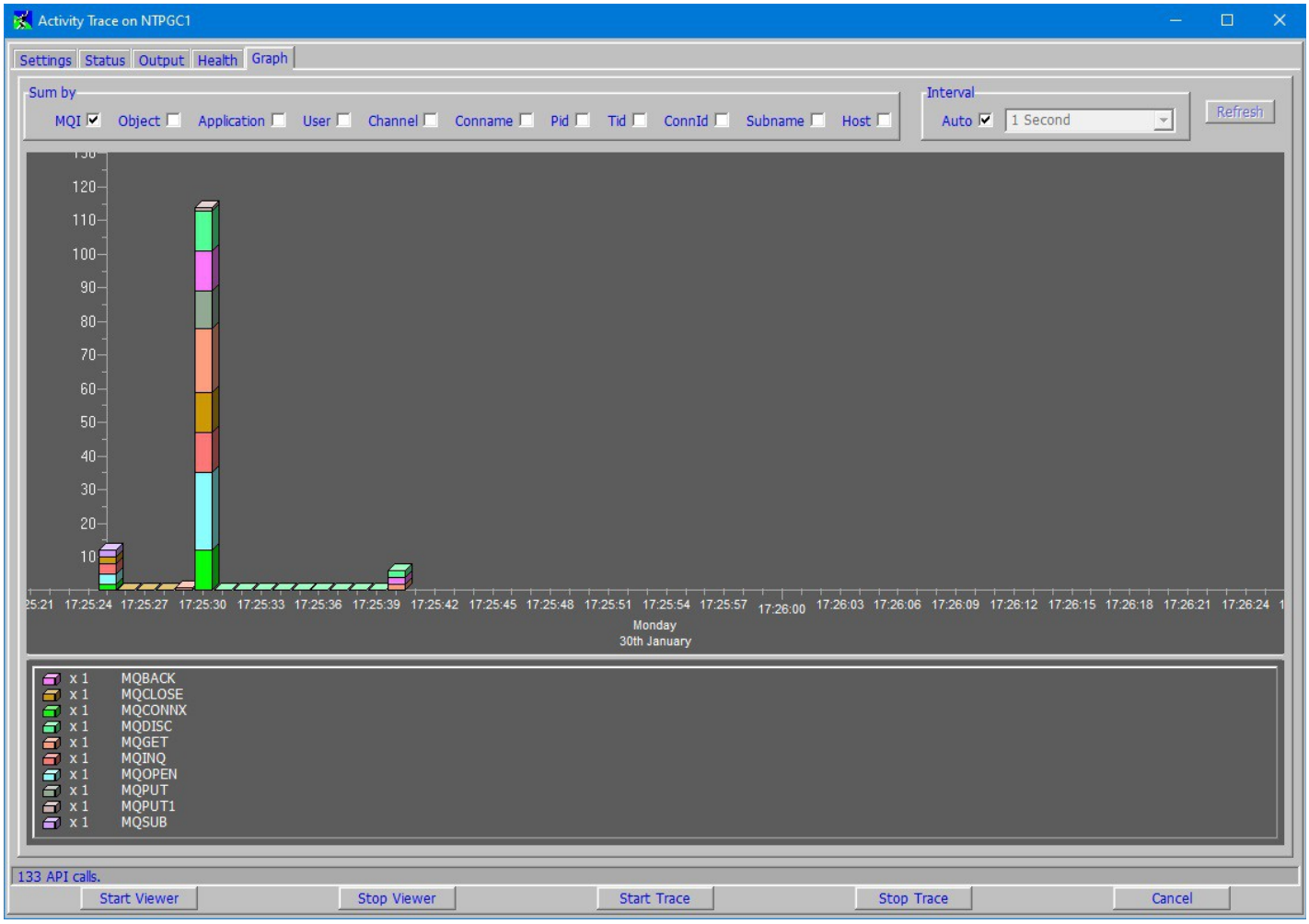
So, now we see the locations of the problem API calls however we still can't see the whole API call. Well, the answer to that is to press the Magnifying Glass symbol. Pressing that will immediately select the offending API call in the Output tab of the Activity

Trace Dialog. You can then expand the API call and see just what call was made. Of course, as has been said before, the amount of the API call displayed and the completeness of the API problem check will depend on the Activity Trace Detail level.

The API Error Occurrence display is 'sticky'. By that we mean that the records shown will persist longer than the Activity Trace records themselves. For example, suppose in the settings dialog you say that you only want to see the last 100 API calls. MO71 will still 'see' and process each API call as it is made. If a problem is detected, and the API Error Occurrence Dialog is running, then a problem record will be added to the dialog. Once the API call is no longer one of the last 100 API calls then it will be removed from the Output Tab. However, it will not be removed from the Occurrence Dialog. This means that it is entirely possible to have records in the Occurrence dialog which have no matching API call in the Output Tab. If you press the Magnifying Glass icon for these records you will be told that there is no matching API call. However, it is still useful to know that MO71 did see a problem. You can now change the settings to match the problem API call you are interested in and have MO71 display the set of records. Remember that when you change the filter settings MO71 does not need to go back to the queue but will do a scan of it's internal messages which tends to be very fast.

23.3.9. Graph Tab

The graph tab allows you to get a visual representation of the MQI calls that are contained in the Activity Trace data.



For some types of diagnosis having a graph of what happened can make things much easier to understand. Essentially an example of the saying “a picture is worth a thousand words”.

The graph Tab consists of three sections.

23.3.9.1. Selection Fields

The top part of the screen allows you to select how you would like the data grouped. For example, you could just be interested in the distribution of MQI calls as above. However, you might want to view the data split by the channel name it was issued under. Or perhaps both. You can check as many items as you wish and the combination is what will be graphed. Note that the more items you check the more lines your graph is likely to contain.

The second value to set is the interval. You can leave this to be automatically chosen, in which case MO71 will use the data frequency to determine a value. Or you can specify a longer time interval. For example, you might only be interested in seeing what happened minute by minute or hour by hour. Note that the longer time interval you choose the fewer data points will be on your graph.

When you have chosen the fields you want then you can press the 'Refresh' button which will show the graph. At any time you can change the selection fields and press 'Refresh' and the graph will update to show the new values.

23.3.9.2. Graph

The middle part of the screen is the graph itself. This is essentially just a 'normal' graph that is described in Chapter 14 Graphing on page 114. You can, for example, right-click on the dialog to display the Graph options dialog. From there you can change almost

every aspect of the graph, from the line types, the colours used, the description text etc you can even add new data lines, say from MQEV, if appropriate.

One slightly difference in behaviour from other graphs is that you can jump directly from the graph to the Activity Trace output tab by holding the <Alt> key and double clicking on the graph data point. The best way to find it is just to move your mouse over the graph until the tooltip for the data line is shown and then hold <Alt> and double click. Of course you are only taken to the output tab if the output tab current contains the appropriate record.

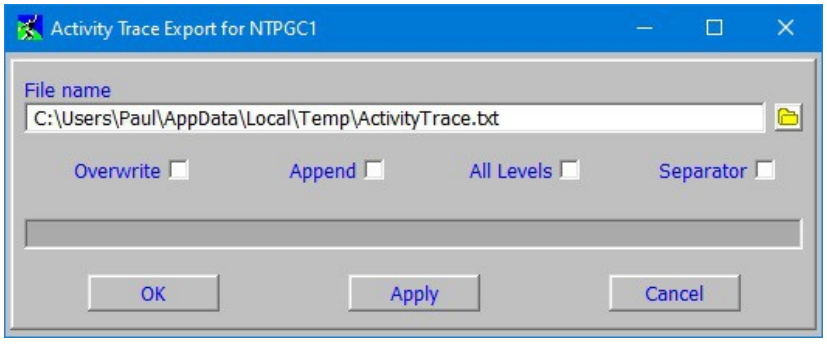
23.3.10. Context Menu

Right-clicking on any tab of the dialog will display a content menu which contains a number of the standard actions described above. It also contains the following:

- **Previous and Next Matching Record**
  These menus, and the PF7 and PF8 keys, allow you to traverse backwards and forwards through the output data looking at 'matching' output records. Initially there is no specific match criteria so every record will match. However, you can change those below.
- **Match Values**
  Match values allow you to limit which records match the prev/next traverse mentioned above to only certain records. For example, you might be interested in only looking through MQI calls from this connection. Or perhaps you want to see the next instance of this type of MQI call. Or the next call issued by this Pid or Tid. Or perhaps the next verb issued by this IP address. Or perhaps a combination of these. You can check or uncheck as many criteria as you wish.
- **Linked MQI call**
  A linked MQI call is an MQI call which is related to the current MQI call is some way given by the following table.

MQI Call	Linked MQI Call
MQCONN(X)	MQDISC
MQDISC	MQCONN(X)
MQOPEN	MQCLOSE
MQCLOSE	MQOPEN
MQGET, MQPUT, MQPUT1, MQCB	MQOPEN
CALLBACK	MQCB
Everything else	MQCONN(X)

- **Export Output**
  This menu will display an export dialog which looks like this:



This dialog allows you to easily copy the output tab contents to a file.

- **All Levels**
  By default only the data shown will be exported. This allows you to just open whichever twisties are of interest and have those exported. However, by selecting 'All Levels' you can request that MO71 exports the full contents of each entry[39].

[39]Depending on the number of API calls this option may take an amount of time since MO71 may have to retrieve and parse many data items. It is best to ensure that only those API calls of interested are selected in the main settings tab.

- **Separator**
  If selected then MO71 will output a line of '-' between each API call. This makes it easier to identify the boundaries between each call.

- **Set and Goto Bookmark**
  These menus allow you to explicitly save a particular record location and then jump back to it at any time.
- **Undo and Redo Position**
  As you navigate through the output records, jumping to linked MQI calls for example, MO71 will remember a history of where you have been and allow you to 'undo' your position move and go back to records you looked at previously.
- **Copy Output Selection to Clipboard**
  This menu will copy the text currently selected in the output tab to the clipboard. If no text is selected then this option will be greyed out and unavailable. This option is equivalent to pressing Ctrl-C on the output tab.
- **Copy Output Shown to Clipboard**
  This option will copy all lines shown on the output tab to the clipboard. This is logically equivalent to selecting all the text and then pressing Ctrl-C.

More discussion of the features of analysing Activity Trace within MO71 can be found on our blog here.

Chapter 24. Customization and Reference

24.1. Menu options

Many of the menu options act on the selected Queue Manager in the main window. The menu options available are therefore dependent on which Queue Manager is selected. Make sure that the correct Queue Manager is selected before issuing the command.

The menu options available are:-

24.1.1. File

- **Refresh Information**
  Causes the application to refresh its cached information about the selected Queue Manager. This should be used to update the information shown in the tree view of the main window.
- **Refresh Default Objects**
  Causes the application to refresh the definitions of the default objects for the selected Queue Manager. Note that an initial refresh of the default objects can be performed automatically by selecting the appropriate option in the location definition.
- **Open Location**
  Displays a dialog showing the values set for the location allowing the current configuration to be changed. See “Location Dialog” on page 214 for a description of the fields.
- **Copy Location**
  Displays a dialog to allow a new location to be added to the list. The dialog will be pre-filled with the values of the selected location. See “Location Dialog” on page 214 for a description of the fields.
- **Add Location**
  Displays a dialog to allow a new location to be added to the list. See “Location Dialog” on page 214 for a description of the fields.
- **Change Multiple Locations...**
  Displays a dialog allowing the user to change many location values in a single go. For more information please see Multiple Location Change Dialog on page 230.
- **Import Locations**
  Allows the user to have location definitions automatically created by importing the definition from an existing source.
  - **From CCDT...**
    This option allows locations to be imported from either a binary or JSON CCDT file. The choice depends on the name of the file chosen. A JSON CCDT file should end with the **.json** file type. Any other file type will be assumed to be a binary CCDT file.

    Any client definitions that are found are presented in the dialog when the “Read CCDT” button is pressed. An icon next to the Queue Manager name specifies the status of the definition.

Exclamation Mark (!)	Cannot be imported because the Queue Manager name in the CCDT is blank.
No entry sign	Cannot be imported because there is already a location definition for this Queue Manager.
Tick	Definition can be imported and is currently selected.
Cross	Definition can be imported but is not currently selected.

    If required you can set the queue manager group, the network names, and whether this location is MVS, using the fields provided below the list of queue managers.

    Pressing 'Import' will import those definitions which are shown with a 'Tick' against the name.

  - **Local Queue Managers...**
    Checks whether all of the local Queue Managers are defined in the MO71 configuration. If one or more are missing a dialog is displayed which allows the user to automatically import a location definition.

- **From MQ Explorer File...**
  Allows the user to specify the XML file created when an export is issued from MQ Explorer. Any client connection information definitions that are found are presented in the dialog. An icon next to the Queue Manager name specifies the status of the definition.

No entry sign	Cannot be imported because there is already a location definition for this Queue Manager.
Tick	Definition can be imported and is currently selected.
Cross	Definition can be imported but is not currently selected.

  You may also choose whether the MQ Explorer Set names should be used as the MO71 Group and Network Names. Pressing 'Import' will import those definitions which are shown with a 'Tick' against the name.

- **From MO71 Configuration...**
  Allows the user to specify another MO71 Configuration file to import locations from.

blank	Can be imported, there is no equivalent current definition.
Delta Symbol	Can be imported. There is a similar definition defined but it is different in some way.
No entry sign	No need to import this location because it is the same as a currently defined location.
Tick	Definition can be imported and is currently selected.

  Pressing 'Import' will import those definitions which are shown with a 'Tick' against the name.

  By default locations will use the same Queue Manager group and network names as the source configuration file. However, you can override these values if required.

- **Delete Location**
  Deletes the current selected location.
- **Save Configuration**
  The current configuration will be written to the configuration file, MQMON.CFG.
- **Preferences**
  Displays a dialog to allow the user to change some global parameters on the way the program operates, such as saving dialog positions. See “Preferences Dialog” on page 193 for a description of the dialog.
- **Print**
  Print the list of Queue Manager and location names. See “Printing” on page 107 for more information.
- **About**
  The About dialog gives version and build date information.
- **Check for latest MO71 version**
  Displays a dialog showing the current version of MO71 in use and the latest available version ready for download. Clearly this function requires an internet connection. If the internet connection fails then a message box will be shown explaining the failure.

  By default MO71 will check for program updates periodically and display a dialog if a later version is available. This feature can be switched off if required in the preference dialog.
- **Show IBM MQ Maintenance Levels**
  Displays a dialog showing the latest MQ maintenance levels available for the IBM MQ products. Clearly this function requires an internet connection. If the internet connection fails then a message box will be shown explaining the failure.
- **Help**
  Will try to display this manual or configured URL. Note that the manual file should be placed in the same directory as the program or its location identified in the *‘Help file location’* preference field. Note that 'F1' will also display help.
- **Exit**
  Exits the program[40].
  The current configuration will automatically be written to the configuration file.

[40]The first time the user requests to end the program it will disconnect from all the connected Queue Managers before shutting down. It is possible that this process will hang because, say, the network connection has been lost. Selecting ‘Exit’ again will end the program immediately bypassing the disconnect processing.

24.1.2. Commands

Various menu items to invoke a command dialog against the selected Queue Manager location. The list of commands available will depend on the platform and release level of the selected location.

24.1.3. Action

- **MQSC Window...**
  If commands are allowed for the selected location an MQSC window will be presented where MQSC commands can be entered and responses from the command server displayed. This has the advantage that any MQSC command supported by the command server can be entered without requiring MO71 to support it directly. No parsing of the command or response is made other than formatting the data to fit the screen. See “MQSC Window” on page 26 for a description of how to use this window.
- **Predefined Dialog list...**
  Shows a dialog containing the currently defined predefined dialogs.
- **Predefined Dialog...**
  Shows a dialog allowing the creation of a predefined dialog. See “Predefined Dialogs” on page 168 for a description of predefined dialogs.
- **Event List...**
  Shows a dialog of the currently defined Events. See “Event Monitoring” on page 126 for a description of how to use this window.
- **Event...**
  Shows a dialog allowing the creation of an event. See “Event Monitoring” on page 126 for a description of how to use this window.
- **User Command List...**
  Shows the current list of user commands. Please see “User Commands“ on page 172 for more information.
- **User Command...**
  Shows a dialog where a new user command can be creasted. Please see “User Commands“ on page 172 for more information.
- **Filters...**
  Displays the Filter Manager dialog. See “Filter Manager” on page 52 for a description of how to use this window.
- **HTTP Connections...**
  Displays the current browser connections. See “Who is connected ?“ on page 165 for more information.
- **Compare Definitions**
  Displays a dialog comparing the object definitions of two different Queue Managers. See “Comparing Queue Manager Objects” on page 145 for a description of how to use these dialogs.
- **Local Error Log File...**
  Selecting this option will display the contents of the local error log for this location. For example, if this location is accessed via an MQ Client then the client error is displayed.
- **Local INI file...**
  Selecting this option will display the contents of the local INI file for this location. For example, if this location is accessed via an MQ Client the client INI file is displayed,
- **Put Message...**
  Will display a simple dialog for the selected location to allow a simple message to be put to a target queue. The message can only be a string (‘MQSTR’ format) but the user can choose how many messages are put and whether persistence or syncpoint is used. The user can also choose whether the message content comes from the dialog string or a file.
- **Publish Message...**
  Very similar to the ‘Put Message’ above but in this case the message is published to the given Topic. The user can choose between publishing the message using the MQI directly or using queued publish depending on which TAB is selected.
- **Load/Unload Queue...**
  Displays a dialog allowing the user to load/unload a queue from/to a file. See “Queue load/unload facility” on page 68 for more details on how to do this.
- **Talk...**
  Presents a dialog which allows the user to type a text message and send it to the given Queue Manager. The remote location must be running another instance of MO71 or other such program which will display the message.

- **Trace Message...**
  Presents a dialog which will show, in a diagram, the route messages took when sent from a source Queue Manager to a target destination.
- **Activity Trace...**
  Shows a dialog where the user can analyse a activity trace messages. Please see “Activity Trace“ on page 175 for more details.
- **API Exerciser...**
  Will display a dialog allowing the user to issue direct verbs to IBM MQ. This is useful to test the effects of certain verbs with certain options. See ‘API Exerciser’ on page 132 for more defaults.
- **Export All Shown**
  Is the equivalent of pressing “Export Now” for all the shown locations.
  You can use the main window search field to choose which locations this menu applies to.
- **Disconnect**
  Normally the application will disconnect from a Queue Manager based on the disconnect interval specified in the location dialog. However, if the user wishes to force disconnection earlier than this then this menu may be used.
- **Disconnect All Shown**
  This command will disconnect from all the locations shown in the list if they are currently connected.
  You can use the main window search field to choose which locations this menu applies to.
- **Connect All Shown**
  This command will attempt to connect to all the locations shown in the list if they are not currently connected and it is appropriate. For example, MO71 will not directly connect to locations which are processed 'via' another location. Connecting to a group of Queue Managers can be useful to make a quick check that the set of Queue Managers are running correctly and are available for connections.
  You can use the main window search field to choose which locations this menu applies to.

24.1.4. View

- **View Network....**
  Displays the network view dialog. See “Network View” on page 79 for a description of how to use this window.
- **View Applications...**
  Displays the application view dialog for this Queue Manager. See “Application View“ on page 96 for a description of how to use this window.
- **View Console**
  Displays the console window.
- **Queue Manager name**
  Shows the list of locations in the main window by it's Queue Manager name.
- **Location**
  Shows the list of locations in the main window by it's location description.
- **Set Font**
  There are two fonts which can be set within the program. One used for ‘Windows’ and one for displaying ‘Data’.
  The windows font is used as the font for the main window and the command windows.
  The data font is used as the font for displaying messages on queues, the MQSC window and the Activity Trace display. It is recommended that the user select a non-proportional font, such as Courier, for the data font since displays of this type are easier to read when the columns of text are aligned on multiple rows.
- **Set Colours...**
  Displays a dialog allowing the setting of the colour of various parts of the dialog windows.
- **Set Default Lists**
  Under this menu are the various categories of dialogs that MO71 supports. In each category are menus for each of the list types supported by the program. Selecting a menu will present a dialog allowing the user to modify the attributes displayed when a list is selected for a location. Note that the user will be able to choose from all the attributes appropriate for **all** the locations added to MO71. However, when a list for a particular location is displayed only those attributes that are appropriate for this particular location will be shown. If an additional location is added which is a later level of Queue Manager those already defined then it may be appropriate to add new fields to the lists. See “Alter list dialog” on page 23 for an explanation of this dialog.
- **List view**
  Displays the list of locations in a list box.

- **Tree view**
  Displays the list of locations in a tree view. If the Preference setting “Display objects in main window” is selected then each location can be expanded to include the major objects for that Queue Manager. For channels and queues the icon changes depending on the state of the object (see “Appendix A: Icons” on page 267 to see what the icons look like).s

  Note that the state of the object displayed will be the state the last time the information for that queue manager was refreshed. To refresh information for a location select the location and press *‘Refresh Information’* from the File menu or use the refresh icons in the network display.

- **Search**
  When this option is selected a search field is shown at the top of the main window Queue Manager list. Entering text in the search field will automatically cause the list to be refreshed so that only locations containing the text will be displayed. The search text can contain the standard wildcard characters '*' and '?'. The full format of the search string is a comma separated list of search strings followed by an optional search qualifier which says which location field should be compared. Without a qualifier MO71 will check all the fields below.

Location Field	Qualifier	Qualifier Shortform
Queue Manager name	qm	m
Location	loc	l
Group Name	grp	g
Network Names	net	n
Client Connection Name	conn	c

  If you choose to use a qualifier it should be specified between < > characters.

  Examples of search strings might be:

Search String	Meaning
abc	Search all fields for the text 'abc'
abc <qm>	Search only the Queue Manager name for the text 'abc'
abc,def	Search all field for either text 'abc' or 'def'
abc,def <qm loc>	Search for either text 'abc' or 'def' but only look in the Queue Manager or Location fields
abc,def <m l>	As above but using short form qualifiers
abc*def	Search for a single string containing 'abc' and 'def' with zero or more characters between them
abc?def	Search for a single string containing 'abc' and 'def' with a single character between them

  Searches are case insensitive. The search will applied after the configured delay interval. This is set in the ‘Network’ tab of the preference dialog.

- **Non-responders only**
  When this option is active only the locations which are not responding to monitor requests will be displayed. This is useful if you have many, many locations and just want to show the ones with problems.

- **Show MQ Version**
  When this option is active the MQ Version (if known) of the location is added to end of the displayed entry.

- **Show Foreign locations**
  Controls whether Foreign locations are displayed in the main window.

- **Show Monitor Time**
  If selected MO71 will display the last time a monitor response was received from this Queue Manager when monitoring is active. The format of the time display is controlled by the preference setting in the 'Time' tab in the Preferences Dialog described on page 209.

24.1.5. Monitor

This menu contains menu items concerned with monitoring of one form or another.

- **Enable Alarms**
  When checked the application will alarm whenever an 'alarm' situation is detected. No audible alerts will be given without this option set.
- **Reset Alarm**
  Select this option to stop the alarm sounding. The alarm will sound again if another 'alarm' situation is detected.
- **Reset Location**
  If an attempt is made to contact a location which is not available the icon for that location will show an error. This error condition is normally only reset when connection is made. However, by selecting this menu option the location can be manually reset.
- **Reset All Locations**
  As above but this menu applies to all locations in error.
- **Enable Monitoring**
  When checked the application will send messages periodically to any location which is not disabled for monitoring. See “Queue Manager Monitoring” on page 109 for more details on monitoring.
- **Monitor all**
  Force the sending of monitor messages to all non-disabled locations.
- **Graph...**
  Displays a new graph window. Items to be graphed need to be added using the Graph Dialog menu, see Graph Options Dialog on page 115 for more information.
- **Monitor List...**
  Displays the list of attribute monitors defined.
- **Monitor...**
  Displays a single attribute monitor definition.
- **Monitor List Status...**
  Displays the list of the status of the attribute monitors.
- **Monitor Status...**
  Displays a single attribute monitor status.

24.1.6. Windows

Menu items concerned with window and dialog management. MO71 allows the user to create as many windows of any type as the user wishes. This can be extremely useful if, for example, you want to look at a list of queues, another of channels and one of channel status and possibly the same for a multiple queue managers. You can get a large number of windows and some options to manipulate them can be useful.

- **Close all**
  Closes all windows
- **Minimize all**
  Minimizes all windows
- **Tile Horizontal all**
  Tiles all the non-minimized windows and favours making the windows 'horizontal' – suitable for lists. If windows are not put in the ideal position then two windows can be 'swapped' by moving the window with the <Ctrl> key held down. The window swapped with will be the window whose top left is closest to the top left of the window being moved.
- **Tile Vertical all**
  Tiles all the non-minimized windows and favours making the windows 'vertical' – suitable for object dialogs.
- **<Windows>**
  Most dialogs and windows opened in MO71 will be displayed in this list. By selecting an entry the window will be brought to the foreground and restored.

24.2. Preferences Dialog

The preferences dialog allows the user to set the behaviour of various aspects of the application. The dialog is split into a number of different parts. The first being a set of on/off switches in a scrollable list. The item is on if a tick is displayed next to the item.

24.2.1. General

- **Auto save configuration**
  When checked MO71 will automatically save the configuration when a significant change, for example adding a new location, is made.
- **Import local Queue Managers on startup**
  This option controls whether MO71 will check whether there are any local Queue Managers defined which do not exists in the MO71 configuration. If there are missing Queue Managers a dialog will be displayed on startup allowing the user to import the definitions.
- **Show objects in main window**
  If checked then the locations in the main window can be expanded to show the main objects of the Queue Manager at that location. Most users use the command menu to get the lists which provide a much richer set of facilities with the returned objects.
- **Save object definitions**
  With this option checked the application will save the key fields in the Queue Manager objects in a file called MQMON.DFN in the program path directory. This allows the application to start with a list of the objects defined on each Queue Manager. The user must select refresh information or hit the refresh button in the network display to update this data.
- **Refresh object definitions when first displayed**
  When selected the application will automatically refresh the object definitions the first time the location is expanded in the main window. This requires that the 'Show objects in main window' is also selected.
- **Show frequent menus**
  When checked the command menu will display the menus most frequently used. The number of menu items shown is controlled by the *'Menu items'* setting in the Preference 'Display' tab.
- **Show recent menus**
  When checked the command menu will display the menus most recently used. The number of menu items shown is controlled by the *'Menu items'* setting in the Preference 'Display' tab.
- **Show list menus**
  When checked the command menu will display the menus concerned with showing lists of objects.
- **Show menu in categories**
  When checked the command menus will be grouped in to categories such as Channels, Queues, Security etc to make finding the required command easier.
- **Show all menus in categories**
  If unchecked then only menus which haven't already been shown will be included in the category. If checked then the full set of commands for that category will always be shown.
- **Show object names in menus**
  When checked the dialog names displayed in the *'Windows'* main menu will contain any appropriate object name.
- **Confirm on program exit**
  When checked MO71 will always ask for confirmation before ending. This can be useful to prevent accidental ending of the program.
- **Show confirmation dialogs**
  When checked the application will display a confirmation dialog whenever a potentially harmful command such as "delete queue" is issued.
- **Confirm message operations**
  When checked destructive message operations such as 'copy' and 'delete' will cause a confirmation dialog to be displayed.
- **Confirm message 'all' operations**
  When checked message operations which apply to all messages, such as 'copy all' and 'delete all' will cause a confirmation dialog to be displayed.

- **Confirm on object change**
  When checked the application will show a confirmation dialog if the user tries to update an object definition after changing the object name from the one whose details are shown. This can prevent accidental changing of the object definition.
- **Confirm running of MQSC file**
  When checked the program will show a confirmation dialog whenever it is about to run an MQSC command file.
- **Include the MQ Maintenance level in the MQ version if known**
  In the main window and the Network View display you can choose to have the Queue Manager name appended with the MQ Version that Queue Manager is running. This option allows you to go one step further and also have the MQ maintenance level, such as the fix pack number, appended if it is known.
- **Show MQ Version in dotted format**
  The MQ Version attribute of the Queue Manager is in the format VVRRMMFF such as '08000004'. This can be quite difficult to read. By selecting this option MO71 will display this value as 8.0.0.4 which is far easier.
- **Menu Items**
  The number of menu items which should be displayed at the top level when using the preference options frequent or recent menu items.
- **Notify of program updates**
  MO71 has the ability to periodically check for program updates. This allows you to set three different modes.
  - **Never** MO71 will not check for updates
  - **New version only** MO71 will only notify the user if a new version is available
  - **New version or build date** MO71 will notify the user of any newer version

  Clearly this feature requires an available and working internet connection. If, for some reason, the web page cannot be accessed then a message box explaining the issue will be displayed to the user.
- **Startup Splash, Qm Normal Splash, Qm Specific Splash**
  These settings are all concerned with displaying a user configurable splash screen to the user. For more information about splash screens please see “Splash Screens“ on page 103.
- **Licence Command**
  This setting allows the user to specify a command which will run should there be either an invalid licence or it is soon to expire. For more information please see “Licence Command“ on page 112.

  The licence command is followed by a 'Test' button. This button has two purposes. Firstly it allows you to run any configured licence command and will display an error dialog if there are any problems. Secondly, whether you have configured a licence command or not, it will re-load any configured licences be they **MO71**, **MQEV** or **MQEdit**. This allows you to change the install licences without having to end the program.
- **Group Text**
  This setting controls what is displayed against each group in the main window. The following text inserts are available:

  %c Number of connected Queue Managers

  %f Number of failed Queue Managers

  %n Group Name

  %t Number of Queue Managers in the group

  %u Number of Queue Managers with unknown connection state.

  If any text is put in curly brackets {} then the text will only be shown if the first insert has a non-zero value.
  Entering an empty string for this value will re-set it to the default value of **%n (Total:%t{ Connected:%c}{ Failed:%f})**

24.2.2. Dialogs

- **Save Dialog Positions**
  When checked the positions of the command dialogs will be saved
- **Save Dialog Sizes**
  When checked the sizes of command dialogs will be saved
- **Integral monitor positioning**
  Normally MO71 will try to ensure that new dialogs are initially presented so that they do not span between multiple monitors,

even if that was the last saved position of the dialog. By unchecking this option this behaviour is switched off and dialogs will be presented exactly where they were last.

- **Adjust Dialog positions for dual monitors**
  When this option is checked MO71 remembers the position and size of dialogs independently for when the programs is running with one or two monitors. This reduces the chance that a new dialog is placed on the second monitor which isn't there any more.[41]

- **Show object name field as a dropdown list**
  When checked the main object name in a dialog is show in a dropdown list. This has the advantage that the list of recently used names can be scrolled through. However, it does have the disadvantage that the value can be inadvertently changed by moving the scrollbar. If required this option can be unchecked which will display the field in a normal entry field.

- **Show thousands separators**
  When checked numerical values in dialogs will contain thousand separators. So, for example, by default 4194304 will be displayed as 4,194,304. If you wish you can change the separator character which is used at the bottom of this tab.

- **Display dialogs on taskbar**
  By default the dialog windows are displayed on the windows taskbar. However, if you have many of them then this can use up valuable taskbar space. By deselecting this option the dialogs will not be on the taskbar but can be easily accessed via the 'Windows' menu item on the application main window.

- **Show refresh time in titlebar**
  This option controls whether MO71 will display the last refresh time in the titlebar in some way.

- **Refresh time in title**
  This option determines whether the refresh time is put in the normal dialog title or should MO71 try to draw the refresh time on the right-hand side of the window in the titlebar. However, this facility may not be available in some Windows themes. In particular Window 7 Aero themes don't allow applications to draw over the non-client area of the titlebar. To see the refresh time you may need to switch to the 'classic' theme or check this option so the refresh time is part of the actual dialog title. Another alternative is to run the program in compatibility mode – for example Window XP.

- **Refresh time on the right**
  This option applies on when MO71 is putting the refresh time into the dialog title and controls whether the time is put on the left or right of the other title text.

- **Show object name in dialog title**
  This option controls whether the object name (if any) associated with the dialog should be included in the title of the dialog.

- **Show tooltips as balloons**
  When a list dialog contains more columns than can be displayed across the screen the titles of the columns are compressed. Hovering the mouse over the column title will display the full name of the field in a tooltip. This option controls whether the tooltip is displayed as a balloon or a simple rectangle. Note that changing this option will only take effect the next time the application is started.

- **Include MQSC name in field title**
  This option controls whether the MQSC name is also included in the field title on the left-hand side of the object dialogs.

- **Up/Down keys should tab between fields**
  Versions of MO71 prior to 5.3.3 used the up/down keys to tab between fields in a command dialog. The disadvantage of this is that to select from a pull-down list required actually bring the pull-down list down. By disabling this option the user can scroll through the available choices of a pull-down with the Up/Down buttons. The tab button should be used for tabbing between fields.

- **<esc> should end dialog**
  When selected the escape key will end the current dialog.

- **Allow selection of read only fields**
  This options changes whether read only fields in the object dialogs can be selected. Allowing them to be selected means there can be more fields to tab over but it also means that you can select text to copy/paste into other non-read only fields.

- **Check for printable characters on input**
  It is possible to enter unprintable characters in to a windows entryfield. As a consequence these characters can be 'accidentally' entered into, say, an object description. By checking this option the application will check that all characters on input are printable before adding them to the object. Input is truncated to the last printable character.

[41] Note that MO71 does actually know whether a second monitor is physically connected or not. If the Windows settings says there are two monitors but there is only one connected then dialogs will still be placed on this non-displayed monitor.

- **Reflect dialog changes to predefined dialog definition**
  When checked changing the attributes of a dialog created using a predefined dialog definition will be reflected back to that definition. For example, re-sizing a predefined window will change the size of the values in the definition of the pre-defined dialog. Please see "Predefined Dialogs" on page 168 for information on predefined dialogs.
- **Double click should *Switch* on predefined dialogs**
  When checked double clicking on a predefined list will switch focus to the instance of the predefined dialog rather than bring up the definition of predefined dialog. Whatever the setting of this value holding **'Alt'** while double clicking will get 'the other' behaviour. Please see "Predefined Dialogs" on page 168 for information on predefined dialogs.
- **Autostart predefined dialogs**
  When checked predefined dialogs will be started automatically when MO71 itself is started.
- **Restore running dialogs across restarts**
  Normally, any dialogs running when the application is ended will be restarted automatically when the application is restarted. This option should be unchecked if this behaviour is not required.
- **Show existing dialog by default**
  When the user requests a dialog, for example Queue List, this option controls whether any existing Queue List dialog for this location should be displayed or whether a new one should be created. Holding down the <Shift> key when the dialog menu is selected will give the alternate behaviour than specified in the preferences dialog. For example, suppose we already have a Queue List dialog for location X. With 'Show existing dialog by default' selected then pressing the Queue List menu for location X will display the existing Queue List dialog. However, holding down the <Shift> key and then selecting the Queue List menu item will cause a new Queue List dialog to be created.
- **Subkey dialog check**
  Some objects do not have unique names. For example, it is possible to have multiple cluster queues with the same name or multiple channel status objects for the same channel. In these cases the application uses a subkey to determine which object is which. For example, in the case of a cluster queue the cluster queue manager name is used to uniquely identify the objects returned from the Queue Manager. By selecting this option this subkey checking can be switched off.
- **Use global MQSC command history**
  When selected commands entered into any queue manager are available in the drop down list on any other queue manager. If not selected only the history of commands entered on 'this' queue manager are shown. Press 'alt' when dropping down the list will show the 'other' list.
- **Execute MQSC file selection immediately**
  If the user specifies ***file(?)*** on the MQSC window then a file selection dialog is displayed. If an MQSC file is then selected this option controls whether the file is executed immediately or whether the selected file is merely displayed on the MQSC dialog window.
- **Strip topic stem from subscription topic strings**
  Prior to MQ V8.0 for some strange reason, known only to the MQ development team, a display of subscription that was created with a TOPIC object will not contain the same value in the topic string as was used in its creation. Instead the command server prepends the value of the TOPIC object so you get a full qualified topic string. This can be misleading and it causes problems when exporting the subscription in MQSC format since the new definition ends up with the TOPIC stem twice. So, MO71 will automatically remove the TOPIC stem if configured to do so. The value of this setting will affect subscription dialogs and subscription list dialogs but will not affect the Queue Manager export settings. The strip option must be set if you wish to export subscriptions in MQSC format from the subscription dialogs.

  Since this inconsistency was fixed in MQ V8.0, with the addition of 'display type', this option only applies prior to MQ V8.0.
- **Auto-dismiss progress dialog**
  If this option is checked then a progress dialog will be closed if it completes without error. Note that the progress will always be closed if it was never actually shown to the user.
- **Always keep CRLF from Put Dialog files**
  By default **MO71** will strip out the CRLF character combinations from loaded files and replace them with just LF characters if the put is targetting a non-Windows platform. However, by checking this option you can request that MO71 leaves these CRLF characters untouched.

- **Allow display of full remote product name**
  MQ uses four character product identifiers for the product name shown in, say, the channel status command. This options controls whether **MO71** just display this four character id or whether the full product name is shown. The ids which MO71 understands are as follows:

Product Id	Full Product Name
MQCC	MCC – 'C' Client
MQJB	MQJB – Classes for Java
MQJF	MQJF – MFT Agent
MQJM	MQJM – Classes for JMS
MQJN	MQJN – JMS (migration mode)
MQJU	MQJU – Java interface for MQI
MQMM	MQMM – Distributed Queue Manager
MQNC	MQNC – HP NonStop Client
MQNM	MQNM – .NET full managed Client
MQNU	MQNM – .NET unmanaged Client
MQVM	MQVM – z/OS QMgr
MQXC	MQXC – XMS client C/C++
MQXD	MQXD – XMS C/C++ (migration mode)
MQXM	MQXM – XMS .NET (migration mode)
MQXN	MQXN – XMS client .NET
MQXU	MQXU – XMS .NET (unmanaged/XA)

  A separate option in the 'Lists' tab controls whether the full name should be shown in a list dialog.

- **History of Object Names**
  How many object names should be kept in the dropdown history

- **Topic Cache Max Age**
  In order to strip the TOPIC object stem from the displayed subscription topic string values MO71 needs to query the TOPIC object definition. MO71 will use this setting to determine whether it 'needs' to query the TOPIC value. In many installations the TOPIC objects very rarely change so it may be reasonable to set a high value for this max age so that MO71 doesn't continually ask for the TOPIC definitions. The lowest value which can be set is 5 seconds which means that the TOPIC object definition will age out pretty quickly.

- **Progress Dialog Delay**
  This is the number of seconds a message operation should should run before the progress dialog is shown. The default value is 3 seconds. A value of 0 will display the dialog immediately. A value of 'Never' requests that Progress Dialogs are never created. Setting a large value, for example, 99999, will effectively only display the dialog if the user explicitly chooses it from the 'Windows' menu of the main window.

- **Separator thousands**
  Here you can specify the character which is displayed as the thousands separator if thousands separator option is checked. This allows you display numbers such as 4194304 as 4,194,304. You can specify up to 3 characters. The default is ','.

- **Separator Lists**
  Here you can specify the characters which are displayed between a list of values. For example, the list of names in a namelist, or the list of channel exits defined on a channel. You can specify up to 3 characters. The default is '; '.

- **Separator Indicators**
  Here you can specify the characters which are displayed between indicator values such as QTIME or NETTIME. You can specify up to 3 characters. The default is ' : '.

24.2.3. Lists

- **Right justify numeric columns**
  When selected the majority of numeric data will be shown right justified. Exceptions to this are 'identifier' type numerics such as monitor identifier, queue position and codepage.
- **Show QM list initially**
  When checked the Queue Manager list pane is show by default on appropriate list dialogs.
- **Lists should be striped**
  When checked list dialogs are candy striped. The main colour and the stripe colour can be set in the View-Set Colours… menu item. Please see “Colour Dialog” on page 229 for more information.
- **Show rule lines on lists**
  When checked list dialogs will be displayed with a ruled line underneath each item displayed in the list. The colour of this line will be the list selection background colour.
- **3D List Titles**
  When checked the titles in list dialogs appear like three dimensional buttons. Regardless of this setting the titles will always behave like buttons and allow sorting based on the field title selected.
- **3D List Sort Arrow**
  For most colour schemes the arrow displayed in the list titles to identify the sorted field looks better when displayed in a 3D fashion. However, for some colour schemes it may be preferable to switch off the 3D attribute.
- **Select first entry on refresh**
  If checked the first entry of the lists will be selected when the list is refreshed if the list has no current selection.
- **Filled List Sort Arrow**
  For the same reason as the previous option it may be preferable to colour fill the sort arrow with the foreground colour.
- **List displayed using non-proportional font**
  By default the list data is displayed in the proportional font since most people find that easier to read. However, if you prefer you can check this option which will cause the list to be displayed in a non-proportional font. This has the advantage that the columns of data are more closely aligned between each row of data.
- **Show Column Filters**
  Column Filters are shown as a small icon to the right of a column title in all the list dialogs. By default these icons are shown but if you don't use Column Filters you can choose not to display them by unchecking this option.
- **Use small font for column filters**
  By default the Column Filter dialog will be shown with a slightly reduced font size. However, by unchecking this option you can choose to have the dialog shown using the full size font.
- **Use column filter background colour when enabled**
  If a column filter is enabled but not currently active the background colour will be shown slightly darker than a non-enabled column filter. This allows you to easily see which Column Filters are currently in-play. However, depending on the colour scheme this darker colour may be difficult to see. So, by enabling this option you can choose to display an enable Column Filter icon using the Column Filter colour. You can set the colour of the filter window by selecting *'View-Set Colours...'* from the main window. The filter windows colours are set by changing the 'Menu background' and 'Menu Foreground' colours. For information about setting colours please see Chapter 24.10.Colour Dialog on page 229.
- **Show column filters over column title**
  By default the Column Filter icon will only be shown if there is sufficient space in the column width after the Column title and sort arrow is drawn. However, by selecting this option you can choose that the Column Filter icon is shown in preference to any title text. The Column Filter icon will be drawn over the top of any Column title text.
- **Display full remote product name in lists**
  For a list of remote product names please see the table in the 'Dialogs' tab description. This setting controls whether the full product name, rather than just the identifier, should be shown in a list.
- **Show compare difference using change colour**
  When using the compare dialog and highlighting differences you can choose whether the difference is highlited using a difference background colour. This colour can be chosen in the 'Colour Dialog' described on page 229.
- **Double-click field is significant**
  To bring up an object dialog from a list dialog you either double-click on the line or select 'Open...' from the context menu. This preference option chooses whether the actual column you click on is significant. If the setting is checked then the object dialog will be shown with the focus on the selected field. If this option is not checked then the dialog will be shown with the first field active, usually the object name.

- **Raised, Bump, Sunken, Etched, Flat, One dimensional, Half, Softer**
  These options can be used to change the way the list items are displayed. These options control calls to the windows DrawEdge function and the exact display results depending on the implementation of that system. In particular, in my experiments, **Softer**, has no effect on most of the border types.
  The **Half** option causes the border to be drawn only on the bottom and right hand sides of the box. This means that although the appearance is softer it also reduces the 3D look of the border.
- **3D Selection**
  When selected the selected items in the dialog listbox will be drawn in a 3D fashion. The effectiveness of this option will depend on the colour scheme being used.
- **Show multi-fields in list dialog**
  Some object values can be lists such as the names in a namelist object or channel exits in a channel definition. When displaying a list of these objects the application needs to know whether to display all of the values in the list or just the first one. Normally you would want this option checked but if you have channel definitions with a large number of channel exits the list might be excessively large and so it might be worth switching this off.
- **Auto update lists**
  When checked the application will refresh list displays if actions may have caused the contents to change. For example, creating or deleting a queue would update the queue list display. The application cannot always determine which lists may have changed however so the user should periodically refresh the display or set auto-refresh on the dialog if you want to be certain you are looking at the latest data.
  Auto updates can be disabled per location in the location dialog.
- **Reposition list on sort**
  If checked the list is repositioned, after a sort, to try and display the same entry as displayed before the sort. If not checked the top of the list is show after sorting.
- **Size columns by data portion only**
  When displaying a list of objects the column width are automatically set by the greater of the title width and data width. Some fields, generally numerics, tend to be quite narrow despite having fairly large titles. By selecting this option the column width will only be sized based on the data rather than the title. Note that this can make the title unreadable.
- **Remove old list data**
  If this option is checked then old data will be removed from the list display if no response is received, from a Queue Manager, for a refresh after 'Maximum Refresh Response Time' seconds.
- **Copy main Queue Manager Search field**
  If checked then when a new list dialog is started the contents of the main window Queue Manager search field will be copied to its search field if it supports multi-QM display. Note that the Queue Managers will not be selected but they will be filtered in the Queue Manager pane. This means that if, temporarily, you wish to work on a subset of your Queue Managers it can make it easier to find the Queue Manager you are interested in.
- **Ignore surrounding blanks on compare**
  Controls whether leading and trailing blanks are considered 'different' when comparing objects.
- **Compare all definitional fields**
  Controls whether all definitional fields are compared or only those that make sense. For example, Inhibit Get and whether triggering is enabled can change during the life of a Queue but should they be included in the comparison.
- **Show object DISPLAY authority failure count**
  When MO71 requests a list of objects from a Queue Manager the Queue Manager **can** notify MO71 that some objects were not returned because the user was not authorised to view them. Using this option you can control whether MO71 will display how many objects are not shown due to lack of authority. Bear in mind though that this feature is dependant on the Queue Manager sending these responses. It is possible for some Queue Managers to suppress these responses.
- **Check whether lists are complete**
  When checked MO71 will verify that for each object list request the right number of message responses are returned. This is done by counting the response messages which should match the Message Sequence number of the last response. This can be useful to inform the user that some messages have 'gone missing'. The usual reason for this is that a Reply Queue or Transmission Queue has filled up causing messages to be put to a Dead Letter Queue somewhere.
- **Check MAXDEPTH of Reply Queue**
  A Reply Queue should always be capable of holding the reply messages. If the Reply Queue fills then messages may be put to the Dead Letter Queue causing incomplete lists or delayed responses as the Dead Letter Queue handler re-puts the messages. To avoid these problems it is recommended that the MAXDEPTH of the Reply Queue is always at least twice as big as the largest number of items that are requested. There is little downside to making the MAXDEPTH of your Reply Queue **much** larger than it needs to be. If this option is checked then MO71 will check that no list response contains more items than half the

MAXDEPTH of the Reply Queue. If this option is checked then the Reply Queue is opened for INQUIRE since MO71 needs to query the MAXDEPTH value. For this reason you may wish to uncheck this check.

- **Use Global filter history**
  If checked then the history of filters used in the list dialogs will the same for each location. If unchecked then a separate list will be maintained per location. Holding the 'Alt' key when showing the filter history will show the 'other' history. So, for example, if you chose to use the non-global filter history then by holding the 'Alt' key as you displayed the filter list the displayed list would be the global list,
- **Import shared filters**
  This option controls whether MO71 is a generator or a consumer of shared filters.
  For a description of shared filters please see Shared Filters on page 53.
- **History of filter expression**
  How many filter expressions should be kept in the drop-down history
- **Maximum Refresh Response Time**
  The maximum number of seconds a list dialog should wait for a response to a refresh command before it should consider the Queue Manager unresponsive. If the option 'Remove old list data' is checked then any data for this Queue Manager will be removed from the list.
- **Default Max. column width**
  This option controls how wide a list column should be by default. This ensures that large fields do not make the lists hard to read because all the other fields are pushed to the right. If the column is not wide enough to display the data MO71 will display ' … ' at the end of the field. The user can make the column wider if they wish to see more of the data. A value of 0 implies that no limit should used.
- **List Flash Interval**
  How many seconds between flashes for for lists that have used the flash() or flashcell() functions.
- **Shared Filter File**
  The location of the shared filters. For a description of shared filters please see Shared Filters on page 53.
- **Queue Manager Search add without confirmation limit**
  In a multi-QM list dialog it is possible to use tools to the right of the Queue Manager search field to select multiple Queue Managers in a single click. When a Queue Manager is selected MO71 may have to connect to the Queue Manager and retrieve the data. It follows therefore that if you are selecting a large number of Queue Managers then it could cause a significant amount of work and traffic. To mitigate the risk of the user clicking on the tool accidentally MO71 will confirm that it was intentional if the number of Queue Managers exceeds this configurable limit.

24.2.4. Browse

- **Convert EBCDIC for display**
  If checked then MO71 will try to display the EBCDIC data in messages as readable characters. This will probably make messages easier to understand but can be misleading as to their content.
- **Warn about possible data conversions**
  If selected then MO71 will write a warning message in the status display if some of the characters shown have been converted from EBCDIC by MO71. In addition, the text '(after conversion)' is added to any CCSID values if conversion is selected and the code page is the local codepage. This text reminds the user that the message on the queue may be in a different code page.
- **Predict Hexadecimal characters**
  When browsing a queue the user can request that hexadecimal fields such as the Message Id are displayed in character form as well as in hexadecimal. It is not clear, however, whether these fields will be in ASCII or EBCDIC format. If this item is checked the application will display ASCII or EBCDIC depending on the number of characters falling into the alphanumeric range in each of the character sets. If not checked the value will be assumed to be ASCII.
- **Double click should browse on queue lists**
  By default double clicking on an object on a list window will show its definition. In the case of browsable queues though it may be more convenient to have double click bring up the browse screen. Whatever is chosen as the default action, definition or browse, holding the '**ALT**' key and double clicking will perform the other 'non-default' action.
- **Disable read ahead**
  Read Ahead can give significant performance advantages particularly over slow networks. However, it can also cause more data to be retrieved from the server than necessary. When checked this option switches off read ahead for all locations, both the reply queue processing and browsing message queues. For more granular control the user can also disable read ahead by location in the option section of the location dialog. For more information please see “Location Dialog“ on page 214.

- **Auto update browse lists**
  When checked the application will refresh browse queue displays if actions may have caused the contents to change. For example, putting, copying, moving or deleting messages would update the queue browse display. The application cannot always determine which lists may have changed however so the user should periodically refresh the display or set auto-refresh on the dialog if you want to be certain you are looking at the latest data.
  Auto updates can be disabled per location in the location dialog.
- **MQEdit Installed**
  This option controls whether the 'Edit' option is available when browsing a message. By default MO71 will check whether the licence file contains a valid licence for the MQEdit product. Alternatively you can check this option manually if you have installed the MQEdit licence in a different place.
- **Default Msg Display Range**
  This option allows the user to set the default browse range when browsing a queue. By default it has the value '1,100' which is the first 100 messages. Any value can be set but bear in mind that large values could cause large numbers of messages to be retrieved from the server with the related impact on performance.
- **Default Max browse message size**
  When browsing messages in a queue the message size is limited to a maximum size which is displayed on the dialog. This prevents accidentally downloading very large messages from the queue and enables quick navigation of the messages. The default maximum message size is 10,000 bytes but if you frequently want to browse messages larger than this you might like to increase this default size.
- **Default Max Browse Messages**
  MO71 supports browsing up to 30,000 messages at once. However, the default is 10,000 to avoid unnecessary delays when browsing deep queues. Essentially this value controls how wide the message display range display range. Even with a small maximum browse range, and therefore a small display range it is still possible to browse deep queues. For example, to see a few messages near the 100,000th message one could specify a range of 99995,100005.
- **Message Summary Size**
  Message summary is feature which summarizes the message based on it's first few bytes. For example, a message on the Dead Letter Queue will have a header which identifies the cause of the message being put on the Dead Letter Queue. A message on the transmission queue has a header which identifies where the message it going. This setting controls, in a message list, how many bytes of the message are requested so that the message can be summarized. Of course the higher the setting the more detailed the message summary for some messages at the cost of retrieving more data from the server.

  The minimum value is 108, the maximum value if 500 and the default value is 120.
- **Message Summary Display**
  This setting controls how many bytes of message data is displayed if the message cannot be summarized in the browse message list. If a message can not be summarized, for example it is just a bytes message with no format or is not a recognised format, then MO71 will just display the first few bytes of the message as characters (if they are printable). This setting allows you to set how many of those bytes are displayed.

24.2.5. Connection

- **Auto start command server**
  With this option checked the application will check whether the command server is running when it connects to a local Queue Manager**.** If it isn't it will be started with the 'strmqcsv' command.
- **Uppercase MODEL Queue userid**
  When using a model queue as a location reply queue the application will construct the name of the queue based on the application name and the userid running the program. For some security profiles it may be useful to always have the userid folded to uppercase.
- **Use MQSET to free reply queue**
  When MO71 tries to open a reply queue for a location it might find that queue already in use. This may happen because a previous instance of MO71 was not ended cleanly. By default, MO71 will then inhibit get on the reply queue forcing any outstanding get to return. This then allows the current instance of MO71 to open the reply queue successfully. The ability to issue MQSET against a queue does, however, require a higher level of authority which may be inconvenient. This MQSET processing may be switched off by unchecking this option.
- **Reconnect 'not authorised' failure**
  This option controls whether MO71 should reattempt the connection if it gets a 2035 (not authorised) failure.
- **Download objects is on by default for added locations**
  This option controls whether the 'download objects' setting is on by default in the add location dialog.

- **Connect to host list individually**
  When this option is checked MO71 will connect to each host in a connection name individually. This is done so that MO71 can know which server is hosting the Queue Manager. By adding a text string after the host address MO71 will display the currently hosting server on the screen.
- **Save resolved hostnames**
  When this option is checked MO71 will save any resolved host names to a file called **mqmon.dnc**. This means that MO71 can save the cached host names across restarts. For more information please see “Hostname Support“ on page 24.
- **Check whether Queue Manager command level exceeds the level supported by this version of MO71**
  By default MO71 will check, and warn you, if you connect to a Queue Manager whose command level exceeds that supported by this version of MO71. You can continue to administer such Queue Managers but be aware that some objects and attributes will not be displayed because MO71 doesn't know about them.
- **Location 'Log Console' includes successful QM monitor**
  If you are logging the Queue Manager monitoring to the console then this option controls whether successful monitors (rather than just failures) are logged to the console. By logging successes you will be effectively removing logging of any failures.
- **Auto Connect Delay**
  This parameter sets the frequency, in seconds, an auto connect location will try to connect to its target Queue Manager. Please see ‘Time Interval Format’ on Page 232 for the format.
- **Command Expire Interval**
  This parameter sets the expiry interval of the command messages sent by the MO71 program. Expiring messages is useful to prevent a build-up of messages on, for example, a transmission queue which can’t connect to a remote location. Please see ‘Time Interval Format’ on Page 232 for the format.
- **DNS Expire Interval**
  This interval controls how long MO71 will cache a host name, retrieved by *gethostbyaddr()*, before issuing the call again.
  For more information please see “Hostname Support“ on page 24.
- **DNS Suffixes**
  This field can contain a comma separated list of DNS suffixes, for example *.mqgem.com*, which will be removed from any resolve host names.
  For more information please see “Hostname Support“ on page 24.
- **Reload DNS File**
  This button will cause MO71 to reload the user DNS host name definitions from **mqmon.dns**.
  For more information please see “Hostname Support“ on page 24.
- **Clear DNS Cache**
  This button will cause all the System Host name values to 'expire'. This means that the next time they are accessed a new value will be requested from the DNS server using *gethostbyaddr()*.
  For more information please see “Hostname Support“ on page 24.
- **Password Cache File (**various fields)
  This box contains various fields concerned with stored a cache of userids and password. Please refer to “Password Cache File“ on page 222 for the meaning of these fields.

24.2.6. Network

- **Resolve Queue Paths**
  In the Network Display the application will try to resolve the paths for all queues in the Queue Manager definition. For large numbers of queues and large numbers of Queue Managers this may take some time. If this information is not needed then this value may be unchecked.
- **Resolve SYSTEM Objects**
  Normally the objects starting with the string ‘SYSTEM.’ are only used as default definitions and are therefore not used as real objects by applications. As such you would normally not want the SYSTEM objects resolved.
- **Check for Problems**
  The Network View display can display possible problems with the current configuration of Queue Manager, for example, an alias queue that targets an undefined queue. If this information is not required then this option may be unchecked to improve performance.
- **Auto-Analyse Network**
  Normally checked, this tells MO71 to automatically re-analyse the relationships between the Queue Manager objects in the Network view. They may be some occasions though where the user always wishes to manually request analysis and therefore this option may be unchecked.

- **Wildcard Search**
  This option controls whether automatic '*' wildcards are added to the Network view search string. When selected an entered string of 'abc' is actually treated as '*abc*'.
- **Case Sensitive Search**
  This option controls whether the Network view search is treated in a case sensitive or case insensitive manner.
- **Allow Queue Manager discovery**
  This option controls whether MO71 will look for unrecognised Queue Managers in channel status records or monitoring responses. If discovery is on then a foreign location will be automatically defined.
- **Display problem priority icon**
  Each problem is given a priority based on how serious the situation is considered to be. Red is high priority and should really be fixed. Yellow implies it may be problem depending on the circumstances. Green implies that this is not really a problem but may be something you should consider changing. Each problem type can be switched off in the problem selection window. This preference only controls what is shown in the verify dialog – the Problem Selection dialog will always show the icon.
- **Show network Queue Manager selection as red crosses**
  If this option is checked then any unselected Queue Managers will show as a red cross. If this option is not selected then any unselected Queue Managers will be shown as a blank square.
- **Frame includes icon**
  Controls whether the frame around the Network display Queue Manager includes the icon or not.
- **Use Digram Frame Colour**
  Controls whether the diagram frame colour in the colour scheme is used.
- **Use Diagram Focus Colour**
  Controls whether the diagram focus colour in the colour scheme is used.
- **Fixed distance links**
  The lines connecting the Queue Managers, representing the channels, in the Network display stop short of the actual Queue Manager. This option controls whether they stop short a fixed distance away or whether they reach the frame or icon.
- **Display different z/OS icons**
  Controls whether the icon of a mainframe is automatically shown for an z/OS locations or whether the standard machine icon is used.
- **Network Snap**
  The network display allows the user to move individual Queue Manager icons. The movement can 'snap' to certain locations :-
  - **None** no snap is performed
  - **Qmgr** the icon is snapped to other Queue Managers
  - **Grid** the icon is snapped to a fixed size grid
- **Network Snap Size**
  The option allows the user to set the size of snap grid in pixels.
- **Auto Channel Status Update**
  The application may be configured to periodically query the running state of the channels. Which Queue Managers are queried is set by this value
  - **Network (only connected)** Only Network Queue Managers already connected to
  - **Network** All Network Queue Managers
  - **All connected** All Queue Managers currently connected to
  - **All** All Queue Managers
  - **None** Don't perform automatic channel status update
- **Auto Channel Status Update Rate**
  Sets the interval, in seconds, for channel status update. Please see 'Time Interval Format' on Page 232 for the format.
- **Location Frame**
  Controls what type of border is used around the frame when the Queue Manager is not in focus
- **Location Frame Focus**
  Controls what type of border is used around the frame when the Queue Manager is in focus
- **Network Border**
  Controls how much border space is left around the Queue Managers in the Network display when scaled.
- **Object Refresh Rate**
  This interval controls how often MO71 should try to automatically refresh the objects in the verify window. Please see 'Time Interval Format' on Page 232 for the format.

- **Object overdue interval**
  This interval sets the point at which definitions in the verify window are considered stale. If the set of object definitions is overdue it is identified by a red icon. Overdue Please see 'Time Interval Format' on Page 232 for the format.
- **Object retry rate**
  This sets how often an attempt should be made to automatically refresh the objects if the initial attempt fails. Please see 'Time Interval Format' on Page 232 for the format.
- **Search update delay**
  This option sets the delay between data being entered in to the search field and the search being applied to verify data. The interval is specified in milliseconds. Please see 'Time Interval Format' on Page 232 for the format.

24.2.7. Network UI

24.2.7.1. Icons

These fields allow you to set the default bitmaps which should be displayed in the Network display for the locations. Note that these are only the default bitmaps, they can be specified per location if required. Essentially any size, within reason, can be provided. The bitmaps can be created using a variety of tools including the standard Windows paint program. It is not necessary for the different state bitmaps to be the same size. For example you may decide that a 'good' ie. online Queue Manager should be twice the size of a 'bad' offline Queue Manager.

Please see A.7. User Icons on page 269 for more information about user icons.

24.2.7.2. Media Titles

The Verify Display in the Network View will show the Storage Classes, Pagesets and Buffer Pools defined on a z/OS Queue Manager. These objects, together with local queues, form a hierarchy. Queues live in a particular Storage Class, Storage Classes live in a particular Pageset. Pagesets in turn are associated with a particular Buffer Pool. If can be useful to know, for example, how many Queues are associated with a particular Buffer Pool or Pageset.

For this reason the name of the object, displayed in Verify Display can be augmented by adding a title using the fields here. The titles can, and probably should[42], contains inserts. The inserts allow you to show information not only about the object itself but also about the other objects in the hierarchy. For example, it might be useful to see what percentage of Queues are defined using each Storage Class. Or perhaps what are the number of Buffers assigned to a Pageset.

The table of inserts are the following:

Type	Description
%B	Total number of Bufferpools
%P	Total number of Pagesets
%S	Total number of Storage Classes
%Q	Total number of Local Queues
%X	Total number of Pages
%F	Total number of Buffers
%b	Number of involved bufferpools
%p	Number of involved pagesets
%s	Number of involved storage classes
%t	Number of storage classes used by pageset
%u	Number of storage classes used by the bufferpool
%q	Number of involved queues
%x	Number of involved pages
%f	Number of involved buffers

[42]Since without an insert this is just a fixed string, the same on all objects.

%d	Description (Storage Class only)
%=b	Percentage of involved Bufferpools
%=p	Percentage of involved Pagesets
%=s	Percentage of involved Storage Classes
%=t	Percentage of involved Pageset Storage Classes
%=u	Percentage of involved Bufferpool Storage Classes
%=q	Percentage of involved Queues
%=x	Percentage of involved Pages
%=f	Percentage of involved Buffers
%%	Percentage sign

A single Media Title can contain many inserts. Not only that but you can make the data more expressive by using a simplistic markup based loosely on BB codes. For example, you can specify **Queues:[i]%q[/i]** and the number of Queues will be shown in italics. For the complete list of codes which can be used please see Appendix C: BB Codes on page 274.

24.2.8. Applications

- **Show Applications**
  This switch is the main on/off switch for applications. If you decide that you are not interested in the state of your applications then you can switch it off and applications will not appear in the main window or verify displays.
- **Monitor Applications**
  This switch is the main on/off switch for monitoring applications. If unchecked then MO71 will not automatically inquire the state of any applications. Instead they must be requested manually.
- **Application error reflected in QM icon**
  This option controls whether the Queue Manager icon should change to the error icon when an application is marked as in error. An application is considered in error if the number of connections it has is outside it's configured limits.
- **Application error reflected on console**
  This option controls whether an application error should cause an entry to be written to the console. An application is considered in error if the number of connections it has is outside it's configured limits.
- **Application error sounds alarm**
  This option controls whether an application error should cause the alarm sound. An application is considered in error if the number of connections it has is outside it's configured limits.
- **Frame includes icon**
  Controls whether the frame around the Application view Application includes the icon or not.
- **Use Diagram Frame Colour**
  Controls whether the diagram frame colour in the colour scheme is used.
- **Use Diagram Focus Colour**
  Controls whether the diagram focus colour in the colour scheme is used.
- **Snap**
  The application view display allows the user to move the application and queue icons. The movement can ‘snap’ to certain locations :-
  - **None** no snap is performed
  - **Grid** the icon is snapped to a fixed size grid
  - **Object** the icon is snapped to other objects in the display
- **Snap Size**
  The option allows the user to set the size of snap grid in pixels.
- **Frame**
  Controls what type of border is used around the frame when an object is not in focus

- **Location Frame Focus**
  Controls what type of border is used around the frame when the object is in focus
- **Border**
  Controls how much border space is left around the objects in the Application view display.
- **Local Refresh Delay**
  This interval controls how often MO71 should try to automatically refresh the application usage data for locally bound locations. Please see 'Time Interval Format' on Page 232 for the format.
- **Client Refresh Delay**
  This interval controls how often MO71 should try to automatically refresh the application usage data for client connected locations. Please see 'Time Interval Format' on Page 232 for the format.

24.2.9. Application Icon

The fields allow you to set the default bitmaps which should be displayed in the Application View display for the applications. Note that these are only the default bitmaps, they can be specified per application if required. Essentially any size, within reason, can be provided. The bitmaps can be created using a variety of tools including the standard Windows paint program. It is not necessary for the different state bitmaps to be the same size. For example you may decide that an active application should be larger than an inactive application.

Please see A.7. User Icons on page 269 for more information about user icons.

24.2.10. Export

- **Export Text Title**
  Controls whether a title line is written preceding any exported data when exported in Text format.
- **Export MQSC Title**
  Controls whether a title line is written preceding any exported data when exported in MQSC format.
- **Add CSV sep= meta line to CSV export**
  Controls whether a meta line is added to the start of the CSV export which states what character is being used as the separator character.
- **Text Title**
  This field allows you to set the format of the title line exported with any text exports. The field can contain inserts please see see Text Inserts on page 227 for a description of these inserts.
- **Text Sample**
  This field shows an example of how the title line or lines would look.
- **MQSC Title**
  This field allows you to set the format of the title line exported with any MQSC exports. The field can contain inserts please see see Text Inserts on page 227 for a description of these inserts.
- **MQSC Sample**
  This field shows an example of how the title line or lines would look.

24.2.11. Naming

This tab control aspects of naming that affect all Queue Manager, referred to as global naming. For more information Chapter 8. Naming Objects on page 74.

- **Only warn when defining a badly named object**
  By default MO71 will prevent an object being defined with a name which does not match the name patterns. However, by checking this option you can request that the user is merely shown a warning dialog. If the user wishes they can continue to define the object.
- **Check SYSTEM object names**
  By default MO71 assumes that SYSTEM objects are not included in the name patterns, either global or local. As a consequence SYSTEM objects are exempt from the name checks. However, by checking this option you tell MO71 to check the SYSTEM objects as well. Clearly if the pattern does not actually include a pattern which includes the SYSTEM object then this is an effective way of preventing the accidental creation of any SYSTEM objects.
- **Object type tabs – Queues, Xmit Queues, MCA Channels, MQI Channels etc**
  The fields allow you to specify the global object name patterns for the various types of object. For a more details description please see Global Object Name Pattern Specification on page 76.

24.2.12. Sounds

- **Warning Sound**
  The sound played when the application wishes to issue a warning. For example an invalid value has been entered or a command failed.
  The value can be beep, none, or .WAV file.
  Depending on the OS it may be necessary to enter the full path to the .WAV file
  The application will play the sound when the user double-clicks on the entry field or presses the 'Play' button.
- **Alarm Sound**
  The sound played for the application alarm.
  The value can be as above for the warning sound.
- **Sound Times**
  Allows the specification of how long and how often the sounds should be generated.
  The format of the field is **X [: Y [: Z]]** where :-
  - **X Minimum alarm interval**
    Minimum time which must have elapsed before sounding the alarm again
  - **Y Maximum alarm period**
    Maximum time to sound the alarm for
  - **Z Minimum sound interval**
    Minimum time which must have elapsed between making any sound

24.2.13. Display

- **Display menu icons**
  When checked some of the menu items will have simple bitmaps drawn next to them to aid in visual recognition. Changing this option may only have an effect the next time the menu is created. It may be necessary to restart the program to see this option change reflected in all menus.
- **Alter main icon when QM failure**
  When checked the main application icon will go red if any of the Queue Managers are in error.
- **Track selection in container view**
  By default the application high lights the Queue Manager entry as the mouse is moved over the main window. By checking this option this behaviour can be turned off.
- **Use DOS font**
  There are slight differences between the Windows and DOS versions of the same codepage. The codepage itself is identified later in the preferences dialog in the locale field. This preference option just controls the characters displayed when browsing messages on queues. The right value is probably best found by trial and error.
- **Flash main window on alarm**
  Check this option for a visible indication of when the alarm is sounding.
- **Ticks should be green**
  When selected, tick symbols (sometimes known as check marks) are displayed as green. When not selected the symbol will be displayed as red. Remember than until you press ‘OK’ or ‘Apply’ no change will take place.
- **Show only ticks, no crosses**
  In various lists MO71 will show whether an option is selected by displaying ticks and crosses. Using this option you can choose that crosses are not shown – it can make the display look 'cleaner'.
- **Tabbed**
  If checked then some of the dialogs, for example Queue Manager, Queue and Channel, will be displayed using Tabs to group together relate fields in categories. These dialogs have large numbers of fields and using Tabs can make finding a particular field much easier. If you un-check this setting then dialogs are shown as a just a (long) list of fields.
- **Multi-line**
  If Tabs are switched on then this setting controls whether the Tabs should be displayed across multiple lines or as a scrollable list.
- **Bottom**
  For horizontal Tabs this setting controls whether the Tab is drawn at the top or at the bottom.
- **Buttons**
  This setting controls whether the Tabs are actually shown as buttons.

- **Flat**
  If the Tabs are displayed as buttons this setting sets whether they should be displayed as normal buttons or as flat buttons.
- **Vertical**
  This setting controls whether the Tabs are drawn horizontally or vertically.
- **Right**
  For vertical buttons this setting controls whether the Tabs are drawn to the left or the right.
- **System Draw**
  Tabbed controls in Windows are quite odd little controls and are, quite frankly, riddled with problems. One of the main problematic areas is changing their appearance. A simple search on the internet will find dozens of confused programmers tearing their hair out wondering why the control is not behaving. In the end, some of them do as I have done and take on the responsibility of drawing the Tab themselves. So, by default, MO71 draws all the Tabs in the dialogs itself. That way the Tab can be drawn 'correctly' and using the configured colours. This has the advantage that the Tab fits in with the colour scheme. The disadvantage of this approach though is that the Tabs may not look like other Tabs in the Windows system. So, this setting allows the user to choose whether they prefer the command dialog Tabs drawn by Windows itself which don't conform the the colour scheme or Tabs drawn by MO71 which may not look exactly like other Tabs on the system.
- **Toolbar Raised,Highlight,Highlight Border**
  Selects the style that tools are displayed on the toolbar.
- **Below Edge**
  Determines the type of edge which should be used on the bottom of the toolbar. Note that you can deselect all.
- **Above Edge**
  Determines the type of edge which should be used on the top of the toolbar. Note that you can deselect all.
- **Locale**
  Any valid Windows locale string can be entered in this field. The value specified will control the characters displayed in dialogs such queue browsing. It will also control which characters are considered as printable. A value of 'Default' returns the application to the default windows setting.
  The drop down list contains the basic settings for this field. Selecting one and pressing apply will then display the locale selected by windows. The number at the end of the locale is the codepage. This can be changed if required provided the entire string is recognised by windows as a valid locale string.
  For more information see *"Appendix E: Display National Language Characters"* on page 276.
- **Browse CCSID**
  This field allows you to specify a codepage to convert messages to when browsing messages on queues. In general if you change the value of 'locale' you would want to change this CCSID to match the codepage number. A value of 'Default' will cause the application to use the standard Windows codepage setting.
  For more information see *"Appendix E: Display National Language Characters"* on page 276.
- **CSV Separator**
  The character to use for comma separated lists generated by the export command. An option in the Export tab controls whether a meta line is added to start of an export file which says what separator character is being used.

24.2.14. Events

- **Events can update queue lists**
  If the user has defined event listeners for queue events this preference option controls whether the events should be used to drive a refresh of Queue List displays. The events that drive the refresh process are :-
  - **Queue Depth High**
  - **Queue Depth Low**
  - **Queue Full**
- **Events can update channel status lists**
  If the user has defined event listeners for channel events this preference option controls whether the events should be used to drive a refresh of Channel status List displays. The events that drive the refresh process are :-
  - **Channel Activated**
  - **Channel Not Activated**
  - **Channel Started**
  - **Channel Stopped**
  - **Channel Stopped by User**

- **Reconnect Interval**
  If an event has been defined and started but cannot connect to the Queue Manager it will attempt a re-connect. This value allows the user to set the retry frequency in seconds. Please see 'Time Interval Format' on Page 232 for the format.
- **Event Averaging Interval**
  As each event message is caught a smoothed average arrival frequency/ per interval is kept. This value allows the user to set the interval in seconds. A large interval allows a smoothed average over a large amount of time; a small value will only average over a small period. Please see 'Time Interval Format' on Page 232 for the format.

24.2.15. Console

- **Maximum items**
  This value controls how many items are maintained in the console. If the console reaches this limit the items in the console will be removed in priority order - lowest first. Within each priority, items will be discarded oldest first.

24.2.16. Units

Many of the MQ attributes have an associated unit such as Seconds or Bytes. This preferences dialog tab controls when MO71 displays these units. For more information about Units please see Chapter 11.2 Units on page 99.

- **Units Enabled**
  This is the main on/off switch for units. If not enabled then MO71 will not display any unit information.
- **Units Entry Field**
  This setting controls whether units are displayed in the entry field on an object dialog.
- **Units Style**
  These selections control how the unit information is displayed.
- **Unit Locations**
  The following sections control in which locations MO71 should use the units and how the units themselves should be represented.

24.2.17. Time

- **Time Zone**
  The time-zone which should be used for the MO71 application itself. You can specify different time-zones for each location in the location dialog.
- **Show local times**
  When checked the values for times, such as put time in a message browse and MQ object times will be displayed in local time rather than their original time value.
- **Use System settings**
  When checked the value for the local time-zone will be taken from the windows system. This value must be unchecked to be able to specify your own time zone.
- **Date Separator**
  This field selects what is the preferred separator character for dates. This is used in the subsequent date fields as well as the date format used in exports.
- **Date Format**
  This field selects the preferred simple date format. This is used in exports as well as made available as the %DP insert in the Formatted time fields.
- **Date Example**
  This field shows what the preferred date format would look like.

- **Dialog Refresh Time (Large Format, Medium Format, Small Format)**
  These values control what is displayed on the right hand side of the dialog titlebar to display the last refresh time for the dialog[43]. Any character string may be entered with the addition of certain inserts preceded by a '%' character. The inserts characters are :-

Insert	Meaning
H	Two digit hour
M	Two digit minutes
S	Two digit seconds
d	Two digit day of month
m	Three character month name eg.Jan,Feb,Mar….
y	Four digit year
D	Three character day of week eg.Mon,Tue,Wed…
t	Simple time format eg. 18:14:03
%	Percent character

  A preference option controls whether the refresh time is shown on the left or the right of the titlebar. If the time is shown on the left then only the small time format is used.

- **Time Format**
  This field controls what the formatted time field looks like. This is the format of the Date/Time fields in the MQ objects as well as the Date/Time field of a message browse. As changes are made to this field the Time Sample field will change to show the result. This field can use any of the time inserts described in Chapter 24.7 Text Inserts on page 227.

  A number of suggested formats are provided in the drop-down list. Entering a null string value will result in the default being applied.

- **Time Sample**
  This field shows an example of the current formatted time format. As changes are made to the field above so they are reflected here. This field may not be edited but you can select a format from the drop-down list. This will change the Time Format field accordingly.

- **Today/Yesterday or Last 24h/48h**
  The User can optionally provide a different date format for recent dates. This field selection allows you to choose the meaning of 'recent'. Either it means in the last 24 or 48 hours. Or it can mean 'Today' and 'Yesterday'.

- **Time Format and Time Sample**
  Two mode format and sample time formats which allow you to specify the format which should be used for 'recent' dates.

- **Monitor Time Format**
  This setting controls what is displayed on the main window if 'Show Monitor Time' is selected. The text can contain any text including inserts as described in chapter 24.7 Text Inserts described on page 227.

24.2.18. HTTP

- **Auto start HTTP Server**
  When checked the HTTP listener (using either HTTP or HTTPS depending on other configuration) will be started as soon as the program starts.

- **Below root only**
  When checked the program will limit file requests to those below the HTTP root path

- **Send XML by default**
  MO71 is capable of sending either HTML or XML as the result of a query. If the HTTP requesting application doesn't specify which of these it wants then this setting decides which format the data should be returned in.

- **Show HTTP Dialogs (Debug)**
  When checked this option causes the dialog windows used to serve the HTML pages to be visible on the machine running MO71. Normally you would expect this value to be unchecked. It can be useful, though, to determine what dialogs and values are set when processing an inbound URL.

- **Allow Mini-dump Generation**
  You should normally leave this unchecked unless you have a problem with web access. Please see the Mini-Dump section of this manual for a description of mini-dumps.

[43]This will not be displayed if the desktop is configured to use a Windows Aero 'glass' theme. This is because the Windows OS prevents the application painting to the non-client area of a window.

- **Synchronous Response**
  If this is checked then pages will always be served with new data. If it is unchecked then MO71 can serve up data from its cache. This can have a significant effect on performance. For example, consider a web page showing data from 20 different Queue Managers refreshing every 30 seconds. If synchronous response is set then each page refresh will wait for the responses of data from each of the 20 Queue Managers. This could introduce a significant delay. If it is unchecked then MO71 will request new data but it will not wait for the response if it already has some data in its cache. Consequently the web page may show data which is 30 seconds old but will respond much more quickly.
- **First Response**
  When checked MO71 will respond as soon as it has 'something' to send. If the web page is for multiple Queue Managers then there is no attempt to wait for all of the data to be returned.
- **Wait for non-responding Queue Managers**
  If a web page consists of data from many different Queue Managers it is possible that one or more of those Queue Managers are either not responding or a very slow. If this option is not checked MO71 will not wait for these non-responding Queue Managers before returning the data for the web page. If this item is checked then the refresh of the web page could well take the entire 'Max Response Interval' as MO71 waits for the data to be returned from these bad Queue Managers.

24.2.18.1. General

- **Root Path**
  The root directory for HTML pages
- **Dump Code**
  This field is only relavent if you enable mini-dump generation over the web. Please see the Mini-Dump section of this manual for a description of mini-dumps.
- **Enable HTTP**
  If this option is selected, the listener will run using the HTTP protocol.
- **Enable HTTPS**
  If this option is selected, the listener will run using the HTTPS protocol.
- **HTTP Port**
  The port number to listen on when using the HTTP protocol.
- **HTTPS Port**
  The port number to listen on when using the HTTPS protocol.
- **HTTPS Certificate Location**
  This group of fields allows you to specify where the certificate used to identify the HTTPS MO71 web server resides and how to identify exactly which certificate to use. For full details about these fields see 20.5. Using HTTPS on page 163.
- **Request Interval**
  If MO71 has lots of browsers issuing the same query it would be inefficient for it to issue the refreshes to the Queue Managers every time. For example consider 100 browsers all querying a list of queues within the same few seconds. You probably don't want 100 requests to the server for an entire list of queues. This parameter controls the minimum time between requests for the same data. So, if a browser ask for a list of queues within 10 seconds of another browser request then it will just be sent the same data as the original request without requesting more data from the command server.
- **Max Response Int. (Interval)**
  This is the maximum amount of time a browser request should wait for data from the Queue Managers. After this time MO71 will send whatever data it has managed to collate by this point, which may be nothing.
- **Retain Int. (Interval)**
  When MO71 requests data from the command server it will cache the dialog and data for subsequent requests. This parameter controls how long MO71 should maintain this infrastructure without any active browser clients. A large value will increase efficiency at the cost of more memory. A low value will incur more CPU cost and greater latency of browser requests.
- **Max Field Width, Max Column Width**
  When MO71 sends an object display these fields controls the maximum amount of field data to display. This is useful because some MQ fields can get quite big which can make the display hard to digest as well as involving sending much more data to the browser. A value of 0 implies no limit should be applied.
- **Incl. Filetype**
  The list of file types which may be served. If blank then there are no restrictions.
- **Excl. Filetype**
  The list of file types which are excluded from being served. If blank then there are no restrictions.

- **Connections**
  The current number of connections.
- **Peak Connections**
  The peak number of connections.

24.2.18.2. IP Address List

- **Allow Address**
  Here you can specify an address or address range which is acted on by the 'Add' and 'Check' buttons.
- **Add**
  The specified 'Allow Address' will be added to the 'Allow List'.
- **Check**
  This button will check whether the address specified in the 'Allow Address' is currently allowed to connect.
- **Delete**
  This button will delete the currently selected entry from the 'Allow List'.
- **Allow List**
  This list shows the list of IP addresses which are allowed to connect to the MO71 web server. If the list is empty then all devices are allowed to connect. For more information please see “Limiting connections“ on page 165.

24.2.18.3. Proxy Settings

- **Host**
  The host name or IP address of an HTTP Proxy. An HTTP Proxy is used to provide extra security when making HTTP connections by providing a session break and greater machine anonymity when accessing the internet. This feature is used when MO71 accesses the internet to determine the latest MO71 and IBM MQ versions.
  A secure HTTP Proxy, which requires a userid and password, is not supported.
- **Port**
  The port number to use for the HTTP Proxy. If not specified then 80 will be used.

24.2.18.4. HTTP Server buttons

- **Start**
  A button which will start the HTTP or HTTPS listener using the above configured settings. If the port number or choice of protocol (HTTP vs HTTPS) has changed from the currently running listener, this will cause the current listener to terminate before starting a new listener.
- **Stop**
  A button which will stop MO71 accepting new connections but any existing connections will be maintained.
- **Close**
  A button which will terminate the HTTP listener and any existing connections.

24.2.19. Graph

The graphing tab allows you to specify the way in which, in general, you would like graphs to be displayed. Note that the preferences are not always adhered to. For example, if you save a graph then the graph is restored to the way it was saved, not the current preference settings.

Most of the settings in the preference tab are copies of the values you would find in the Graph Options Dialog, which is described on page 115, so it not worth repeating this here. However, one of two fields are worthy of note.

- **Line Colour and Symbol**
  These settings allow you specify your preference for the first eight lines of any graph. The symbols are only relevant if you choose to display the graph as a line. If, for example, you chose a column of some sort then the symbols are not displayed. The other point to make is that, of course, a graph could contain many more lines that eight. If that is the case then MO71 will construct new colours by combining the colours you have chosen.
- **Scheme – Light and Dark**
  As a convenience MO71 has a couple of preset colour schemes. These can be ignored, used 'as-is' or used as the starting point for your own colours. If you generate a new set of colours that you believe are particularly appealing then by all means let us what they are and perhaps we will include your new scheme in a future version.

- **Set All**
  Note that if you change the default values they will, by default, only affect graphs at creation time. However, if you want the values to take effect immediately then you can use the 'Set All' button to apply them to all graphs. If you only want one graph to pick up the new defaults then you can use the 'Reset to Graph Defaults' on that graph's Graph Options Dialog. Note that the graph line type is not adopted in this fashion.
- **Sample Graph**
  As a convenient way of seeing what a graph might look like and playing around with the options you can use this button to show a set of graph lines. The data shown is entirely random and serves no other purpose than giving some data to play with.

24.2.20. Help

- **Help file command**
  This field is provided when help is provided by the PDF file. If can be left blank if you machine has an association that knows what program to run for PDF files. If no association exists then enter the program that should be used to read the PDF file.
- **Help file location**
  Help can be provided to the user either in the form of a PDF file of a Web Page (URL). By default MO71 will try to display this manual as help but if you wish you can provide your own PDF file or a Web location.

  Enter the location that you want MO71 to use. This can take any of the following formats:

  - c:\mo71\mqmon.pdf
  - http://www.acme.help/mo71.html
  - d:\mo71\myhelp.html

  The help is displayed when the user presses ‘F1’ or selects *File-Help* from the main window.

24.3. Location Dialog

Similar dialogs are displayed when the user chooses to Add or Open a location. A description of the fields is given below. Note that the Add dialog does not contain all the fields given.

- **Location**
  A free format text description of the location.
  This field is to allow the user to assign a more meaningful name.
- **Queue Manager**
  The Queue Manager name that should be used by MO71 when sending messages to the remote location.
- **Logical Name**
  This checkbox should be selected if the Queue Manager name is merely a logical name and not the actual name of the Queue Manager. This is useful when using client connections; it has the effect that a '*' is pre-pended to the Queue Manager name when connection is made[44].
- **Foreign**
  Checkbox which indicates whether this is a 'foreign' location. A foreign location is a read-only Queue Manager which is known to exist in the network somewhere but outside the scope of this MO71 configuration. Usually foreign locations are added automatically when MO71 discovers a location via a returned channel status or event message. However, it is possible to manually create them if required. Similarly a foreign location can be changed to a non-foreign location by unchecking this checkbox. If the user does this then normally they would also add connection information to allow MO71 to connect to the Queue Manager.

24.3.1. Connection

- **QM Group**
  A free format group name for this location
  The effect of this value is that the location will be displayed in the tree view 'under' the group name. This allows all your z/OS machines, for example, to be contained in a group such as "z/OS Queue Managers".
  Since objects are placed in the tree view in alphabetic order you can place all of your groups at the top of the main window by having a group name with a leading blank.
- **Network Names**
  You can give a comma or space separated list of Network names with which you want this location to be associated with. This allows groups of Queue Managers to be associated with particular network name. In the network view you can then include or exclude groups of Queue Managers associated with a particular name. See Network Names on page 88 for more information.
- **Via QM**
  If the administrator should not try to connect directly to the Queue Manager name given above but instead should send the messages via a different Queue Manager then this field allows the user to tell the application which Queue Manager it should use.
- **Reply Queue**
  Only required if this location is connected to directly i.e. no Via QM is specified.
  This is the name of the predefined reply queue on the above Queue Manager. It defaults to the name SYSTEM.DEFAULT.MODEL.QUEUE. See "Reply Queue Details" on page 229 to see how another name can be used.
- **Reply Prefix**
  If a model queue is given in the Reply Queue field then the Reply Prefix field can be used to specify the prefix which should be use to construct the actual temporary queue name. See "Model Reply Queue" on page 229 to see how another name can be used.
- **Command Queue**
  The name of the queue at the location which processes the MQ commands. This value is set automatically to the appropriate name depending on whether the MVS check box is selected or not. It can be changed to another queue if required.
- **Server Queue**
  The name of the queue at the remote machine where MO71 command messages should be sent. It is not required if this location directly connects to the queue manager. But if you want to browse a queue at a remote location which you are not directly connected to, then you must have a program, for example MO71, at that location to service those requests. This is the name of the queue which that program is reading.

[44] For those interested look at the section on Queue Manager groups in the MQ clients book for a description of this behaviour.

- **Client**
  If checked this indicates that MO71 should connect to this location via a client connection rather than directly.
- **Configure(d)**
  Only available if 'Client' is checked
  If the location is to be connected to via a client then the client definition to be used can be entered in a dialog presented when this button is pressed. If a separate client channel definition is not entered then the client connection will use the normal MQSERVER or Client Channel mechanisms to connect to the server.
  If the target location is a multi-instance Queue Manager then the connection list field should be a comma separated list of the IP addresses of the locations where the Queue Manager can be located.
  The name of this button will either be 'Configure' or 'Configured' depending on whether there is currently a client configuration specified.
- **MVS**
  Select this if the remote location is z/OS. This will cause the name of the command queue to change. When checked, messages will be sent to the command queue in MQSC, rather than PCF format.
- **Auto Connect**
  When checked the application will always try to ensure that it is connected to the Queue Manager. If the button is unchecked then connection to the Queue Manager will be made 'when necessary'. In other words the connection attempt will only be made when a monitor message or command is issued against this Queue Manager.
- **CCDT URL**
  This field can be used to specify the location of the CCDT file to use. The field is only enabled when the 'Client' check box is checked and there is no local client definition specified.
- **Retry Interval**
  If the connection to the remote queue manager fails for some reason (for example, the Queue Manager is ended) then this field allows the specification of a number of seconds before a reconnect attempt is made. No automatic reconnection attempt will be made if the value is 0.
- **Disconnect Interval**
  This field allows the specification of the number of seconds of inactivity before the administrator disconnects from this location. A value of 0 means that the administrator will never disconnect.

24.3.2. Security

The security tab of the location dialog contains fields which are primarily concerned with the security of the connection. However, these are by no means the only fields associated with a location which are related to security. For example, the channel definition contains a number of fields concerned with security such as the SSL/TLS values which affect encryption levels and authentication across the client link.

- **Passwords**
  - **Userid and Password**
    Check this item if you wish the program to pass a userid/password combination over on the connection. This option requires at least MQ Version 6 to be installed on both the client and the server. The program will remember the userid across invocations of the program. The password will only be remembered across restarts if a password cache file is configured.

    The userid and password is always cached for the life of the program so that the user is only prompted to enter the details as infrequently as possible.
  - **Upper Case Userid / Password**
    Check one or both of these items if you wish the Userid and/or Password fields to be automatically folded to upper case before being sent to the queue manager. This can be especially useful for z/OS TSO logins where it can be easy to forget that you have an upper case password because the TSO login panel always folds it to upper case for you.
  - **Security Exit Only**
    This field only applies if the 'UserId' checkbox is checked. This option controls whether the userid is passed solely to the security exit or whether it is also passed to the Object Authority Manager (OAM) component of the Queue Manager. Essentially this option controls whether the MQCSP_AUTH_USER_ID_AND_PWD option is specified when MO71 connects to the Queue Manager.
  - **Cache on Success Only**
    This field only applies if the 'UserId' checkbox is checked. This is to avoid the user having to specify the values multiple times, each time MO71 makes a connection. This option controls how MO71 caches any entered

Userid/Password. If selected then the userid will only be cached if the connection is successful. If unchecked then the userid will be be cached unless the connection attempt fails with either MQRC_NOT_AUTHORIZED or MQRC_CONNECTION_NOT_AUTHORIZED. If you are using a Channel Security exit to validate your userid/password then you probably want to check this option.

- **Store in cache file**
  If checked then MO71 will retrieve and store any entered userid and password for this location to the defined password cache file if it is defined. The password cache file is set in the Connection tab of the Preferences Dialog.
- **Security Group**
  The idea of a Security Group is that a number of Queue Managers may be secured using the same mechanism and therefore it useful if they can share the same userid/password.
  - **Name**
    All locations with the same Security Group Name will be considered part of the same group. The actual name is entirely arbitrary and need not have any bearing on other security values/domains. It is just a name which MO71 will use to group locations together. Changing the userid/password for one member of the group will change it for all members. If the password cache is being used then just one entry for the entire group is made.
  - **Description**
    Having a description for your Security Group in entirely optional. If specified then the user will be presented with this description when asked for a userid/password so it can be useful as a memory jogger.
- **Set Password**
  Pressing this button will allow the user to enter a new userid/password combination for this location. The values will be cached if configured to do so.
- **Delete Password**
  Pressing this button will remove any record of the userid and password from the cache.

- **Transport Layer Security (TLS)**
  - **Allow Weak Cipher Specs**
    A number of Cipher Specs (those using the SSL V3.0 protocol, and those using no encryption, or weak encryption algorithms) are no longer recommended, since they don't really provide sufficient protection for sensitive data. As a consequence by default they will not be presented as choices in new channel definitions. However, if you have a need to define channels using these old Cipher Specs then you can select this option. Note that, depending on the version of MQ you are defining the channel on, you may need to also configure MQ to accept these weak Cipher Specs. For further information please see https://mqgem.wordpress.com/2015/07/21/deprecated-cipherspecs/.
  - **Forget KDB Password**
    If you have provided MO71 with the password for a Key Database file which has subsequently changed, you can cause MO71 to forget any current knowledge of that KDB file and prompt you for a new password next time it is needed, by pressing this button. The same KDB file (and thus password) is most likely being used for many queue manager connections in MO71. Pressing this button forgets the password for all those queue manager connections as it stored only once for all to use.

24.3.3. Time

- **Time Zone**
  Specify the time-zone for this location. The time-zone is used to interpret any dates and times issued by the Queue Manger.
- **Time-zone different**
  If this checkbox is not selected then it will be assumed that this location is in the same time-zone as the MO71-wide time-zone specified in the Preferences Dialog. However, by selecting this checkbox a different time-zone can be defined for each location. Care should be taken to select the right time-zone when there are multiple choices with the same time bias. The difference between the entries is generally the behaviour of Daylight Savings Time (DST). Different locations either don't support DST or have their DST's start and end on different dates.

24.3.4. Options

- **Disable Commands**
  When checked the administrator will not send commands to that particular location. The administrator will 'beep' if the user attempts to issue a command. Monitoring to that location is still permitted. This option is useful if the remote location does not have a command server and generating commands would result in messages being put to the Dead Letter Queue or worse causing the channel to end.

- **Disable Auto Updates**
  Auto updating lists can be a useful convenience. However, refreshing lists on some locations can be expensive, for example a client connection over a slow link. In these cases this option can be used to switch it off just for a particular location.
- **Single Thread**
  By default the application will communicate with a connected queue manager using two threads. One thread is used for outbound messages put to the target queues and a second thread sits in an MQGET waiting for the answers to those requests and any messages from other **MO71** instances. In some cases the use of a second thread may be too wasteful of system resource, for example over a client connection. In those cases you can check this box and the application will run in single thread mode. This does have some disadvantages though, most notably that messages need to be 'expected'. It would be most inefficient if every few seconds the application issued an MQGET on the off chance of retrieving a message. Consequently it will only issue the MQGET if it has recently issued a command to the Queue Manager. Provided that you are just using **MO71** to issue commands to a Queue Manager and display the results you should not notice too much difference running in single thread mode. Responses may take slightly longer to be displayed since the application must now 'poll' for the response. If the command server does not respond within about 10 seconds or so it may be necessary to issue another command, say Refresh, to see the responses.
- **MQTT Commands**
  When checked the MQTT commands will be shown for this location. However, if unchecked the MQTT commands will be hidden.
- **AMQP Commands**
  This option controls whether the AMQP Channel Status dialogs are offered as choices in the commands menu.
- **SYSP Commands**
  When checked the DISPLAY ARCHIVE / LOG / SYSTEM commands will be issued and their results displayed on an individual z/OS queue manager dialog and on a multi-queue manager list dialog. Uncheck this option if you do not want these commands to be issued. This option does not apply to non-z/OS Queue Managers.
- **HTTP Allowed**
  This option controls whether **MO71** will allow this location to be viewed via a browser through the HTTP Web access. Please see 'Web Access' on Page 153 for a description of this feature.
- **HTTP Default**
  This option controls whether this location is considered the 'default' location for the purposes of Web access. In other words the location which is used if no particular Queue Manager name is provided in the URL. Please see 'Web Access' on Page 153 for a description of this feature.
- **HTTP Allow Browse**
  This option controls whether users are allowed to look at the messages on the queues from a browser. It is on by default but can be switched off if required. If it is switched off the users will not see a browse folder icon next to local queues. If the use explicitly enters a URL to browse a queue an error page will be shown. By default **MO71** will look for an HTML file 'error_no_browse.html'. If this file is found then it is sent as the response. If the file is not found then **MO71** will send a 404 error code.
- **Uppercase Options**
  These options allow the user to say whether, by default, characters should be entered into the object names, exit name or namelist names in upper case. When selected pressing the *'shift'* key will allow entry of lower case characters.
- **MQEV Commands**
  When checked the MQEV commands will be shown for this location. Initially this check box will be ticked if MO71 can automatically determine the presence of a queue called MQGEM.MQEV.COMMAND.QUEUE. If your location has "Download Objects" (see below) unchecked, MO71 will not be able to determine this automatically and then you will need to manually check this option. If unchecked (either automatically or manually) the **MQEV** commands will be hidden.
- **MQEV Alerts Console Level**
  This field allows you to choose which **MQEV** alerts, if any, should be added to the **MO71** console. Specifying 'None' is the value to choose if you do not want this feature. If you do use this feature then **MO71** will subscribe to the **MQEV** Alert topic as soon as **MO71** makes connection to this location. If you want to ensure that alerts are always seen then you should also specify "Auto Connect" on the Connection tab.
- **Download Objects**
  When selected the key fields of the major object definitions will be downloaded soon after connection is made with the Queue Manager. This data is used in the Network verification display. For example **MO71** will use the data to traverse resolution paths which will enable you to see which remote queues, on other Queue Managers, may be pointing to your local queues. This data is also used to determine whether you are making use of **MQEV** and to show the **MQEV** related menu options automatically.

- **Download Objects button**
  This button will cause MO71 to download the object definitions for this Queue Manager immediately.
- **Last Object Download Date**
  Displayed after the preceding button, this text will tell the user the last time the objects were downloaded.
- **Download Default Objects**
  When selected the definitions of the default objects will be downloaded soon after connection is made with the Queue Manager. The default objects are necessary if the options to display differences from the default objects are required.

  A fresh copy of the default object definitions for any Queue Manager can be requested manually by pressing the following button or selecting the *File-Refresh Default Objects'* menu option on the main window.
- **Download Default Objects button**
  This button will cause MO71 to download the default object definitions for this Queue Manager immediately.
- **Last Default Object Download Date**
  Displayed after the preceding button, this text will tell the user the last time the default objects were downloaded.
- **Disable Read Ahead**
  When selected this option will disable read ahead when browsing queues from this location. The option only has an effect when the location is a client connection. Read ahead can give a significant speed improvement when browsing queues, particularly over a slow network. However, it can mean that more messages than necessary are sent to the client from the server. For this reason the user is able to disable read ahead processing.
  An option in the preference dialog allows the user to disable read ahead for all locations, please see "Preferences Dialog" on page 193 for more information.
- **Show Splash Screen**
  This button will cause **MO71** to display the Splash screen which is currently in effect for this Queue Manager. If no Splash screen is currently configured then a message is shown to explain that. Please see Splash Screens on page 103 for a description of this feature.

24.3.5. Monitoring

- **Monitor Queue**
  The name of the queue at the remote location which will receive the monitor messages.
- **Disable Monitoring**
  This option, when checked, allows the user to disable monitoring for this location.
- **Enable Alarm**
  When selected an 'alarm' event will be signalled during monitoring if this location does not respond within a given time period.
- **Loop Monitoring**
  When selected it implies that the monitor queue name is actually just a remote queue at the far location which points back to the applications reply queue.
- **Log Console**
  When selected and monitoring is enabled an entry will be added to the console to indicate whether monitoring to this location is successful or not.
- **Monitor Interval**
  This value determines how frequently, in seconds, monitor messages should be sent to this location. The field value can be a number of different fields, the format is as follows :-

  [OK Monitor Interval ['(' OK Expire ')'] [',' Bad Monitor Interval ['(' Bad Expire ')']]]

  The first interval gives the number of seconds between each monitor message provided responses are being received. The second interval gives the number of messages between monitor messages if responses are not being received. This can prevent a large build up of messages if a remote location is not available.

  By default the monitor messages are set to expire a few seconds after a further monitor message is to be sent out. This can be overridden, however, with an explicit value specified in the braces.

  For example the value '**30 (60),3600 (7200)**' has the following meaning :-
  If the target location is available and responding send a monitor message every 30 seconds, each with an expiry of 60 seconds. If the remote location does not respond, send a monitor message every 3600 seconds (1 hour) each with an expiry of 7200 seconds (2 hours).

If a remote location is not available then the monitor message will probably be in a queue waiting to be delivered to that target queue manager so it is not necessary to continually send new monitor messages since the first ones are likely to be delivered eventually. Bear in mind that monitor messages are non-persistent though, so they will be lost if an intermediate queue manager is ended. The default value is 60,1800, note that values cannot be lower than a pre-set value.

- **Last Send**
  The time the last monitor message was sent.
- **Last Receive**
  The time the last monitor message was received.
- **Average Response**
  The average response time, in milliseconds, for monitor messages[45].
- **Reset Average**
  Resets the average response time calculation.
- **Last Response**
  The number of milliseconds the last monitor message round trip took[45].
- **Command OK**
  You can enter here a command string will which be executed whenever MO71 detects that monitoring to the remote location is successful. Note that the command will only be executed when there is a change of state. For example, the first time MO71 receives a response from a remote location it will issue the command but each successful monitor message from there on will not issue the command. However, should monitoring then fail the next successful monitor will cause the command to be issued.
- **Test Command OK / Test Command Fail**
  These buttons cause MO71 to issue the command regardless of the state of the location. This allows testing of the command and/or syntax.
- **Command Fail**
  You can enter here a command string will which be executed whenever MO71 detects that monitoring to the remote location fails. Note that the command will only be executed when there is a change of state as above.
- **Monitor Now Button**
  This button can be used to force a monitor message to be sent immediately.

24.3.6. Export

This tab allows the user to specify values so that the object definitions of a Queue Manager are saved on a regular basis. This can be useful to be able to re-create the Queue Manager definitions on another machine or on the same machine in the case of disaster recovery situations.

- **Auto Export**
  A simple checkbox controlling whether MO71 should automatically export objects based on the export frequency or only do it manually when the **'Export Now'** button is pressed.
- **Objects**
  A list of the object types which can be exported. Only those object types which have a 'tick' next to them will be exported. For an MQSC export, in the case of the Queue Manager, the export is of the form of an ALTER rather than a DEFINE. Most of the object names are self-explanatory. However *Queue Manager(Other)* is perhaps worth explaining. If all you want is the actual Queue Manager definition then you have this unchecked. However, if you want MO71 to also capture the ancillary data, such as Queue Manager status, that is is displayed in the Queue Manager dialog then you should check this option. Note that *Queue Manager(Other)* does not does not apply to an MQSC export.
- **Export File**
  Enter the name of the file which should be used to store the object definitions. The export file format can contain character inserts which are described in chapter 24.8 File Name Inserts on page 228. Below the export file field is a read-only field which will show what the resultant file name is. Note that any time inserts are only representative since the time that will be used is the timestamp of the actual export. The file directory structure will be created if necessary at export time.
- **Export Type**
  Select one of the five export types:-
  - **MQSC** An MQSC define (or alter, set etc) command for the object
  - **Text** A text representation of the object suitable for inclusion in a report
  - **CSV** A comma separated value list suitable for import into anther program such as a spreadsheet

[45]Note that prior to version 7.1.0 this value was displayed in seconds

- **XML** An XML representation which is useful if you wish to have another program which will parse and process the data.
- **JSON** A JSON representation which is useful if you wish to have another program which will parse and process the data.

- **Frequency**
  Enter the frequency with which you would like an automatic save of the Queue Manager objects to be performed. The frequency can be entered in one of two ways
  - **Explicit time**
    An explicit time takes the form of a day followed by a time. So, for example :-
    - **Monday 09:00**
    - **Everyday 15:00**

    Only the first 2 characters of the day need be specified.
  - **Time Interval**
    The units of time are Weeks, Days, Hours and Minutes. So, for example, enter values such as
    - **1 day**
    - **12 hours**
    - **1 week 2 days 4 hours**
- **Default Objects**
  A simple checkbox controlling whether default objects are exported.
- **System Objects**
  A simple checkbox controlling whether system objects (those starting "**SYSTEM.**") are exported.
- **Default Differences**
  When selected only the differences between the current object and the default object definition is exported. This can be useful to export only the key values and allow installation defaults to still apply when the exported script is executed.
- **Blank Fields**
  When selected, fields that have a blank value are included in the export. This may be useful when a value is provided in a default object and then blanked out in another definition.
- **Shared Queues**[46]
  When selected, queues with a QSGDISP of SHARED will be exported.
- **Groups Objects**[46]
  When selected, objects with a QSGDISP of GROUP will be exported.
- **Copy Objects**[46]
  When selected, objects with a QSGDISP of COPY will be exported.
- **Next Export**
  This is time that MO71 intends to try and export the Queue Manager objects. If the value is blank then MO71 is not currently configured to auto-export the objects.
- **Last Attempt**
  A simple time display of the last time (since MO71 started) that an attempt was made to export the object definitions.
- **Last Export**
  A simple time display of the last time the objects were successfully exported.
- **Export Now**
  When pressed MO71 will run the export process now rather than waiting for the export frequency to be reached.
- **Status**
  A simple status field reporting what part of the export process has currently been reached.

[46]Option only available for z/OS Queue Managers

24.3.7. Pub/Sub

This tab allows you to set up the configuration options to allow you to administer the Publish/Subscribe objects in the broker.

Note that these fields apply to Queued Pub/Sub and are not needed for native MQ Pub/Sub

- **Manage Publish/Subscribe**
  This checkbox controls whether Publish/Subscribe commands should be available in the application for this location
- **All Identities**
  Objects such as such as Publishers and Subscribers can be added to the broker anonymously and as such will normally not be returned when a list of subscribers are requested. By selecting the 'All Identities' option even the anonymous Publishers and Subscribers are reported. This option does require greater authority to the Queue Managers, for further information refer to the Publish/Subscribe User Guide.
- **Broker Queue**
  The name of the primary broker queue which should be used when sending control messages.

24.3.8. Network UI

These fields can be used to set the icon displayed for this location in the Network display. If not specified the default values from the Preference Dialog will be used, please see "Preferences Dialog" on page 193 for more information. Note that a different icon can, and probably should, be displayed for each different state. These icons can be different sizes for the different states. For example, you may think it appropriate to have a larger icon for 'online' Queue Managers than for 'offline' Queue Managers. For further details please see Appendix A.7. User Icons on page 269.Applications

24.3.9. Applications

This tab allows you to configure the settings concerned with Application monitoring.

- **Refresh Rate**
  This field sets the frequency at which MO71 will query application information for this location. The special values of 'Off' and 'Default' are also supported. 'Off' indicates that no monitoring is required for this location. A value of 'Default' tells MO71 to use the frequency defined in the .
  Please see 'Time Interval Format' on Page 232 for the format.

- **Refresh Rate (Appl. View Active)**
  This field sets the frequency at which MO71 will query application information for this location when the application view is active. This means that you can set a higher frequency while you are viewing the application view diagram.
  Please see 'Time Interval Format' on Page 232 for the format.

24.3.10. Naming

This tab control aspects of naming particular to a single Queue Manager.
For more information Chapter 8. Naming Objects on page 74.

- **Queues, Xmit Queues, MCA Channels, MQI Channels, Processes etc**
  These fields allow you to specify a comma separated list of local naming patterns for the respective object. A local pattern is much simpler than a global naming pattern and can only contain '*' and <qmgr> values. These fields should only be used to specify those elements of the names that are peculiar to this Queue Manager. For example, on a test Queue Manager one might have a value of ***.TST**

- **Check names on define**
  If checked then MO71 will check that an objects matches any defined global and local name patterns for this type of object[47]. Note that the name must match **both** the global and local patterns.

- **Check names on health check**
  If checked then MO71 will verify that objects matches any defined global and local name patterns when the Verify Display of the Network Display is shown. The checks will only be made on the major objects of queues, channels, processes and topics.

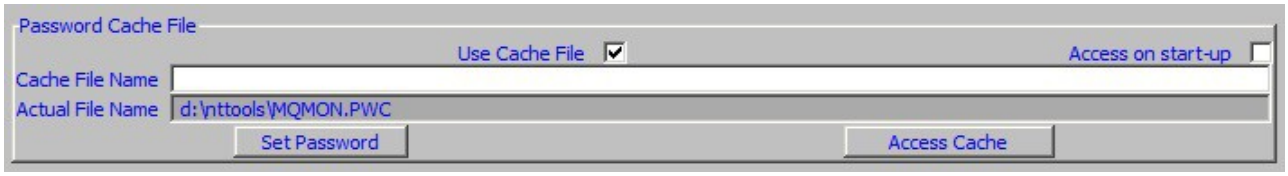
[47]This check is only made when issued from a dialog. The check is not made if issued via MQSC.

24.4. Password Cache File

Since Version 6.0 MQ has had the ability to pass a userid/password combination across the connection. However, in Version 8.0 MQ will actually validate the userid/password combination against either the OS credentials or an LDAP repository. This means that providing a userid/password combination at connect time is now an easy and viable way of validating the user to MQ. However, it also means that, potentially, the user has to supply a set of passwords each time they run any applications, including MO71. To mitigate this problem MO71 provides a password cache file. This is a file which contains the userid/password combinations for all the locations. These passwords are stored in encrypted form. Access to the password cache file itself requires a password. With a password cache file configured the user need only 'logon' to the cache file and all the credentials for the locations can then be provided automatically from the cache file.

24.4.1. Configuration

In order to use a password cache file you must tell MO71 to use the file. You can then set, on a per location basis, which of the locations should store their passwords in the file. The setting of the cache file name is done in the Connection tab of the Preferences Dialog. On this tab we see a number of fields related to the cache file in a box labelled 'Password Cache File'.

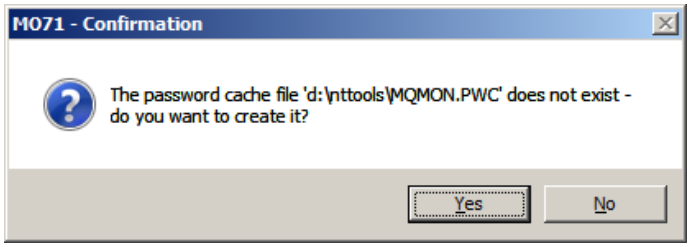


The two key pieces of information are a check-box indicating whether MO71 should use the cache file, and the name of the cache file itself. The cache file name will default to a file called 'MQMON.PWC' in the same directory as the MO71 configuration file, however, if you wish you can specify an alternate file name to use in the entry field. You could just simply give the full file name you wish MO71 to use, or see Specifying the Cache File Name below for other options.

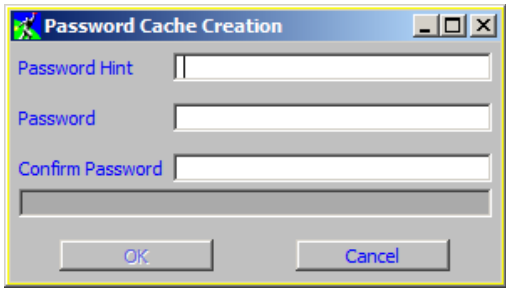
By default MO71 will access the cache file only when it needs to. For example, when connecting to a location which has a password stored in the cache. However, by checking 'Access on start-up' MO71 will prompt the user for the cache password as soon as you start MO71. This has the advantage that it “gets it out of the way”.

If you specify a cache file that doesn't currently exist MO71 will try to create it for you. Note that MO71 will create a file but it will not create the directory structure. If you wish to use a file in different directory it will be necessary for you to create the directory if it doesn't already exist. This is done so that you create the directory with the right security access. Bear in mind that the password cache file contains your Queue Manager password so you want to store the file in a secure location that, ideally, only you have access to.

So let's assume this is the first time that we are using a cache file. MO71 will present the message shown to the right:



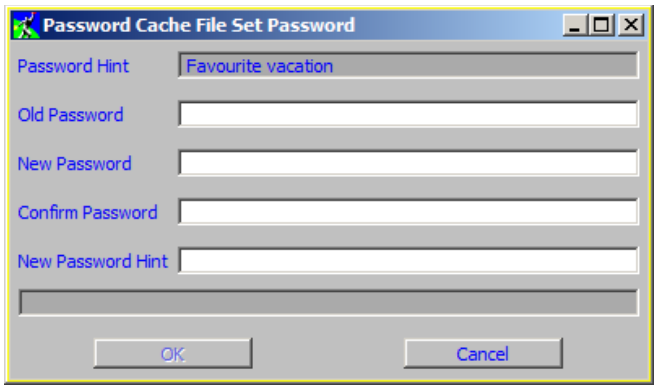
This alerts the user that you really are creating a new file. If you select 'Yes' then MO71 will present the dialog below:



Since this cache file does not currently exist you have the opportunity to set the password for the file. MO71 also allows you to set a password hint if you wish. The can be useful if you have lots of passwords you need to keep track of. You can provide a clue that should only mean something to you but gives you a reminder as to what your password for this system is. It goes without saying that you should not make the hint too obvious or guessable.

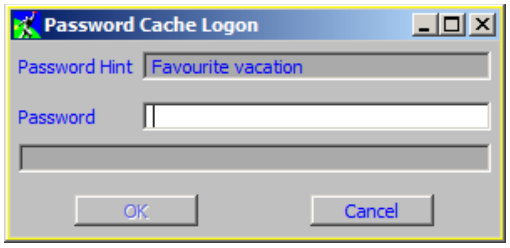
Anyway, fill in the fields you wish and press OK. If you did it right the dialog disappears and the Preferences dialog now shows the button as 'Cache Accessed'. This indicates that MO71 currently has access to the cache. The cache file itself has now been created and will exist in the file system. By all means go and have a look at it. If you are interested in the file format then take a look at Cache File Format on page 223 which describes a little of how the file is constructed.

If at any time you wish to change the password of the cache file then you can press the 'Set Password' button. This will show a dialog like the following:



This dialog requires that you provide the cache password and gives you the opportunity to provide a new password and a new hint for the file. As with most password mechanisms it is advisable to change the password on your cache file on a regular basis. MO71 will not force you to change the password at any interval.

Once the cache file has been created, then when MO71 has need to access the file it must be given the same password. You can either pass the password in as a parameter to the program (see parameter **-P** in MO71 Parameters) or you will be shown the following dialog:



MO71 will give you four attempts to get the password right. If after four attempts you still have not provided the correct password then the cache file functionality will be suspended and you will have to enter all location passwords manually. The only way to re-instate access to the password cache is to end and restart the MO71 application.

24.4.2. Specifying the Cache File Name

By default MO71 will use a file called MQMON.PWC in the same directory as the MO71 configuration file. However, there are times where you might wish to change this location. For example, you might wish to have a specific directory with very stringent access rights. Another reason may be that you share the same configuration file for multiple users but wish each user to have a different file and/or directory. You can achieve these by entering your own file name in the given entry field. However, as you may expect it may not be quite as simple as just typing the file name you wish to use. The rules MO71 follows is:

- If the path is not specified then the MO71 configuration path is used
- If the file name is not provided then MQMON.PWC is used
- Any sequence of characters %XXX% will be replaced by the value of environment variable XXX

As you type a value into the entry field you will see that the value of the 'Actual File Name' change. So, at any time you can see the exact file name that MO71 will use. For example to give each user their own cache file in their own directory it may be as simple as specifying a value of %USERPROFILE% depending on what environment variables you have defined.

24.4.3. Cache File Format

The cache file is just a normal OS file. You can look at it in 'notepad' or load it up into your favourite editor. However, perhaps not surprisingly you will not see the passwords of either the cache file itself or the Queue Manager locations in the file. Instead these values are encrypted. MO71 uses two mechanisms. The cache file password uses a one-way hash algorithm to construct an authorisation code. The authorisation code of the original password and any given password at access time must match.

The passwords for the Queue Managers themselves use a different encryption algorithm that needs to be reversible so that the password can be passed to MQ. The key for this encryption is based on the cache file authorisation code. So, the same Queue Manager password will be encrypted differently depending on the cache file password itself. The fact that the password encryption is reversible is is one of the reasons why it is recommended that you change the password on your cache file from time to time. Note that the encryption of the passwords uses a non-trivial encryption routine but, like any encryption mechanism, it could be decrypted using a brute force attack and sufficient time. It is therefore recommended that you take the additional precaution of ensuring that your password cache file is protected from prying eyes as much as possible. Put the file in a protected directory. When MO71 creates the file it will make the file non-shareable by default which should mean that only the current user and machine administrators have access to the file.

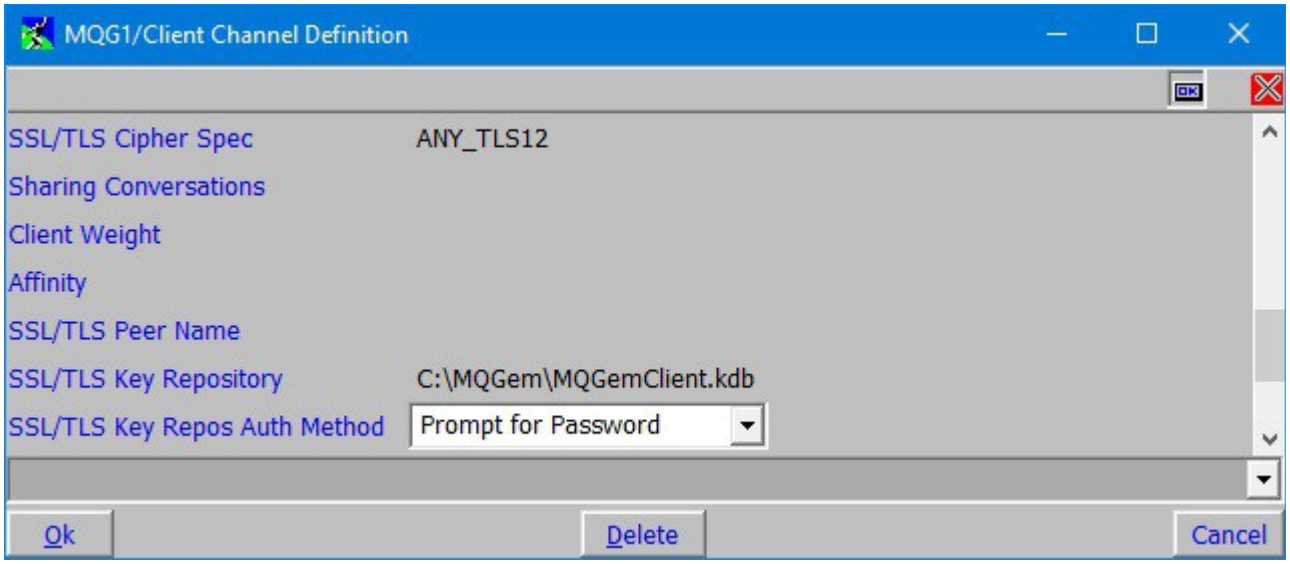
The file format itself is fairly straightforward. You can if you wish edit the file manually as long as you follow a few simple rules.

- Comment lines may be added a anywhere in the by starting the line with an asterisk (*) character.
- Blank lines may be added anywhere
- The first two 'active' lines in the file must be the AuthCode and Hint lines
- Queue Manager lines must be on a single line with fields Qm, Q, User, Pwd in that order.

So, if you wish to add comments to the password file then by all means do so. Another thing that may be convenient is to delete passwords for certain locations. MO71 does provide a 'Delete Password' button in a location dialog but if you wished to delete the passwords for, say, twenty locations it may be quicker to just edit the file and remove the lines you want. Bear in mind that if you do choose to edit the file then you should try and ensure that MO71 is not trying to access the file at the same time. Of course at any time you can always just delete the file and re-create it. Clearly the user would then be prompted to re-enter any Queue Manager passwords as required. These would then be stored in the a newly created password cache file.

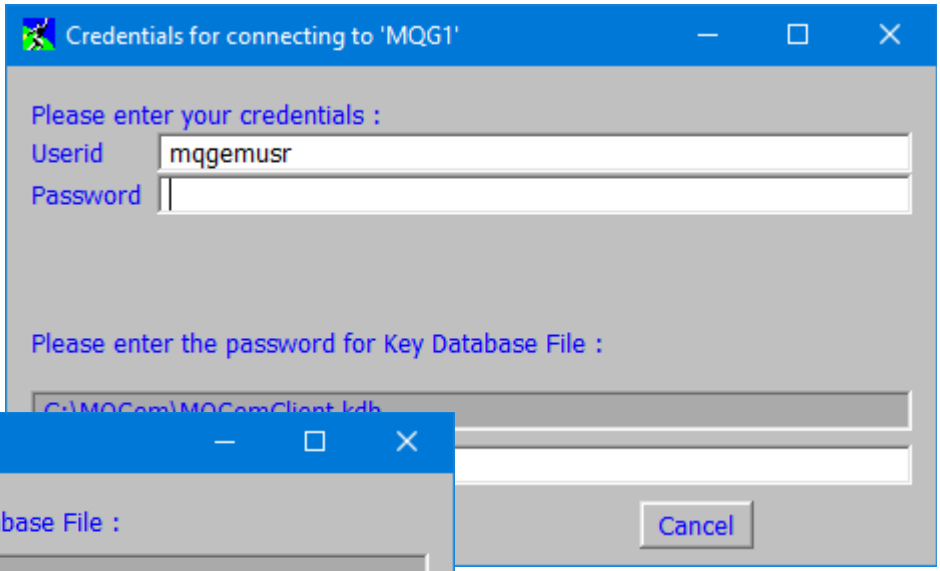
24.5. Key Database File Password

IBM MQ V9.3.0 removed the reliance on a stash file in order for the MQ Client to know the password to the Key Database (KDB) file and allowed the password to be provided on the connect call. To use this in MO71, configure your client connection from the location dialog and ensure your KDB file name is provided in the “SSL/TLS Key Repository” field; then choose “Prompt for Password” in in the “SSL/TLS Key Repos Auth Method” field. The alternative choice in this field “IBM MQ Configured” assumes that your KDB password will be found by the MQ Client in either the MQKEYRPWD environment variable; the mqclient.ini file field SSLKeyRepositoryPassword; or a Stash File.

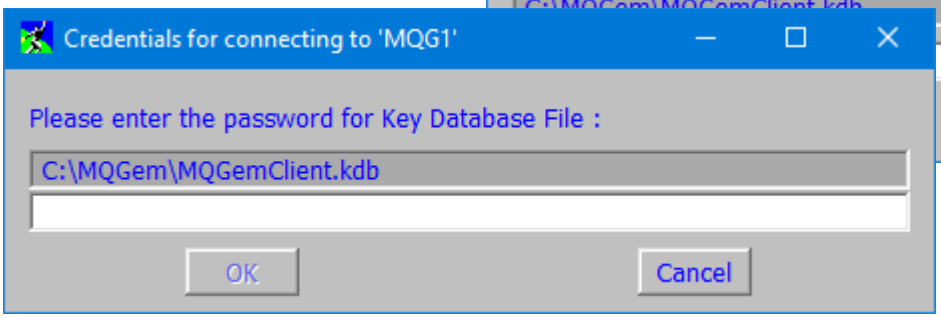


Using the “Prompt for Password” method will cause MO71 to prompt you for the password when it is needed. The password will be cached in memory for the lifetime of the program, and all queue manager connections specifying the same KDB file name will use the same remembered password.

A dialog will be presented for you to enter the password. If you also need to provide a user id and password to connect to the queue manager (and you are not using the password cache) the password prompt dialog will ask for both at the same time to reduce the number of pop-up dialogs you are presented with.



So you will see a dialog like one of the following:



If you change the password of the KDB file and don’t want to cycle MO71, you can use the “Forget KDB Password” button found on the Security tab of the Location dialog (see 24.3.2. Security on page 215) to clear what MO71 is remembering and force the password prompt dialog to be used for the next connection that needs it.

24.6. Container Windows

A number of Dialogs in MO71, which display hierarchical grouped data, contain Windows called 'Container Windows'. A container is a window suitable for showing text, bitmaps and buttons in a tree view. They are used in such features as the Network View, the Application View and the Activity Trace viewer,

MO71 Container windows are similar in concept to the standard windows container control but I had a number of problems using the standard control. Firstly it was very difficult to add buttons to the control. Secondly the standard windows control is rather slow to update and paint, particularly when it has thousands of entries. For this reason I decided to write my own container control from scratch which allows me full control over the interface, features and performance.

The container can be driven either by mouse or keyboard. For keyboard control the item in focus is shown either with a slightly different background colour or with a focus rectangle around the text.

24.6.1. Keyboard Control

The following keys can be used to navigate the container:

Key sequence	Action
Arrow Keys	To move the focus from one item to another.
Space	To activate a button, selectable text or a sub tree arrow
Home	Move to top of container data
End	Move to end of container data
PageDown	Move one page down through container data
PageUp	Move one page up through container data
Shift+Home	Move to the far left of container data
Shift+End	Move to the far right of container data
Shift+PageDown	Move one page to the right in the container data
Shift+PageUp	Move one page to the left in the container data
F9	Toggle Expand/Contract state of current line
F9+Shift	Expand current line and all sub trees of current line
F9+Ctrl	Contract current line and all sub trees of current line
F9+Alt[48]+Shift	Expand all lines in container
F9+Alt[48]+Ctrl	Contract all lines in the container
Ctrl-C	Copy selected text to the clipboard Note that only the text is copied, any icons included in the selection area will not be copied.

[48]Depending on keyboard mapping it may be necessary to use the 'Alt Gr' button rather than just the 'Alt' button.

24.7. Text Inserts

In various places such as the Formatted Time, Export Title line and the Print Dialog you can specify text strings which contain inserts. This means that a fixed string can show more dynamic content, for example, such as the current date. The general mechanism is that inserts are identified by specifying a '%' character followed by one or more characters denoting the type of insert. However, note that a new line is signified using a backslash (\) character. Note that these inserts do not apply to Media Titles, the inserts for those entries are described in Chapter 24.2.7.2 Media Titles on page 204.

The available inserts are as follows:

Insert	Meaning	Conditional	Notes
%%	The % character		
%c	Current Command Name	Yes	
%m	Queue Manager Name	Yes	
%l	Location Name	Yes	
%d	Day short form e.g. Sun, Mon, Tue		
%dd	Day e.g. Sunday, Monday, Tuesday		
%D	Day of the month e.g. 16		
%DD	Day of the month e.g. 16th		
%DP	Preferred date format		
%e	Elapsed time in H:MM format		Amount of time until now in H:MM format
%E	Day of the month (always 2 digits)		
%H	Hour (24 hour clock)		
%HH	Hour 2-digit (24 hour clock)		
%HHP	Hour 2-digit (12 hour clock)		Adds
%HP	Hour (12 hour clock)		
%I	Minutes 2-digits		
%j	Julian day (0 based)		
%J	Julian day (1 based)		
%M	Month number e.g. 11		
%MM	Month short form e.g. Jan, Feb, Mar		
%MMM	Month e.g. January, February, March		
%n	Page number		Printing only
%N	Total print pages		Printing only
	Month number (always two digits) e.g. 02		Time Values only
%p	12-hour clock meridian indicator		Adds either am or pm depending on time
%P	12-hour clock meridian indicator		Adds either AM or PM depending on time
%s	Preferred date separation character		
%S	Seconds 2-digits		
%Y	Year short form e.g. 03		
%YY	Year e.g. 2003		
%t	Activity time This character can be followed by any time formatting characters. For example **%tTT**		Show monitor time only This insert acts on the activity time rather than the current time.
%T	Time e.g. 22:18		
%TT	Time e.g. 22:18:35		
%w	Day of week Mon,Tue,Wed,Thu....		
%W	Day of week Monday, Tuesday,....		

%z	Timezone – Local, UTC or Origin		
\n	New line		Export title only
\\	\		

24.7.1. Conditional Text Inserts

Not all of these values are available all the time. For example, depending on the context there may be no current Queue Manager and so it will have no current value.

Let's consider a print example. Suppose one wished to have a header that printed

```
Queue Manager: QMNAME
```

This would be fine for printouts that had a current Queue Manager but on the others you would just get :-

```
Queue Manager:
```

This is not very desirable. We need a way of switching off text if a variable is not present. The answer is to enclose the text in brackets. By specifying a header of '%m{Queue Manager :%m}' we are saying only print out the contents of the brackets if you have a value for %m.

24.8. File Name Inserts

When exporting from a dialog and when defining an event you can specify the name of a file to write the data to. If you like you can specify inserts in the file name to make the file name more dynamic. For example, you can put the time of day or the date in the file name, The inserts are all specified following a % (percent character). The inserts and their meanings are as follows:

Insert	Effect
q	Queue Manager name
l	Location name
i<*n*>[49]	Cyclic index number For example **file%i3** will write to files file1,file2,file3,file1,file2…. etc
g	Group Name or "NoGroup" if the location is not in a group
H	Two digit hour
M	Two digit minutes
S	Two digit seconds
d	Two digit day of month
m	Three character month name eg.Jan,Feb,Mar….
2m	Two digit month number eg. 03 for March.
y	Four digit year
2y	Two digit year
D	Three character day of week eg.Mon,Tue,Wed…
t	Simple time format eg. 18.14.03
j	Julian day of year (zero based)
J	Julian day of year (one based)
%	Percent character

Figure 22:File Name Inserts

[49]This insert is not supported in an event definition

24.9. Reply Queue Details

The Administrator uses a reply queue to receive all of its reply messages from a Queue Manager it directly connects to. By default the name of this queue is SYSTEM.DEFAULT.MODEL.QUEUE[50] but can be overridden by changing the Reply Queue name in the location definition. The local or model queue of whatever name you choose must already exist on the respective Queue Manager.

If more than one instance of the Administrator is run against the same Queue Manager then each instance must use a different local reply queue or a model queue.

24.9.1. Model Reply Queue

Using a model queue is useful if many instances of MO71 will be run using the same configuration file at the same time. The dis-advantage of using a model queue as the reply is that an instance of MO71 has no 'fixed address'. In other words the reply queue name is not predetermined. This means that you cannot use loop back monitoring since that requires a predefined remote queue.

When a model queue is used as a reply queue the actual queue name used is governed by the DynamicQName specified in the object descriptor. The Reply Prefix field of the location dialog allows the user to specify the value for DynamicQName field. The field can be any sequence of characters providing they are valid Queue name characters. Two special characters have special significance:-

*	An asterisk at the end of the string indicates that MQ should add a sequence of hex characters to ensure that the Queue name is unique.
%u	This sequence indicates that the current userid should be inserted at this part of the name.

If the reply prefix is empty the value of ***MQMON.%u.**** will be used. On my own machine using this default setting, with a userid of CLARKEP, the queue names I get are of the form **MQMON.CLARKEP.412EE8F603370020**

24.10. Colour Dialog

The colours of most of the elements of the application can be configured in the colour dialog available under the 'View' menu in the main window.

The colour dialog displays all the dialog elements in a list and a number of action buttons. The list of elements shows the colour of element when the dialog was started, a text description of the element, the colour of the selected scheme for this element and the current colour of the element that would be applied if the 'OK' or 'Apply' button was pressed.

The element list allows multiple selection using the standard extended selection scheme. The action buttons will either apply to those items selected or to all the elements in the list.

For example, to change to use the Grey colour scheme use the following sequence.

1. Select the Grey scheme in the colour scheme drop down list
2. Press 'Reset All to scheme' button
3. Press 'Apply' button

24.10.1. Colour Schemes

Colour schemes are a fast and simple way to set all the colours in the application to a predefined set of values. When the dialog is displayed the colour scheme which most closely matches the colours you have selected will be preselected. You can use the colour scheme as it is provided or use it as a base and make a few selections to certain fields.

If you create a colour scheme which is significantly different than the provided set of schemes then by all means send me your scheme so it can be included in the next release of the program. To do this run the program with the -d flag, for example **'mqmonntp -d'**. In the colour dialog you will now see a **'write'** button. By pressing this the program will write a file called **c:\temp\schemes.txt.** Please send me this file, together with the name you'd like to use for the scheme. Here is an opportunity for you to get directly involved in the development of MO71!

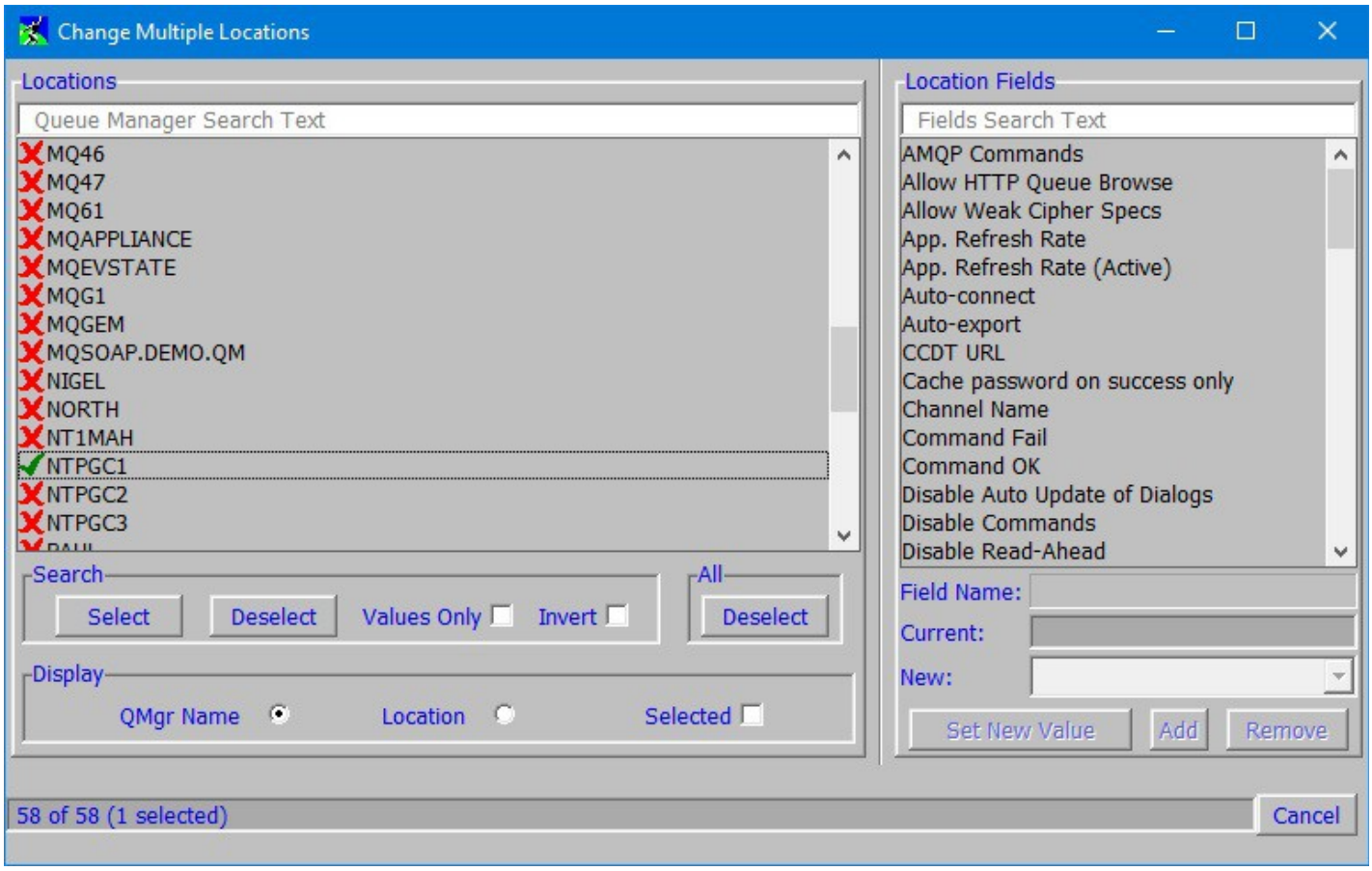
24.10.2. Extended Selection

By holding down the **'Alt'** button at the same time a selection is made from the colour list the application will select (or deselect) not only the particular item in question but all the items in the colour list which have the same current colour. This allows the user to easily change all the items that use the same colour to now use a different colour. On some systems the two **'Alt'** buttons behave slightly differently. One is additive to the current selections and the other will replace any previous selection.

[50]Earlier versions of MO71 used to set a default value of MQMON but this required that the MQMON queue was predefined on the Queue Manager before the program could run against the Queue Manager.

24.11. Multiple Location Change Dialog

There can be times, particularly if you have a large number of location definitions, where you want to make the same change to many different locations. For example, you want them all to use the same Local Address or perhaps you want to add a group of locations to a particular Network Name. To do this the easiest way is to bring up the "Change Multiple Locations" dialog. This is available as a menu option under the *File* menu. By pressing that menu you will get a dialog such as the following:



The dialog is essentially split into two halves. On the left you have the locations to be changed and on the right you can select the field you would like to change. The dialog is fairly intuitive but we shall describe each field you have:

- **Queue Manager Search Text**
  You can type in a text string here and locations containing that string will be shown. The fields which are searched are:
  - Queue Manager Name
  - Location Name
  - Queue Manager Group
  - Network Names
  - Security Group Name

  If just a string is entered for example **MQ** then MO71 will do a wildcard search as though ***MQ*** was entered. However, if you want to do a complete word search then you can enter the text in quotes, eg "**MQ**"

  If any locations are selected but they **do not** match the search string then they will still be shown but below a black bar in the list. This ensures that all the selected locations are always displayed and it is clear what locations will be affected.

- **Queue Manager Locations List**
  Here are listed the locations that can be acted upon. Essentially any field change will only be applied to the Queue Manager locations which are selected. The selection state can be changed just by clicking on the location and is displayed by either a tick or cross.

  The search string above can be used to limit the locations shown and make finding locations easier. There are also various fields below which can make finding the locations you want easier.

- **Search**
  - **Select**
    This button will select all locations which match the search criteria
  - **Deselect**
    This button will deselect all locations which match the search criteria
  - **Values Only**
    Selecting this option will show only those locations which have a value for the selected field.
  - **Invert**
    Selecting this option will invert the current search criteria
- **All**
  - **Deselect**
    This button will deselect all locations
- **Display**
  - **Qmgr Name**
    This option will show locations by their Queue Manager Name
  - **Location**
    This option will show locations by their location name.
  - **Selected**
    Selecting this option will only show currently selected locations
- **Field Search Text**
  This field allows you to reduce the number of displayed fields by entering a simple string. Only fields either containing the string or which are related in some way will be shown. To that end there are certain keywords that can be entered which will match the fields. The list of keywords are:

connection	Fields related to making a connection to the server
channel or **chl**	Fields which are part of the local channel definition
ccdt	Fields related to the CCDT file
security	Fields related to security
name or **naming**	Fields related to the specification of naming conventions
icon or **bitmap**	Fields which specify an icon or bitmap
network	Fields related to the specification of the network view
monitoring	Fields related to Queue Manager monitoring
command	Fields related to send commands to the Queue Manager
export	Fields related to exporting definitions
HTTP	Fields related to Web Access; displaying data in a browser window.

- **Field Selection**
  Here is a list of all the fields which can be changed 'in bulk'. Not every field is available since many of them don't make sense. For example, you can't change the 'Connection Name' since the expectation is that not many locations will exist at the same IP address and port combination.

  Selecting a field will have a number of effects:
  - The field name will be put in the 'Field Name' field
  - The Dialog title will contain the selected 'Field Name'
  - The current value field will show the current value of this field for the selected locations unless those definitions are different in which case it will display ***<Multiple Values>***
  - The current value for the field for each location will be shown in the left-hand list.
  - The new value drop-down is populated with the values of this field across all the Queue Manager.

- **New**
  Here you can enter the required new value for the field. The drop-down will contain all 'known' values.
- **Set New Value**
  Pressing this button will change all selected locations to have the new value. This change will be reflected in the left-hand list and also any open location or client channel dialogs.
- **Add**
  Some values are comma separated lists. For example Network Names. For these fields the 'Add' and 'Remove' buttons are available and instead of replacing the entire definition pressing the add button will add the New value to the comma separated list if it is not already present. The New value is always added to the end of the list.
- **Remove**
  This button will remove the New value from any comma separated lists if present.

24.12. Time Interval Format

A number of configuration options are some form of time interval. These parameters use a common input format. The format allows for a list of value and unit pairs. The units are **milliseconds(ms), seconds, minute, hour, day, week.** Generally speaking only the first letter of the unit need be used. The following inputs are examples of valid input.

- 1 day 3 hours 25 minutes
- 1d3h25m
- 35 ms
- 25

In cases where no unit is specified the default unit for the particular field is used. Note that not all units are valid for all field types. Generally speaking fields which have a default unit of milliseconds do not allow units larger than seconds. The user should always press the ‘Apply’ button and verify that the output field value matches the value which was required. Note that the output format will not match exactly the input but will be in the standard value. For example, entering **48h** will be output as **2 days**.

24.13. MO71 Parameters

The Administrator program does not require any parameters but if you wish there are a number of parameters which can modify it's behaviours. Perhaps the two most useful capabilities are:

1. Passing in the location of the configuration file.
2. Passing in the definitions required for a location

The complete list of flags are :-

- **-a**
  Tells the program to auto-connect. In other words, the program will attempt to always have a connection to this location.

- **-b**

  Tells the program to run in 'background mode'. This used when running MO71 as a service, this is useful when you wish to provide Web Access but have the MO71 program itself running in the background. Please refer to “Running MO71 in background mode“ on page 167 for more information.

- **-c <** *Channel Name* **>**
  For client connections you can specify explicitly the channel name to use on the connection. If this option is used then program will assume it’s a TCP/IP connection and you must specify the TCP/IP Host name or address of the machine below.

- **-E**<Mini-dump File Directory>
  This parameter instructs MO71 to install an exception handler and to write any exceptions in the form of a min-dump file to the directory given. The **-F** parameter can be used to control the type of mini-dump generated.

- **-f <***Configuration Directory/FileName***>**
  This parameter can be either just a directory name or the full path to the configuration file to use. By default the program will store the configuration in a file called MQMON.CFG, in the directory where it finds the program. Using this parameter you can override this location and give a different directory and/or file name. Note that the directory specified will also affect the locations of the objects file (MQMON.DFN) and the authorities file (MQMON.AUT). A common usage of this parameter is to specify '**-f** .' which requests that all configuration is read and written to the current directory.

- -**F**<Mini-dump file format>
  This parameter controls what type of min-dump is generated whenever a dump is taken. The following parameters are supported:

Format	Meaning
normal	This format generates a normal mini-dump. A normal mini-dump contains only thread and stack information, no heap memory is included. A normal mini-dump is typically only about 50K but may not be sufficient to diagnose all problems. This is a default if no **-F** parameter is specified.
full	This format causes a full mini-dump to be generated. A full mini-dump file contains thread, stack **and** heap storage information. Consequently the file is much larger, often around 100MB. Full dump files should only be necessary in rare circumstances. Please do not send full dump files to service via email. Instead use an internet file share of some kind.

- **-g <** *Group Name* **>**
  Tells the program which logical group the definition belongs in

- **-h** < *TCP/IP Host Name* >
  Host name of the target machine for client connections. The Channel name parameters must be specified for this parameter to take effect.

- **-j** <*Master Configuration File*>
  Allows you to specify the location of a master configuration file from which location definitions should be retrieved. Please see Chapter 11.6 Master Configuration File on page 105 for more information.
- **-k**
  By default the location created from command line options will not be saved when the program ends. However, by using this flag you can ask the program to keep the definition.
- **-l**
  Tells the program to create a client connection to this Queue Manager.
- **-m** < *Queue Manager Name* >
  This parameter is mandatory if you wish to create a location from the command line. All other parameters are optional and allow you to change the definition that is created.
- **-n**
  Don't autostart any predefined dialogs.
- **-p <***Filter***>**
  Start any predefined dialogs which have the filter string in their description fields.
- **-P<***Password Cache Password***>**
  The password to access the password cache file.
- **-q** < *Queue Name* >
  Specify the name of the 'monitor' and 'reply' queue that should be used. If not specified the queue name **SYSTEM.DEFAULT.MODEL.QUEUE** will be assumed.
- **-r**
  Don't write to CFG file, open for read-only.
- **-s**
  Identifies the location as an MVS machine.
- **-t**
  Tells MO71 to connect to the Queue Manager with only a single thread
- -u
  Gives the location of the CCDT URL if creating the client location from the command line.
- **-v** < *Via Queue Manager* >
  Tells the program to route all messages to this location via this Queue Manager definition. The 'Via Queue Manager' must be already defined in the programs CFG file.

Chapter 25. Usage Tailoring

You can change the menu options available in MO71 depending on the type of user which will be using MO71. This is known as Usage Tailoring. For example, suppose you want some users to be able to display queues but you don't want them to go near your channels or vice versa. You can also remove menu items, such as network view, mqsc, API Execiser etc from the main window. There are two main reasons why you might want to do this. Firstly because you don't want certain users to be able to use powerful features and 'potentially' cause problems. Secondly, you want to hide any superfluous complexity and present the user with a simpler interface.

Tailoring is achieved using a separate configuration file – MQMON.AUT (where .AUT stands for Application Usage Tailoring). This file must be created manually (using your favourite editor) and placed in the same directory as the MQMON.CFG file. If this file is not present then all options will be made available to all users. The MQMON.AUT file can be write protected since it is constructed and updated only by the administrator setting up the tailoring. Providing the application can read the file it can be protected as required. It should be noted that this mechanism is not intended to be fully secure; you are also recommended to make the corresponding authority configurations in your Queue Managers. For the majority of these kinds of situations, however, a simple way of preventing the full range of options being presented to the user is sufficient.

Object tailoring can either be specified additive, for example **queue_display**, or subtractive, for example **!queue_display**. The advantage of the subtractive it that it future proofs for future securities. For example, suppose you want a user to be able to do anything with a queue except delete them then you could use the object tailoring keywords **queue_all !queue_delete**.

The best way of explaining the format of the file is to show an example:-

```
# Set Usage Tailoring for users of MO71
#
# Global Tailoring
queue_display location_display
nomenu_monitoring

# Per Queue Manager Tailoring
QM: NTPGC1
     all
QM: NTPGC2
     location_change
QM: NTPGC3
     queue_change
```

The format of the file is entirely free-format. White space can be added/removed wherever the user chooses. Comments can be added from the first ';' or '#' character to the end of the line.

The file is structured into global tailoring and tailoring on a per queue manager basis. The global tailoring appears before the first 'QM:' keyword. In addition to object tailoring you can specify explicit keywords which will remove items from the various menus in MO71. This can make MO71 much simpler to use, although it naturally reduces the available functionality.

The 'QM:' must be followed by the name of the Queue Manager. This Queue Manager name may contain wildcards characters '*' and '?'. If wildcards are used then the following values will apply to all Queue Managers which match the value. The Queue Manager name itself is followed by zero or more tailoring keywords. Any tailoring keywords specified in the Queue Manager section are additive to the global authorisations. They do not replace the tailoring. So a user running with this file can:-

- Do all operations against NTPGC1
- Display queues and display and change the location for NTPGC2
- Display and change queues and display the location for NTPGC3
- Display the queue and location against all other queue managers

Note that the only queue manager that this user could display channels, processes and the queue manager object itself for is NTPGC1.

25.1. Object Tailoring Keywords

location_create	Allows the user to create new locations
location_change	Allows the user to update this location
location_display	Allows the user to display this location
location_delete	Allows the user to delete this location
queue_create	Allows the user to create new queues
queue_change	Allows the user to change defined queues
queue_delete	Allows the user to delete queues
queue_stats	Allows the user to view queue statistics
queue_usage	Allows the user to view queue usage
queue_display	Allows the user to display queues and queue lists
queue_admin	Allows the user to issue clear queue
msg_create	Allows the user to use the put message dialog
msg_copy	Allows the user to copy messages from one queue to another
msg_move	Allows the user to move messages from one queue to another
msg_display	Allows the user to browse queues
msg_delete	Allows the user to delete messages from queues
namelist_create	Allows the user to create new namelists
namelist_change	Allows the user to change defined namelists
namelist_display	Allows the user to display namelists and namelist lists
namelist_delete	Allows the user to delete namelists
process_create	Allows the user to create new process objects
process_change	Allows the user to change process objects
process_display	Allows the user to display process objects and process lists
process_delete	Allows the user to delete process objects
pubsub_create	Allows the user to create new topics and subscriptions
pubsub_change	Allows the user to change topics and subscriptions
pubsub_display	Allows the user to display topics and subscriptions
pubsub_delete	Allows the user to delete topics and subscriptions
channel_create	Allows the user to create new channels
channel_change	Allows the user to change existing channels
channel_display	Allows the user to display existing channels and channel lists
channel_delete	Allows the user to delete channels
channel_admin	Allows the user to issue administration commands such as reset and resolve channel.
channel_start	Allows the user to start a channel
channel_stop	Allows the user to stop a channel
channel_ping	Allows the user to ping a channel
chlauth_display	Allow the user to display channel authentication records
chlauth_create	Allow the user to create or delete channel authentication records
policy_display	Allow the user to display policy definitions

policy_create	Allow the user to create, update and delete policy definitions.
clusqm_display	Allows the user to display cluster queue manager objects
authinfo_create	Allows the user to create new authinfo objects
authinfo_change	Allows the user to change authinfo objects
authinfo_display	Allows the user to display authinfo objects
authinfo_delete	Allows the user to delete authinfo objects
qmgr_mqsc	Allows the user to get an MQSC window. ***Care should be taken with this option since it essentially gives the user access to the full range of MQSC commands. No parsing of the command is done by the MO71 program.***
qmgr_change	Allows the user to change the Queue Manager definition and issue the SET ARCHIVE / LOG / SYSTEM commands on z/OS
qmgr_display	Allows the user to display the Queue Manager definition and issue the DISPLAY ARCHIVE / LOG / SYSTEM commands on z/OS
qmgr_admin	Allows the user to issue administration commands against the Queue Manager such as the cluster commands RESET, REFRESH, SUSPEND and RESUME cluster.
network_display	Allows the user to display the network view
location_all	Allows the user to do anything to the location objects
msg_all	Allows the user to do anything to messages
queue_all	Allows the user to do anything to queue objects
channel_all	Allows the user to do anything to channel objects
authinfo_all	Allows the user to do anything to authinfo objects
qmgr_all	Allows the user to do anything to the Queue Manager definition
pubsub_all	Allows the user to do anything to the topics and subscriptions
all_create	Allows the user to create objects of all types
all_change	Allows the user to change objects of all types
all_display	Allows the user to display objects of all types
all_delete	Allows the user to delete objects of all types
all	Allows the user to do anything to all object types (MQ or MQEV).
mqev_event_display	Allows the user to display MQEV event data
mqev_statchl_display	Allows the user to display MQEV STATCHL data
mqev_statmqi_display	Allows the user to display MQEV STATMQI data
mqev_statq_display	Allows the user to display MQEV STATQ data
mqev_acctq_display	Allows the user to display MQEV ACCTQ data
mqev_acctmqi_display	Allows the user to display MQEV ACCTMQI data
mqev_data_display	Allows the user to display any of the above sets of data
mqev_alert_display	Allows the user to display MQEV alerts
mqev_alert_admin	Allows the user to do anything to MQEV alerts
mqev_cfg_display	Allows the user to display MQEV configuration
mqev_cfg_admin	Allows the user to create, change or delete MQEV configuration
mqev_display	Allows the user to display any MQEV data or configuration
mqev_admin	Allows the user to do any MQEV administration (currently the same as

	mqev_cfg_admin but in future may include other administration commands)
mqev_all	Allows the user to do anything to MQEV
queue_mask	This keyword allows the user to specify the list of queue names which can be displayed by the user. Filtering is done in MO71 (not on the server). It can be useful to reduce the amount of information which is shown to the user when they display the list of Queues dialog. The format of the entry should be: queue_mask = { <Mask1> , <Mask2>, ….} The list can either be comma or space separated. If the mask starts with a '!' character it indicates that Queues matching the mask **should not** be shown. The order of the masks is relevant and matching is done in the order of the definition. For example, you can define a set which is shown and then remove a subset. Alternatively you can define that nothing should be displayed and then define a subset which should be displayed. For example: queue_mask = { *PRD*,!SYS, SYSTEM.DEF* } This example says display any queues which contain the string 'PRD' except queues which start with 'SYS' should not be shown, except queues which start with 'SYSTEM.DEF' should be shown.
queue_name_required	This keyword indicates that the user cannot use a single '*' in the queue field of the Queue list dialog.
queue_no_generic	This keyword indicates that the user cannot use a '*' character anywhere in the queue field of the Queue list dialog.

25.2. Menu Tailoring Keywords

25.2.1. File Menu

nomenu_about	Removes the named option from the menu		
nomenu_add_location	“	“	“
nomenu_check_mo71_version	“	“	“
nomenu_show_mq_maint	“	“	“
nomenu_change_locations	“	“	“
nomenu_copy_location	“	“	“
nomenu_delete_location	“	“	“
nomenu_help	“	“	“
nomenu_import_ccdt	“	“	“
nomenu_import_local	“	“	“
nomenu_import_location	Will remove all the import actions		
nomenu_import_mo71_config	“	“	“
nomenu_import_mq_explorer	“	“	“
nomenu_open_location	“	“	“
nomenu_preferences	“	“	“
nomenu_print	“	“	“
nomenu_refresh_default_objects	“	“	“
nomenu_refresh_information	“	“	“
nomenu_save_configuration	“	“	“

25.2.2. Action Menu

nomenu_api_exerciser	Removes the named option from the menu
nomenu_compare	“ “ “
nomenu_connect_all	“ “ “
nomenu_disconnect	“ “ “
nomenu_disconnect_all	“ “ “
nomenu_export_all	“ “ “
nomenu_filters	“ “ “
nomenu_http	“ “ “
nomenu_ini_file	“ “ “
nomenu_log_file	“ “ “
nomenu_mqsc	“ “ “
nomenu_predefined	“ “ “
nomenu_predefined_dialog	“ “ “
nomenu_predefined_event	“ “ “
nomenu_publish_message	“ “ “
nomenu_put_message	“ “ “
nomenu_qload	“ “ “
nomenu_talk	“ “ “
nomenu_trace_message	“ “ “
nomenu_user_commands	“ “ “

25.2.3. View Menu

nomenu_colours	Removes the named option from the menu
nomenu_default_lists	“ “ “
nomenu_font	“ “ “
nomenu_foreign	“ “ “
nomenu_list_view	“ “ “
nomenu_mq_version	“ “ “
nomenu_search	“ “ “
nomenu_view	“ “ “
nomenu_view_apps	“ “ “
nomenu_view_console	“ “ “
nomenu_view_network	“ “ “

25.2.4. Monitor Menu

nomenu_graphs	Removes the named option from the menu
nomenu_monitoring	Removes any menu options concerned with QM monitoring
nomenu_monitors	Removes any menu options concerned with monitors

25.3. Dialog Menus

nomenu_alterlist	Removes the named option from the menu
nomenu_autorefresh	" " "
nomenu_context	Removes any context menus (those displayed when button 2 is pressed) from the application.
nomenu_defaultfilter	Removes the named option from the menu
nomenu_defaultobjs	" " "
nomenu_export	" " "
nomenu_initialrefresh	" " "
nomenu_listtitles	" " "
nomenu_msg_format	Removes all the menus concerned with message formatting from the message browsing dialogs.
nomenu_msg_ops	Removes all the operations such as Copy, Move, Delete from the message browsing dialogs.
nomenu_msg_selection	Removes all the menus concerned with message selection from the message browsing dialogs.
nomenu_showhide	Removes all the show/hide menu menu options such as 'Show Buttons'
nomenu_splitlist	Removes the named option from the menu

25.4. Miscellaneous Tailoring Keywords

nomenu_application	Removes any menus concerned with Applications
nomenu_connection	Removes any dialog menus concerns with MQ Connections
nomenu_listener	Removes any menus concerned with Listener objects
nomenu_service	Removes any dialog menus concerned with Service objects
nomenu_set_archive	Removes the Set Archive command from z/OS Qmgr context menu
nomenu_set_log	Removes the Set Log command from z/OS Qmgr context menu
nomenu_set_system	Removes the Set System command from z/OS Qmgr context menu

Chapter 26. Security

MO71 is an MQI application, it is written in 'C' and uses the standard MQI to interface with IBM MQ. It does not use any private interfaces into IBM MQ. As such is it is subject to exactly the same security mechanisms as any other MQ program. Consequently **MO71** does not allow a user to do any more than they could do using other mechanisms, such as writing an MQI program themselves.

MO71 always communicates, through queues, to the command server, even when locally connected to the Queue Manager. Therefore, for successful operation:

- **The command server must be running**
- **The user must have the authority to connect and inquire the connected Queue Manager.**
- **The user must have the authority to display the administered Queue Manager.**
- **The user must have MQPUT authority to the SYSTEM.ADMIN.COMMAND.QUEUE**
- **The user must have MQGET and MQPUT authority to the reply queue which is being used.**
  This means that if the reply queue is a model queue, authority to create a dynamic queue will be needed, and if the reply queue is a local queue then the above authorities will need to be granted on the local queue being used.
  If the reply queue is a model queue, which it is by default, then the reply queue will be created with a dynamic queue name of MQMON.<*userid*>.*. This can be useful in identifying queue usage and also creating security profiles specific to **MO71** if required. Read more about reply queue names in 24.9. Reply Queue Details on page 229.

  On z/OS you will need MQPUT access to the generic profile MQMON.*
- **The user must have the authority run the particular command in question.**
  If no responses are returned to **MO71** then the most likely cause is that the request has been rejected by the command server for some reason. Look on the Dead Letter Queue and see whether there are messages put by the Command Server complaining of a security failure.

Other features in MO71 will have additional security requirements. For example, looking at Application Activity Trace requires you to have access to the queue where it is collected; viewing a list of MQEV alerts benefits from a subscription for immediate notification of newly added alerts; and using the API Exerciser will require authority to use the resources your API calls target.

26.1. Example security commands for Distributed Platforms

Suppose you have a user, in the group MO71ADM, who wishes to use **MO71**. Below are listed the minimum authorities the user requires in order to use **MO71**.

```
SET AUTHREC OBJTYPE(QMGR)  GROUP('MO71ADM')                                    AUTHADD(DSP,CONNECT,INQ)
SET AUTHREC OBJTYPE(QUEUE) GROUP('MO71ADM') PROFILE(SYSTEM.ADMIN.COMMAND.QUEUE) AUTHADD(PUT)
  * either *
SET AUTHREC OBJTYPE(QUEUE) GROUP('MO71ADM') PROFILE(SYSTEM.DEFAULT.MODEL.QUEUE) AUTHADD(GET,DSP)
  * or *
SET AUTHREC OBJTYPE(QUEUE) GROUP('MO71ADM') PROFILE(MY.LOCAL.REPLYQ)            AUTHADD(GET,PUT)
```

26.2. Example security commands for z/OS (using RACF)

Suppose you have a user, in the group MO71ADM, who wishes to use **MO71**. Below are listed the minimum authorities the user requires in order to use **MO71**. It assumes that all the classes in use below are active on the system through the necessary switch profiles, and that the profiles used have already been **RDEFINE**-d.

```
PERMIT qmgr.BATCH                       CLASS(MQCONN) I D(MO71ADM) ACCESS(READ)
PERMIT qmgr.DISPLAY.QMGR                CLASS(MQCMDS)  ID(MO71ADM) ACCESS(READ)
PERMIT qmgr.SYSTEM.ADMIN.COMMAND.QUEUE CLASS(MQQUEUE) ID(MO71ADM) ACCESS(UPDATE)
  * either *
PERMIT qmgr.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(MO71ADM) ACCESS(UPDATE)
PERMIT qmgr.MQSCX.*                     CLASS(MQQUEUE) ID(MO71ADM) ACCESS(UPDATE)
  * or *
PERMIT qmgr.MY.LOCAL.REPLYQ             CLASS(MQQUEUE) ID(MO71ADM) ACCESS(UPDATE)
```

Chapter 27. Troubleshooting

27.1. Troubleshooting interacting with IBM MQ

Because of the nature of messaging the application cannot guarantee that replies from remote Queue Managers will be returned. If you find that a command window continually waits for a response, then some common issue are described below.

27.1.1. The Command server for the Queue Manager is not running

MO71 provides a graphical interface for the various commands that an MQ administrator might need to issue to their queue manager. The queue manager receives and answers those commands through the command server. If the command server is not running, these questions will be ignored and no answer will be returned to MO71.

In modern versions of IBM MQ, the command server is usually automatically started, however, your installation may have changed this default behaviour. To determine whether the command server is running, you can run the IBM MQ control command `dspmqcsv`, which you must run from a command prompt on the queue manager machine, i.e. you cannot run this remotely. If the output from this command indicates that the command server is not running, you can start it using the IBM MQ control command `strmqcsv`, again run on the queue manager machine.

27.1.2. The channels to and from the remote Queue Manager are not running

When connecting MO71 to your queue manager in via mode, the messages sent to the command server, and the reply messages returned by the command server are moved by the channels between your local queue manager and the remote queue manager that you are administering. These two channels, one to the remote queue manager, and one from the remote queue manager, must both be running in order for the messages to be delivered to MO71.

When using MO71 with a client connection, the server-connection (SVRCONN) channel is automatically started as a result of the connection MO71 makes to the queue manager. The above problem is specific to running in via mode.

27.1.3. The user does not have security access to the Queue Manager

The usual symptom of this is that messages will build up on the remote Queue Manager's Dead Letter Queue. The 'Q' program is one example of an application that can be used to browse the messages on the Dead Letter Queue to see the reason code.

The most common cause of this is that the user id in the incoming message does not have a security profile on the receiving machine. For Windows it will be the logged on user id. Therefore, the receiving Queue Manager, be it IBM i, z/OS, AIX or whatever, must have a security profile for this user id. Alternatively, Channel exits can be used to change the user id as the message is being transferred.

27.1.4. Message "MO71 configuration file version is newer than program"

This message is issued if MO71 reads the MQMON.CFG configuration file and discovers that the file was written by a later version of MO71 than the currently running one. MO71 prevents you running in this situation to avoid corrupting the configuration file, for while MO71 is backward compatible (it can read any old version of the configuration file), it can not hope to make sense of a future version.

The user response to this situation would depend on what was intended.

- The most likely cause of this error is that the user has accidentally run an earlier version of MO71 and intended to run a newer version. As a general rule we would always recommend that users actually run the latest version available.
- It is just possible that, for some reason, perhaps to avoid a bug in the latest version, the user has decide to revert back to an earlier version of MO71 while the problem is fixed. In such a case then we would recommend that you re-instate the configuration file that (hopefully) you backed up when you installed the new version of MO71.
- Later versions of MO71 will automatically take a backup of the configuration file if they see you running a later version of MO71. By default this will write the backup to a file such as MQMON_930.CFG where the version is the version of the configuration file before it was migrated. So, if you have no manually backed up version of the MO71 configuration file then perhaps this automated backup has what you need.
- As a last resort you can manually change the configuration file to modify the version. Make sure you do take a backup of the configuration before you attempt this. In the first couple of lines you should see a statement like **ver=930.** Change the number to reflect that of the version of MO71 you are running and MO71 should start. Bear in mind that you are still loading a newer configuration file and therefore no guarantees are made that all data will load correctly.

27.1.5. Message “Problem not checked because of incomplete definitions”

This message is shown in the verify window of the Network View if MO71 does not have enough object definitions, for the given Queue Manager, to perform the health check for that Queue Manager. This can happen for a number of reasons. For example the user does not have sufficient authority to view any, say, channels. Another reason might be that, through Usage Tailoring, MO71 has been configured to not retrieve a set of objects. To fix the problem ensure that MO71 is 'allowed' to retrieve all of the verify object types and refresh the data to cause the health check to re-analyse.

27.1.6. An SSL/TLS connection fails and MO71 shows a Message Box

MO71 can be configured to have different SSL/TLS key repositories for different queue manager connections. This configuration can only be successfully used if these connections are not active at the same time. An IBM MQ client process can only use one SSL/TLS key repository at once. Note that this is a restriction in IBM MQ itself, not of MO71.

As a diagnostic aid, if the MQCONNX call returns MQRC_SSL_INITIALIZATION_ERROR, MO71 will check to see if you have multiple SSL/TLS key repositories configured, and whether a different one is currently active, and if so, will warn you with the message box you saw.

As an immediate fix, disconnect from all your other SSL/TLS protected connections (you can use Action->Disconnect All Shown to do this quickly). If your new connection now succeeds, this confirms that the problem was that MQ was not using your specified SSL/TLS key repository because there was already one in use. If your connection still does not succeed, diagnose the SSL/TLS failure using standard MQ diagnostics, including the AMQERR01.LOG on both the client and queue manager machines.

In the longer term, there are multiple solutions to the problem.

1. **Combine the key repositories**
   There is often no need to have separate key repositories just because the certificates are different. Multiple personal certificates can be hosted in a single key repository and given identifying labels. These labels can then be provided in your client channel definition in the MO71 location. A word of caution with this solution however. There may be good reasons to keep your certificates separate in different repositories. For example you may have some production keys and some for your test environments. In such cases it is often wiser to keep those environments entirely separate and thus we would recommend you use the next solution in this list.
2. **Run more than one MO71 instance**
   To keep your production and other environments separate and avoid any accidental changes, it is wise to have separate MO71 instances for each. This can mean having a separate key repository for production, different confirmation settings for the production environment, and even a different colour scheme to ensure it is obvious when you are connected to and interacting with, the production environment.
3. **Use conflicting connections independently**
   If you must continue to use multiple key repositories in the same MO71 instance then you need to ensure that only one key repository is in use at any one time. When connecting to a queue manager that uses a different key repository first you just need to disconnect from all your other SSL/TLS protected connections (you can use Action->Disconnect All Shown to do this quickly). Once the first key repository is no longer in use, connections using the new key repository should succeed.
4. **Utilise IBM MQ feature “Environment Scope”**
   IBM MQ V9.2.5 CD introduced an mqclient.ini file SSL stanza attribute called “EnvironmentScope” (read here for more information). When set this to CONNECTION it causes the MQ client to make a copy of the key repository for EVERY connection. This removes the described problem above but with an as yet unmeasured impact to CPU and memory resources. This may be appropriate for small numbers of queue manager connections or key repositories containing a few certificates. We don't know the full impact of this option and so can not recommend this for large key repositories and/or large numbers of concurrent queue manager connections and instead refer you to the first two solutions in this list.

27.2. Troubleshooting Web Access

Setting up the MO71 web access feature (see Chapter 20 Web Access on page 153) includes a different set of possible problems. We capture any common issues that you might come across here.

27.2.1. Twisties not working

The left hand pane of the main MO71 web display shows the list of queue managers that are web enabled as twisties. Each twistie can be opened to show a list of links to the various object type displays. Prior to version 9.2.0 of MO71, these twisties were implemented using JavaScript, but in V9.2.0 the reliance on JavaScript was removed when these were changed to use the HTML5 `<detail>` and `<summary>` tags.

If you are using a browser that does not support these tags, such as Internet Explorer or older versions of the Microsoft Edge browser, there are a few options for you.

- We recommend that you upgrade your browser to one that does support these tags. Most modern browsers support them by now, including the Chromium-based version of Microsoft Edge (version 79) available since the beginning of 2020.
- Add some JavaScript that handles the `<detail>` and `<summary>` tags. For example details-polyfill by Rico Sta. Cruz on GitHub, or one of the alternatives that are also listed on that page. MQGem Software does not guarantee the behaviour of these solutions, but provides this information to you simply as a convenience.

  The ***index.js*** supplied from that GitHub project can be very easily added to the ***mqadminnav.html*** file, by adding the following line into the `<head>` section of the HTML (Note the slash character is required before the file name).

  ```
  <script src="/index.js"></script>
  ```

- If the above options don't work for you, it is possible to revert MO71's web server to the pre-V9.2.0 behaviour which didn't use the HTML5 `<detail>` and `<summary>` tags and instead generated HTML that used the supplied inline JavaScript to allow the twisties to open and close when clicked on. To tell the MO71 web server to run in this way, set the environment variable `MO71_JS_TWISTY` in your environment to anything at all.

  The required inline JavaScript is still delivered in the MO71 zip file as an *.***html.old** file in case you require it. If you have customised your HTML files at all, copy the inline JavaScript from the old file into your current HTML files. If you are using the HTML files exactly as supplied and have not edited them at all, you can simply rename the old file.

27.2.2. Navigation twisty does not surround all its queue managers

If you find that your navigation group twisty does not surround all the queue managers it is supposed to contain, that is some of the queue managers are outside of the twisty and thus still visible when the twisty is closed, check whether you are making use of a file name to provide the links inside the twisty. A simple way to generate a set of links might be to view the HTML source generated by MO71 using the default list, and then copy that snippet of HTML into your file. Make sure if you do this that you do not end up with a trailing `</details>` tag in your file that will close the twisty too early.

27.2.3. HTTPS setup

We recommend when setting up your HTTPS web browser, you keep the preferences dialog open and manually Start the HTTPS server the first time. This way, if any configuration is not correct, a status message will be written to the bottom of the dialog to aid in troubleshooting.

For example, if the 'Search String' you provide to locate the certificate finds nothing, you will see the following in the status bar:

27.2.4. HTTPS connectivity

Make sure that you explicitly use the `https` protocol in your URL to connect to the MO71 web server when running HTTPS. If you simply type the address without the protocol, your browser will likely assume HTTP and try to use port 80. There will be no listener using port 80 if you have enabled HTTPS. It will be listening on port 443 (by default). There is no protocol forwarder from HTTP to HTTPS built into the MO71 web server.

27.2.5. sChannel function 'AcquireCredentialsHandle' failed RC(0x00000057) The parameter is incorrect

If you see the above error when starting your HTTPS listener (it will be reported in the status bar of the preferences dialog), then you may be suffering from an access problem with your certificate store. This generic error could be reporting other things too, but we know that it is used for reporting this access problem. Please see https://mqgem.wordpress.com/2023/09/01/mo71-https-localmachine-store/ for full details and instructions on how to rectify the issue.

27.3. Service

As with any software product of any complexity it always possible that there are bugs in the code. If you notice strange behaviour or the program crashes then by all means raise a bug report by sending an email to support@mqgem.com. Before you do this though please read the relevant part of this manual. Most of the 'problems' reported are user errors and have arisen through unfamiliarity of that part of the product. The next thing to ensure is that you are using the latest version of the software. The latest version is always available for download here https://www.mqgem.com/mo71_download.html Provided you have a current licence you can upgrade to the latest version of the software free of charge.

If upgrading doesn't help and you are certain it is a product bug then please do send an email to support@mqgem.com Please be as specific as possible about the problem you are having and how the problem can be recreated. Screen-shots of the problem often help enormously. Remember that the more detail you put in to your problem description then the faster your problem can be resolved.

27.3.1. Mini-Dump

If the problem is a program crash then it is possible to get MO71 to generate what is known as a mini-dump. A mini-dump is a small file created at the time of the crash which shows service what the program was doing at the time of the failure. Before generating a dump file ensure that you are using the very latest version of MO71. This is for two reasons. Firstly, it is quite possible that any bug you have found has already been fixed. Secondly a dump file contains lots of offsets, such as the offsets to program statements and variables. These offsets must match the latest version of the source code if the dump file is to be analysed. If the program version is too old then the dump file will be of no use and you will need to generate another more recent one.

To get MO71 to catch exceptions and generate a dump file you need to start MO71 with a '-E' parameter. For example:

```
mqmonntp -E c:\mo71dumps
```

The -E parameter is followed by the directory where you would like the dump files written. The files will have a name something like **mqmonntp20120416_20141029_092103_114692.dmp**. The parts of the file name are the name of the program, the program build date, the current date and timestamp and lastly the dumping threadid. If the file is not generated then ensure that you have spelled the directory name correctly and that the directory exists and is writeable.

There are actually two types of mini-dump. The default mini-dump file contains only stack storage but is good enough to solve many problems since it shows where the program was and what it was doing at the time of the dump. However, it doesn't contained any heap storage which is why the file is only around 50K. There are times though where a dump file containing the heap storage is needed. The files are often typically 100MB in size. To obtain a full dump file you need to add the parameter **-F full** to MO71. With this parameter all dump files generated, by any means, will also contain the heap storage. Please do not send full dumps by email.

At any time, even if you don't use the -E parameter, you can ask MO71 to generate a dump file by pressing <ALT>,<SHIFT> and '1' keys all at the same time. If the -E parameter is not specified the file will be written to the same directory as the program.

Generating dump files via the Web interface is also possible. To do this you need only to connect your browser to the mini-dump address. Something like : http://localhost/minidump/?code=secretcode. To allow the browse to generate mini-dump files you must first ensure that 'Allow Mini-dump Generation' is enabled in the HTTP tab in the Preferences Dialog. The browser must also specify the same code as configured in the 'Dump Code' field. This prevents an unauthorised browser from generating dump files on your server. However, if you wish not to define a 'Dump Code' then you can leave the value blank.

Generally speaking you would create a dump file only when directed by service. However, if you have a crash then by all means send the dump file along with your problem report. Remember though that a dump file is not a replacement for good problem text. It can show what the program is doing at that instant but not how it got there. It is just a tool which **can** help. When you send the dump to us please try and include as much additional information as possible. Remember that the more information you give the faster we are likely to be able to help you. The type of information you could include are:

- What you were doing at the time? Is it only when you do this action ?
- Consistency, does it always happen or only occasionally.
- What are the MQ versions that you are using ? – both MQ client and MQ server if they are both involved.
- What changed to make this happen ? Or has it always happened ? Or only since you installed a new version of MQ, or MO71 or a Windows SupportPac etc
- What platform are you doing this on ? Window 8.1, Windows 10, Linux ?
- Have you any idea what else MO71 may be doing at the time ?

The ideal situation for us is that you give us enough information for us to be able to recreate the problem here. Once we've done that then it is usually pretty simple for us to give you a fix.

Chapter 28. Changes made in previous versions

This chapter will give you an idea of the changes that have been made if you have used a previous version.

Changes made in Version 9.3.2

1. **MO71 converted to a 64-bit program**
   IBM MQ have announced that they intend to drop support for 32-bit applications. While they haven't said exactly when this will happen they have indicated that it may be fairly soon. So, the sole reason for this release is to convert MO71 to a 64-bit program and ensure that it is runnable in all environments. Clearly this means that there is a migration step to ensure that MO71 now has access to the 64-bit rather than 32-bit IBM MQ product libraries.

   Of course anyone who has ever ported an application, of any complexity, from 32-bit to 64-bit will know that it is a lot more involved than just recompiling with a new set of options. There can be lots of unintended consequences that are not easily predictable. For this reason we re keeping other changes to a minimum. There are very few actual functional changes in MO71 9.3.2 compared with MO71 9.3.1. The focus is to ensure that the program operates, at least externally, exactly like it's predecessor.

Changes made in Version 9.3.1

1. **Graphing**
   There have been a number of enhancements to the Graphing capabilities. In brief these are:
   - MQEV and Monitor graph profiles can now be saved and restored
   - New types of graph line types, for example 3D columns
   - Configurable legends
   - Greater colour flexibility, axes, labels etc
   - Log Y scale
   - Pie chart 'Other' threshold
   - Graphing preferences in Prefer Dialog
   - Sample Graph to enable easy playing with the options

   For more information please see Chapter 14.Graphing on page 114.

2. **Activity Trace**
   Various additions have made to activity trace to make analysis of the data much easier
   - New tab added to allow the graphing of activity trace data
   - Next, Previous and Matching menus options to allow easy traversal of output with matching values
   - History of visited records in the output allowing quick and easy return to prior records of interest.
   - Ability to jump to the 'linked' MQI call.
   - Ability to set an explicit bookmark in the output

   For more information please see Chapter 23.Activity Trace on page 175.

3. **Graphing of dialog list fields**
   Any MO71 dialog list can now be instantly turned into an informative pie chart merely by selecting the required field and selecting 'pie chart' from the context menu.
   This can also be achieved by using the new **piechart()** function.

4. **Concept of 'Connection Owner" added to Connection List dialog**
   A connection can be identified in many different ways – the application name, the CICS transaction, the IMS region etc. This field will coalesce these fields into a single field which is both more informative and more concise.

5. **New Predefined Dialog options**
   - Listeners and Listener Status
   - Services and Service Status

6. **Master Configuration file can now be in XML**
   Previously the master configuration file had to be an MO71 configuration file. In this version we now allow this to be an XML file which could be an MQ Explorer config file or an XML file that has been generated centrally.

7. **Location Objects Export will now create the directory path if required.**

8. **Export Type, Export Objects and Export Frequency added to the Multiple Location Change Dialog .**

9. **New 'Export all Shown' feature added to the Action menu.**

10. **New export feature “Blank Fields” added to Location export**
    In addition “Blank Fields” are now adhered to for CSV, Text, JSON and XML exports.

11. **Location export can now export Queue Manager ancillary data**
    The export location can now export the Queue Manager status/Log/Archive/System information where appropriate.

Changes made in Version 9.3.0

1. **MQ Command Level 930 and 931 Support**

2. **KDB Password prompting**
   Have MO71 prompt you for the password to open the Key Database file when it is needed rather than storing it anywhere. For more details see 24.5. Key Database File Password on page 225.

3. **Export MQSC commands in a single line**
   A new option '**One Line**' is available in the Export Dialog when exporting MQSC from a list or single object dialog. This option will cause the exported MQSC for each object to be output in a single line. Each MQSC attribute is given it's own column. This makes it very much easier to compare the values specified between different object since they are lined up above one another.

4. **Activity Trace improvements**
   When filtering the Activity Trace messages you can now use wild-carded object names. In addition you can ask for the inversion of the search which means that queues can now be excluded from the results. For more information please refer to Section 23.3.2 Filter on page 177 which describes the filtering one can do with the Activity Trace Dialog.

5. **Trace Message Diagram improvements**
   The Trace Message Diagram now allows the various text labels to be copied to the clipboard using the right-mouse click context menu. Additionally, the channel names, queue names and queue manager names shown on the diagram can be used to open the referenced object in a dialog from the same context menu. For more information see Chapter 19. Trace Message on page 148.

Changes made in Version 9.2.3

1. **Object resolution for queues → storage classes → page sets → buffer pools and queues → process objects**
   Extending the object resolution currently in MO71 of channels and transmission queues, now various other objects are resolved and have their definitions checked.

2. **Title information can be configure for Storage Classes, Pagesets and Bufferpools**
   This allows information about the relative usage of each object in the verify window. Please see Media Titles in the Network UI tab of the Preferences Dialog for more information.

3. **Problem Selection dialog added**
   Making it much easier to examine all the problems that MO71 will check your queue managers for, and to deselect any that you do not wish to use. Full problem descriptions provided with suggested resolutions; counts of problem instances found; and a display of all the instances of an individual problem on all queue managers checked.

4. **Extended list of configuration problems checked for**
   Several additional configuration problems are now checked for including; missing or incorrectly defined system objects; resolution issues with process objects, storage classes, page sets and buffer pools; backout queues; topic object model queues and more.

5. **DISPLAY ARCHIVE / LOG / SYSTEM added to queue manager dialog and multi-queue manager list**
   See these other configuration attributes of your z/OS queue manager, and view these attributes in a multi-queue manager list dialog making it easy to sort or compare the values used across several queue managers. Not all fields from the DISPLAY commands are able to be SET, so you will see some fields are shown as read-only. More details about which fields are able to be SET can be found in IBM Docs about these three commands. This can be disabled if not required, using the Location dialog, Options tab checkbox “SYSP Commands”. See 24.3.4. Options on page 216.

6. **SET ARCHIVE / LOG / SYSTEM added to queue manager dialog context menu**
   Make changes to these other configuration attributes of your z/OS queue manager, using Set commands. These commands can be removed from the context menu using Usage Tailoring. See 25.4. Miscellaneous Tailoring Keywords on page 240.

7. **Reduced start-up time for the API Exerciser dialog**

8. **You can now export the dialog contents in JSON format**

9. **Automatic export allows choice of QSGDISP for export of z/OS queue manager objects**
   Choose to export QSGDISP values of SHARED, GROUP and/or COPY, in addition to all QSGDISP(QMGR) objects for queue managers that are a member of a Queue Sharing Group (QSG). This is a change in behaviour. Please read 29 Migration from previous versions on page 261

10. **Initial focus field in object dialogs**
    When starting an object dialog from a list dialog (for example double-clicking on a Queue) then the field that is clicked on is the field that is in focus when the single object dialog is displayed.

11. A new Predefined Dialogs field “Context Available” which controls whether a Group context command is shown only when all group members are applicable or any of them.

Changes made in Version 9.2.2

1. **Improvements to Activity Trace feature**
   - You can now filter by multiple reason codes
   - You can exclude, rather than just include, particular reason codes
   - A Context Menu has been added allowing easy access to actions like “copy to clipboard”
   - An export dialog has been added allowing easy export of output data to a file.

2. **Addition of HTTPS to Web Access**
   It is now possible to run the built-in web server in MO71 using the HTTPS protocol. See 20.5 Using HTTPS on page 163.

3. **User Agent field added to HTTP Connections List**
   The HTTP Connection List, in addition to being enhanced to include HTTPS information, also shows the Browser User Agent string. See 20.7 Who is connected ? on page 165.

4. **Support for DISPLAY CFSTATUS added**
   New dialogs for CF Struct Status and the CF Struct Status list have been added.

5. **New options to automatically fold Userid and Password fields to uppercase**
   See 24.3.2 Security tab for the location dialog on page 215.

6. **Allow export of completely blank columns to CSV files**
   When exporting a list dialog to a CSV file a new option allows you to request that completely blank columns, which would normally be excluding from display and from export should be included in the export to the CSV file. See 11.3 Exporting Definitions on page 101.

7. **Add Authorisation Record support to HTTP *%qmlist* keyword**
   Please see 20.4.1.1. %[qmlist] HTML Insert on page 160 for more information.

8. **More customisable navigation links for Web Access**
   You can now have any links you wish in the navigation section of MO71 Web Access. See 20.4.1.1. %[qmlist] HTML Insert on page 160 for all the details.

9. **IBM MQ Command Level 924 and 925 supported**

Changes made in Version 9.2.1

1. **Addition of the concept of Security Groups**
   It is now possible to identify groups of locations as belonging to the same security 'group'. This means that each member of the group will use the same logon userid/password. For more information, please see the description of the Location dialog Security tab on page 215.

2. **Ability to make multiple changes to your locations in a single click**
   To make it easier to configure and organise your locations you can now select a group of locations and change a location settings, say the Network Names field, in one go. For more information please see Chapter 24.11 Multiple Location Change Dialog on page 230.

3. **Ability to specify whether the location is 'MVS' when importing from a CCDT**
   This applies to both the standard binary CCDT and the JSON CCDT.
4. **Reduced memory usage for multi-instance Queue Manager lists**
5. **Usage tailoring now applies to Web Access (Browser displays)**
   It is now possible to prevent users from displaying certain types of objects over the HTTP interface. Please see Chapter 20.6 Limiting what is displayed on page 165 for more information.
6. **MO71 will now check whether command responses from command server are complete**
   Essentially MO71 will check that the number of message responses match the Message Sequence Count in the last response message.
7. **MO71 will now warn the user if the number of list objects is of comparable size to the MAXDEPTH of the Reply Queue**
   This can prevent problem where the Command Server sees a full reply queue and starts putting messages to the Dead Letter Queue.
8. **Support for IBM Command Level 9.2.1**

Changes made in Version 9.2.0

1. **MQEV graphing support**
   MQEV data can be accessed directly from the graphing dialog.
2. **Improvements to graph output display**
   New features added to graphing such as :

   - Column Stacks
   - Long and Short Average lines
   - Saving of graphing defaults
   - Saving of line defaults
   - Drag resize
   - Improved time labels added to graphs.
   - Auto-range of axes
   - Column relative sizing and overlap
   - Pie chart labels

3. **Improvements to HTTP displays.**
   - A whole variety of minor improvements as well as adding MQEV displays
   - Conversion to HTML 5 so that twisties are not reliant on JavaScript, and instead make use of the `<details>` and `<summary>` tags.
   - Add “all”, “allmq” and “allev” to qmlist[]

4. **Import from JSON CCDT file**
5. **Import now allows you to specify Network Name values**
6. **Fields added to Queue Manager list dialog**

   - Group name
   - Network Names
   - Connection Name
   - Channel Name
   - CipherSpec
   - UserId

7. **Addition of STOP EV command**
8. **Trace message pie chart labels has been added**
9. **Double click on main window group will expand/contract**
10. **Menu *“View - Set Default Lists”* now groups dialogs into relevant categories.**
11. **Compare 'Services' added**

Changes made in Version 9.1.4

1. **MQEV command support**
   MO71 now understands MQEV commands. Please see Chapter 15 MQEV Event Monitoring on page 123.
2. **New List Title Option**
   When automatically sizing list columns, there are now two options, one to prefer longer list titles (and thus wider columns) and one to prefer shorter list titles resulting in less extraneous space in columns. Please see 4.1.5 List Windows on page 12.
3. **Command Level 914 support**
4. **Mouse Hover over truncated list data will show full text in a side window**
   If a list dialog column is too narrow to display the entire column data then the data will end with an ellipsis (…) indicating that there is more data to display. If the user hovers the mouse over the '…' characters then a side window will be displayed showing the full text.
5. **New option to display the MQSC name after the field title in object dialogs**
   In the Dialogs tab of the Preferences Dialog a new option 'Include MQSC name in field title' allows you to show the MQSC name alongside the field title.
6. **Support for AMQP Channel Status added**
   New dialogs for AMQP Channel Status and the Channel Status list have been added. If you don't use AMQP channels then these dialogs can be removed from the menu by unchecking the option in the location dialog in the Options tab.

Changes made in Version 9.1.3

1. **IBM MQ 9.1.3 Support**
   The following changes have been made:
   - Support for the new Application Status command
   - SPL Protection Channel Attribute added for z/OS channels
   - API Exerciser has been changed to allow specification of MQCNO_GENERATE_CONN_TAG
2. **New 'Changed Background' colour**
   A new colour has been added for highlighting differences when using the compare dialogs.
3. **Change to the way 'Refresh Time' is displayed on titlebar**
4. **Structure Version dropdowns added to API Exerciser**
5. **Ability to remove filters from the filter history.**
   Press Ctrl-Delete while text is in the entry field.

Changes made in Version 9.1.2

1. **Date and Time Fields**
   MO71 now allows you to display Date and Time fields in a single fields. This makes lists easier to sort and allows the user to choose their own date format. In addition you can configure a Queue Manager time-zone meaning that the time from a Queue Manager can be converted to local time making it far easier to understand.
   For more information please see Chapter 11.1 Dates and Times on page 98.
2. **Units**
   Many of MQ attributes have an associate unit – be it Bytes, Seconds, Megabytes etc. MO71 now understand this association and will let you display these values in a more natural fashion as well as input the data using the units.
   For more information please see Chapter 11.2 Units on page 99.
3. **RPRODUCT conversion**
   MO71 will now convert the RPRODUCT identifier into a more meaningful product name.
4. **Filters can now use AND, OR and NOT operators**
   For those of you not keen on the &, | and ! Symbol operators.
5. **New option on list dialogs to hide/show Column Filters**
6. **New actions in location dialog**
   You can now, 'Connect' and 'Disconnect' and view your error log and INI files direct from the location dialog.

7. **Main window group text**
   You can now configure MO71 to display status information such as how many Queue Managers are in a group and how many of them are connected.
8. **Download object status**
   MO71 will now display the times that the objects and default objects were last downloaded.
9. **New export options**
   You can now request that the dialog fields and current filter is exported along with your data.
10. **New usrcol() options**
    New options to average the values as well as options for indicating the value is a rate or has an associated unit such as Byte.
11. **Allow the user to contact MO71 with special User Events**
    This allows you to add and remove entries on the MO71 console from another program. For more information please see Chapter 16.4 User Events on page 130.

Changes made in Version 9.1.1

1. **Search Fields added to Alter List Dialog**
   Some dialog lists have large number of fields and it can be hard to find the particular field you are looking for. Search fields have been added which filter the list content making the target field much easier to find.
2. **Multiple fields can be moved in the Alter List Dialog**
   When re-ordering the list fields by moving them up and down you used only be able to do this one field at a time. You can now select as many fields as you wish and move them up and down together.
3. **3 Monitor support**
   MO71 will now remember the dialog positions and sizes for a 1, 2 or 3 monitor set-up.
4. **Command Level 911 and 912 support**
5. **HTTP Proxy Support**
   You can now configure MO71 to make HTTP connections via an HTTP Proxy. Please see the HTTP settings of the Preferences Dialog on page 210 for more information.
6. **Progress Dialogs**
   MO71 will now report on its progress as it is moving, copying or deleting large numbers of messages. For more information please refer to section 7.3. Progress Dialog on page 73.
7. **Confirmation of leading/trailing spaces**
   Warn user if leading or trailing spaces are present in Topic Strings at creation.
8. **A health check for topics containing topic strings with either leading or trailing spaces**
9. **New function exists()**
   This simple function will return TRUE if the field exists. For example exists(curdepth) will return TRUE only for local queues.
10. **New function usrcol()**

    Please see Function - usrcol(Name, Value, Options, MaxValue, Colour1, Colour2, FgColour, BgColour) on page 44 for a description of the new function.

11. **hidecol can now take a special value 'all'**
    hidecol(all) will hide all columns. Allowing you to use showcol() to display just the columns you want.

12. **New user defined colours**

13. **New colour constants and the ability to use colours of the current colour scheme**
    It is now easier to write filters which blend in with the current colour scheme of the application.

Changes made in Version 9.1.0

1. **Command Level 910 recognised**

   IBM MQ has increased in version to MQ 9.1.0 and has therefore introduced a new command level 910. The MO71 version policy is always to have the same two digits as the version of IBM MQ is supports. As such it is necessary to introduce this new

version of MO71 even though command level 910 introduces no actual command changes over and above command level 905. For MQ users on the LTS Release though you will see quite a number of changes between 900 and 910.

Changes made in Version 9.0.6

1. **Column Filters additions**

   Version 9.0.5 introduces the notion of column filters. In this release additions have been made to Column Filters to improve their usefulness. In particular integer fields are now available in the selection list making it easy to see when you have a few distinct values. For more information please see Chapter 5.1 Column Filters on page 28.

2. **Splash Screens**
   It is now possible to configure MO71 to display a splash screen upon start-up and when connecting to a Queue Manager for the first time. For more information please see Chapter 11.5 Splash Screens on page 103.

3. **MO71 Program update notification**
   It is always recommended that you run the latest version of MO71 and get the benefit of new features. To make this easier MO71 will now check whether there is a new version and display a dialog to you.

4. **IBM MQ Maintenance notification**
   When managing an MQ estate it can be difficult to keep track of the IBM MQ Maintenance level availability, such as fix packs, and which Queue Managers they have been installed on. MO71 now makes this easy by displaying this information in the Queue Manager List Dialog, the main Window and the Network view. For more information please see Chapter 11.4 MQ Maintenance on page 103.

5. **Import from another MO71 Configuration File**
   MO71 now supports migrating location definition from another MO71 configuration file. Please see Import Locations on page 187.

6. **Master Configuration File**
   MO71 now has the notion of a master configuration file. This is passed as a parameter on start of the program and causes any locations definitions to be copied from that configuration file in the current configuration. Please see Chapter 11.6 Master Configuration File on page 105 for more information.

7. **Predefined Dialogs in Context menu**
   When a Predefined Dialog is defined you can now set 'Context Command'. By setting this value invocation of the predefined dialog becomes possible from the context menu shown when you right-click on a Queue Manager.

8. **Licence Command**
   MO71 can now be configured to run an OS command when the licence file is either not present or has less than 30 days to run. For more information please see Chapter 11.7 Licence Command on page 106.

9. **List dialog authority failure count**
   You can now ask MO71 to report the number of objects not displayed in a list due to having insufficient authority. This is, of course, subject to getting 'Not Authorised' responses from the Command Server.

Changes made in Version 9.0.5

1. **Column Filters**

   This release introduces the concept of column filters. MO71 has always had very flexible filtering using the list filter field. However, this can appear complicated to the novice user. Column filters provide a quick and simple way of filtering by the value in a particular column of a list. For more information please see Chapter 5.1 Column Filters on page 28.

Changes made in Version 9.0.4

1. **Activity Trace Viewer**

   If correctly configured IBM MQ will write Activity Trace messages to a queue which indicate what is currently happening on your Queue Manager. MO71 will now process these messages allowing you easily see the activity and filter the results.

2. **Import from MQ Explorer export file**
   It is now possible to take the XML file generated from an MQ Explorer export and import the location definitions in to MO71. This makes it quicker to get started in MO71 if you have previously been using MQ Explorer. For more information please see “From MQ Explorer File...“ on page 188.

3. **Text Copy Supported in containers.**
   It is now possible to select text in containers and press Ctrl-C to have that text copied to the clipboard. Please see “Container Windows“ on page 226 for a more complete description of container windows.

Changes made in Version 9.0.3

1. **Export of AUTHREC in MQSC format**

   This was one of the most commonly requested features. You can now export AUTHREC definitions in MQSC format from the object and list dialogs as well as include AUTHRECs in the automatic Queue Manager export list.

2. **API Exerciser enhancements**
   We are pleased to say that this version allows you to experiment with the MQCONNX, Message Handle and Message Property API calls.

3. **Hostname display**
   When displaying a Channel Status output, MQ will send you the IP address eg.198.51.100.34. However, this is not hugely memorable. You can now ask MO71 to issue a gethostbyaddr() and display the result in a hostname field. You can go even further and put your own description of this in a dns file, MQMON.DNS, and have MO71 display that instead. See “Hostname Support“ for more information.

4. **User Commands**
   You can define your own commands which are then available in the Queue Manager context menu. See “Chapter 22. User Commands” for more information.

5. **Can now invoke the MQEdit Message Editor directly**
   When you are browsing a message you can select 'Edit...' from the context menu and the MQEdit product will be started editing the same message.

6. **New Default Max. Column preference option**
   Some MQ fields can now be quite large, this can mean that lists of objects can have columns which are wider than you would like. You can now say how wide the column should be by default.

7. **New Max Column and Field lengths for HTTP browser output**
   As above these large MQ fields can mean that the browser output is hard to read or it results in an output with a large horizontal scrollbar. You can now tell MO71 how much data to return in these situations.

8. **Message Operations are now Message Token based**
   By basing message operations on Message Tokens the message operations such as browse, copy, mode and delete are much faster and more specific.

9. **Set of CipherSpec considered weak is increased**
   By default the set of CipherSpecs shown when creating a channel definition is the set advised by IBM. Those that IBM consider deprecated at not shown by default. This deprecated set has recently been increased by IBM, and MO71 has been updated to match.

10. **Horizontal scrollbar added to MQSC window**

11. **Support for MQ Command Level 902**
    The changes in command level 902 were mainly on the queue manager object for the new features Automatic management of linear log extents, and Automatic writing of media images.

12. **Search path for error log and ini file include re-distributable client locations**
    When a re-distributable client is used instead of an installed image of MQ, the data path where the error log and ini files are stored is different. MO71 will check for these if other locations are not found.

13. **Problem detection**
    A few additional problems have been added to the problem detection list. See Appendix B: Constants on page 270 for the current complete list.

Changes made in Version 9.0.2

1. **Connection Host list support**
   If a Queue Manager can be hosted on multiple servers you could always specify a list of host addresses. However, since this list was passed to MQ directly MO71 did not know which server was hosting the Queue Manager. In this version MO71 will process the list directly, one by one, and therefore know which host makes the successful connect. You can configure a text string following the host address which will be displayed on the main window and in the network view.

2. **Message summary of Dead Letter Queue Headers**
   The summary display of MQDLH has changed to make the target Queue and Queue Manager more likely to be shown.

3. **Linux Support**
   Changes were made so that MO71 could now run on Linux under WINE.

4. **Minor bug fixes**
   - Highlighting different fields in comparison objects was not correct if the two objects were mutually immutable.
   - Graph pie chart key would sometimes not paint

Changes made in Version 9.0.1

1. **The Search field in the main window is now a combo-box**
   MO71 will now remember the last 20 search strings used. The most recently used items are shown at the top of the list.

2. **New Search field shown above the Queue Managers in the Multi-QM pane**
   The search field supports the same syntax as the main window search field. So, it is possible to search on Queue Manager name, location name, group and network settings.

3. **New select toolbar**
   To the right of the search field are three tools in a toolbar. These allow you to:
   - **Add all the shown Queue Managers to the list data**
   - **Remove all the shown Queue Managers from the list data**
   - **Show only those Queue Managers in the list data**

4. **Two new preference options have been add to the 'Lists' tab**

   These are:
   - **Copy main Queue Manager Search field**
     With this option checked the current content of the Search field on the main window will be copied to any newly started lists. This means that if you have narrowed the list of Queue Managers shown in the main window this limited list will also be presented in the Queue Manager list in the list dialog.

   - **Queue Manager Search add without confirmation limit**
     If you use the tools to the right of the list search field you could select a large number of Queue Managers. This may cause MO71 to connect and retrieve data from all these Queue Managers. If it is only a few Queue Managers then the workload is not great. However, suppose it is 100's of Queue Managers? This option allows you to set the point at which MO71 will ask for confirmation that this is really what you want to do.

5. **A few minor bug fixes**

   - **Right justified number columns which overwrite previous column if the column is made too narrow**
   - **Database icon in the Queue Manager list can show full incorrectly for the primary Queue Manager**
   - **Reduction to the flickering seen when the main window is resized**

Changes made in Version 9.0.0

1. **Support for MQ Command Level 900**
   There weren't a great deal of changes to command level 900.

2. **Ability to define, update and delete Protection Policies**
   Other aspects of MO71 have been updated accordingly. For example updates to Default Lists and Usage Tailoring.

3. **Support of CCDT URL**
   MQ V9 allows a connecting application to specify the URL location of the CCDT file to use. This field can be specified in the location dialog.

4. **Show last monitor time on main window**
   If you wish you can display the last time (or elapsed time) that a response was received from a Queue Manager that is monitored.

5. **Changing multiple objects can now belong to different Queue Managers**
   If multiple objects are selected from a list and those objects belong to different Queue Managers then the resulting dialog will qualify each object with their owning Queue Manager. It is now possible, for example, to change the maximum depth of multiple queues on multiple Queue Managers with a single click. Or alternatively you can stop/start multiple channels on different Queue Managers with a single press of a button.

6. **Changing multiple objects is reflected in the dialog title bar**
   The text [Multiple Objects] will be shown in the title bar in this case.

7. **Exporting lists now has a new 'All Fields' checkbox**
   Previously, when exporting from a list of objects, only the fields which were actually in the list would be exported. This is useful for certain types of displays, such as text and CSV. However, for MQSC is can mean that you would get an incomplete object definition. A new 'All fields' checkbox now allows you to request that all the object fields are exported.

8. **New Buffer Pool and Page Set dialogs displayable**
   MO71 now supports the display of z/OS Buffer Pool and Page Set objects.

Changes made in Version 8.0.4

1. **Main window Queue Manager search field**
   To make it easier to find Queue Managers in the list you can now enter a search field which will limit the contents of the list to only those locations which contain the search text.
   For more information please see the description of the View menu on page 190.

2. **Connect All Support**
   You can now ask MO71 to connect to all the displayed Queue Managers.
   For more information please see the description of the Action on page 189.

3. **Disconnect All Support**
   You can now ask MO71 to disconnect from all the displayed Queue Managers.
   For more information please see the description of the Action on page 189.

4. **Channel Mask added to Application Definition**
   Application monitoring uses the Application Tag and Application Description fields to identify the applications. Unfortunately sometimes these values are not as distinct as one would like. This often happens in the Java environment where a number of different applications may be all called 'java.exe'. To improve the situation MO71 now allows you to associate particular channels with particular applications. Please see Chapter 10.3 Channel Mask Processing on page 92 for more information.

5. **'QM Group' in Console list**
   When viewing the console you can now filter the results by 'QM Group'. This means that you can effectively have multiple console displays each showing an area of your MQ network.

6. **Message browsing over the Web improvements**
   New HTML inserts %[curr] and %[currpos] have been added to allow you to construct an HTML template file which modifies

the way in which a message is displayed.

7. **HTML template specification in URLs**
   In any HTML link you can now include the HTML file name you want MO71 to use as a base for the response

8. **Option added to location definition to control whether browsing queues is allowed over the Web Interface.**
   Browsing is allowed by default but can be switched off if required.

9. **Improvements to the event message formatting**
   Both message browsing and console display of event messages have been improved.

10. **IBM MQ name change**
    WebSphere MQ has been renamed to IBM MQ. This has been reflected in the MO71 program and the documentation.

11. **Import from CCDT**
    It is now possible for MO71 to import location definitions from a Client Channel Definition Table (CCDT).

12. **Added new display types to HTTP qmlist insert**
    New types def, qsq, conh and cona

13. **Improved licence handling**
    Licence expiry and reminders are checked during the running of MO71. Updating the licence file can be done without ending the program. If running in background mode, reminder messages are written to background file.

Changes made in Version 8.0.3

1. **Support for IBM MQ 801 & 802 Command level**
   IBM introduced some command changes in Fix Packs 8.0.0.2 and 8.0.0.3. The changes in these command levels are given in http://mqgem.wordpress.com/2015/04/13/whats-in-command-level-801/

2. **Support for the IBM MQ Appliance**
   This version includes command and attribute support for the IBM MQ Appliance.

3. **Object Naming Checking**
   Changes have been made to allow the user to ensure that objects are made with the correct name. In addition the health checking done in the Network View can be configured to check whether your main objects conform to your naming convention.

4. **Object comparison with name matching**
   For many releases MO71 has allowed you to compare objects between Queue Managers. However, the comparison only compared 'same named' objects. It is now possible to compare which have different names. For example comparing the definitions of objects Q1.TST and Q1.PRD on a test and production Queue Managers respectively.

5. **Trace Route Messaging**
   MO71 now allows you to use the trace message functionality in the MQ base product. By selecting a source Queue Manager and putting one or more trace messages to a target destination MO71 will show you the route the message took. For a description of this feature please see Trace Message on page 148.

6. **Web Access Improvements**
   The HTML files and the qmlist[] insert have been improved to display the MQ data in an iframe.
   qmlist[] can now filter by location or network name instead of just Queue Manager name

7. **Location Name**
   The length of the location name in a location definition has been increased from 49 bytes to 99 bytes.

8. **MQSC command available in the Commands context menu**
   To make it easier to access the MQSC dialog command has been added to the commands context menu.

9. **MQSC can now auto export multiple commands**
   Auto export was added to the MQSC dialog in the previous release but multiple commands were not supported. In this release

you can now issue multiple commands, such as reading the commands from a file, and have them auto-exported to a file.

10. **Support for DISPLAY SUB(*) DISTYPE**
    When you display the list of subscriptions you can decide whether the returned topic strings should contain the topic object stems or just contain the defined values.

11. **Queue load feature now supports explicit setting of Read-Ahead and Async-Put options**
    Previously MO71 would use the default setting for these values. This could be inconvenient if you want applications to use one setting and MO71 to use another. So, you can now explicitly set whether you want to set these values on. They are on by default since they can offer a significant performance advantage over slow client links and there is little real disadvantage in using the options.

Changes made in Version 8.0.2

1. **Usage Tailoring changes**
   A few new keywords are allowable in the AUT file to control how MO71 looks to users. Please see Chapter 25.Usage Tailoring on page 235 for details.

2. **Export Timestamp date format**
   You can now set the format of the date column in the auto-export timestamp. This is set in the Export tab of the Preferences Dialog on page 206.

3. **Auto Refresh Alignment**
   It is now possible to specify a dialog alignment time as well as a refresh interval. For example, you can now say that you wish to refresh the dialog every hour **on the hour**. Or you could perhaps refresh daily at 9am.

4. **Web Access configuration**
   Greater control of the web access behaviour has been provided in the preference dialog.

5. **Export file inserts**
   You can now use file inserts for a normal dialog export

6. **Web Page Class names**
   CSS class names of **mo71listtable** and **mo71objecttable** have been added to the generate HTML browser pages. This means you can now more easily change the look at feel of the generated data.

7. **Display location in multi-Queue Manager lists**
   It is now possible to display the location, as well as the Queue Manager name, in multi-Queue Manager lists. The location will also be exported in Text, CSV and XML mode. The location is also shown in a Queue Manager dialog.

8. **Attribute List format changed**
   To deal with the ever increasing combinations of fields valid for each Queue Manager version the way in which lists are stored has changed. Extensive testing has been done to try and ensure compatibility with previous versions however with changes of this nature it is always possible that there are some discrepancies. If you notice differences please let us know.

9. **New Preference option to set list flash interval**
   By using filter functions flash() and flashcell() you can ask for certain elements of a list display to flash periodically. By default the period is once a second. However, you can now set this interval to a longer value if required in the 'List' tab of the Preference Dialog.

10. **Export and Auto-Export available from MQSC Dialog**

11. **Added usage tailoring keywords for Pub/Sub**

12. **New filter functions**
    - sortd2() Set descending secondary sort order
    - showcol() Show a new column in a list display
    - hidecol() Hide a column in a list display
    - htmlfile() Allows you to specify the base HTML file for Web access

Changes made in Version 8.0.1

1. **Password Cache**

   MO71 now supports a password cache which means that it will store userid/password combinations for all locations and only require the user to make a single logon to the cache to have use of the passwords. For a description of this facility please see Password Cache File on page 222.

2. **Queue Manager List Dialog**
   It is now possible, using multi Queue Manager display, to show many Queue Manager definitions in a list. You can now, for example, display the product versions of all your Queue Managers in a single list or compare their tuning attributes. You can, of course, also use the powerful filters to make queries against the data.

Changes made in Version 8.0.0

1. **MQ Version 8.0 Support**
   Clearly one of the most important changes is to support the new objects, commands and attributes of MQ version 8.0.

2. **Application Monitoring and Viewing**
   For a complete description of this feature please refer to Application Monitoring on page 89. Essentially MO71 will now periodically check what applications are running and what objects they have opened. This application data can be used in a variety of ways.

   - It provides a quick way of viewing what is currently running.
   - A history of what an application does is built-up over time. For example, you can see which applications regularly use a particular queue.
   - Limits on the minimum and maximum connections by an application can be set. If these limits are exceeded then MO71 can notify the user.
   - The state of the applications can be displayed in a new application view diagram.

3. **Importing local Queue Managers**
   MO71 will now automatically create location definitions from the current local Queue Manager, either at start-up or on user request.

4. **Start-up Performance Improvement**
   Filling the objects in the main window could significantly slow down the start of the application. The objects are now loaded 'as required' when that section of the window is opened.

5. **New filter functions**

findstr()	A case sensitive search for a substring.
findstri()	A case insensitive search for a substring
substr$()	Returns the subtring of a string at the given length and offset.
where("*where expression"*)	Allows a WHERE clause to be specified on Queue Manager object lists

6. **%[qmlist] HTTP insert enhanced**
   Parameters have been added to HTTP %[qmlist] insert to allow specification of whether to use queue manager name, the location name and/or the Queue Manager group hierarchy. In addition you can specify which of the Queue Manager lists should be displayed in the location twisty.

7. **Share Filters**
   It is now possible to put shared filters in a common file and have multiple users of MO71 use the same filters. This makes maintenance of multiple machines far easier. Please see Shared Filters on page 53 for more information.

Changes made in Version 7.5.2

1. **TABs on objects**
   The Queue Manager, Queue, Channel and Channel Status dialogs can now be presented in a Tabbed form. This can make

interaction with these dialogs easer since they contain a lot of fields. Options are available in the “Preferences Dialog“ to control this behaviour.

2. **MQTT Channel Support**
   You can now define, alter, display MQTT channels and their status. In addition, In addition the number of connected MQTT Channels can be monitored. A new option in the location dialog controls whether MQTT commands are shown for a location. Please see “Options“ for more information.

3. **New compare dialogs**
   It is now possible to compare MQTT Channels, Channel Authentication and Subscription definitions.

4. **Browse message summary display**
   The browse message list display will now summarize the message content of known message formats such as DLH, XQH and event messages. Options are available in the Browse tab of the “Preferences Dialog“ to control this behaviour.

5. **Message browsing now supports up to 30,000 messages**
   The previous limit of 10,000 has been increased to 30,000 and made user configurable in the Browse tab of the “Preferences Dialog“ .

6. **Display browser connections**
   You can now display the list of connected HTTP clients from the main menu.

7. **Network Display Performance Improvements**
   The Verify processing of the Network Display is much faster, especially for large numbers of MQ objects.

8. **Limit browser connections by IP address**
   In the preference dialog you can now limit the device IP addresses which are allowed to connect to MO71

9. **Text selection and 'Copy to clipboard' is now supported from message browse and MQSC dialogs**

10. **Dual sort support in list displays.**
    Holding 'Alt' while clicking list titles with set and unset the minor sort field.

11. **Queue Manager status information shown in Queue Manager dialog**

12. **Support added for viewing local Queue Manager error logs and INI files.**

13. **'Alter List ' dialog has been improved allowing it to be resized and field order changed.**

14. **Auto-save of the configuration**

15. **Menu categories**
    Command menus can now be grouped into categories to enable easier finding of command. As before the menu settings to use is set in the “Preferences Dialog“.

16. **SMDS and SMDSConns support**
    Support for z/OS SMDS has been added, including updates to the CFSTRUCT commands.

17. **Dialog Group Refresh**
    It is now possible to refresh an entire group of dialogs with a single click. This option is available in the context menu of the predefined dialog member, the predefined dialog group or any dialog within the group.

18. **Allow filter history to the global rather than by dialog**
    There is a new option in Lists tab of the “Preferences Dialog“.

19. **Network Display**
    The user can choose to display the count of the number of running MCA channels and client connections. The last refresh time is also displayed on the Network display window.

20. **Both Queued and MQI flavours of Pub/Sub supported in Publish dialog**

21. **Last Put Dialog and Publish Dialog options are saved and restored**

22. **Predefined Dialogs enhanced to include more dialog options**

23. **Last used Dialog export options saved and restored**

24. **Thousands, List and Indicator separators have been added**
    To aid readability, especially of large numbers, configurable separators have been added.

25. **Right justified numeric columns**
    To aid readability, especially of large numbers, columns of numerical data can be right justified.

26. **Subtract TOPIC object stem from subscription topic string**
    For some odd reason the display of a subscription object that was created using a TOPIC object will return the TOPIC object stem as part of the topic string. This has ramification on export object particularly when export in MQSC. This version of MO71 can be configured to remove the TOPIC object stem.

27. **Export Title Text Preference setting**
    It is now possible to control whether a title line is output on the first line of a text or MQSC export. You can also control the format of the title line. Look at the 'Export' tab in the Preferences Dialog.

28. **Look and Feel consistency**
    Many of the dialogs have been changed to conform to the colour scheme. In some cases this has meant that the TAB controls and buttons are now drawn by MO71 rather than by Windows itself. As such you may notice a slightly different look in these controls however their operation should be the same.

29. **Various minor usability improvements**
    Improve usability of filter function 'set'.
    Suppress 'Not authorised' responses from list displays.
    Various menus and lists have been alphabetised.
    Queue Manager name added to location dialog window title
    Filter 'ambiguous field' detection has been tightened
    Confirm on program exit
    Message Browse indicates whether message may have been converted
    Colour dialog will now pre-select the closest current colour scheme
    Some dialogs have been reordered to form a more natural grouping
    Changes to MQSC window such as automatic clearing of the command field and scrolling to the bottom of display

Changes made in Version 7.5.1

1. **MQGem re-branding**
   MO71 is now maintained by MQGem Software Limited. The program and documentation has been updated to reflect this. All customer requests should now be directed to support@mqgem.com.

2. **MQMONA removal**
   Information about the MQMONA agent has been removed from the documentation. MQMONA gave a performance benefit over client connections by allowing multiple responses from the command server to be sent and received in a single message. However, with the advent of read-ahead connections this advantage is greatly negated and doesn't really outweigh the advantages so the concept has been removed. If this causes a problem for anyone please let me know.

Chapter 29. Migration from previous versions

We always try to ensure that, as each version is shipped, all the features that were working on the previous versions remain intact. When installing a new version we always recommend you backup the previous MQMONNTP program and the configuration file MQMON.CFG. If you do find a problem then please send us a problem report and we will try to fix your issue as soon as possible.

Migrating from a version prior to Version 9.3.3

Commands menu

Version 9.3.3 added the commands menu to all Queue Manager dialogs rather than just a select few such as Queue Manager List. To help with menu consistency the commands submenu has moved a little lower in the context menu.

Console and MO71Event changes

Two new fields, **Text** and **Category**, have been added to the console. This means that it now makes sense for textual additions to be in the new text field rather than the **Type** field. This has meant that an alert/event may not be displayed exactly as it was before. In the case of MO71Event you can get the previous behaviour by defining environment variable **MO71EVENT_v1=yes**

Rename of 'Queue Statistics' to 'Reset Queue Statistics”

The command “Reset Queue Statistics” was previously referred to just as “Queue Statistics”. However, now with the advent of real Queue Statistics this menu has been changed to Reset Queue Statistics to remove this ambiguity.

Migrating from a version prior to Version 9.3.2

MO71 ported to 64-bit

Every version of MO71 up to this point has been a 32-bit application. With IBM MQ announcing the intention to drop support for 32-bit applications Version 9.3.2 of MO71 has now been ported to 64-bit. This means that MO71 will now want to load the 64-bit versions of IBM MQ libraries and, as such, these must be installed and made available to MO71

Migrating from a version prior to Version 9.3.1

Location Queue Manager Export

This version of MO71 introduces the concept of exporting Queue Manager(Other) data from the location. Exactly what 'other' means depends on the Queue Manager version and platform. However, it essentially captures ancillary data such as the Queue Manager status fields.

Blank Field Location Export

You can now explicitly state whether you want 'Blank Fields' to be exported with the definition. Without the blank fields the export can clearly be considerably shorter but including them gives you a more complete definition which may reduce ambiguity.

Predefined Dialog Types

Prior to Predefined graphs a predefined dialog could either be 'group' or 'member'. However, since we now need to distinguish between a 'list dialog' and a 'graph' then the types are now 'group', 'list' and 'graph'.

Migrating from a version prior to Version 9.2.3

Queue Manager Auto Export

Prior to Version 9.2.3, the automatic export of objects did not export QSGDISP(GROUP) objects, and instead did export QSGDISP(COPY) objects. This is not useful when building a list of DEFINE commands to later replay. With Version 9.2.3 the default behaviour is to export QSGDISP(SHARED), QSGDISP(GROUP) and QSGDISP(QMGR) objects. QSGDISP(COPY) objects are not exported. You can continue to have the old behaviour if you want it, by selecting the appropriate check boxes on the Location Dialog's Export tab. See 24.3.6. Export on page 219.

Migrating from a version prior to Version 9.2.1

Reply Queue

Prior to Version 9.2.1 the Reply Queue was not opened with the INQUIRE option. This has been added so that MO71 can interrogate the MAXDEPTH value of the Reply Queue. This allows MO71 to warn the user if lists which contain too many items (for the size of reply queue) are used. If required this check can be switched off in the Lists tab of the Preferences Dialog.

Complete Responses

MQ Command responses contain a monotonically increasing sequence number. From version 9.2,1 MO71 will check these sequence numbers and warn the user if one or more messages appear to be missing. This can alert the user that messages are being lost, for example being put to a Dead Letter Queue. If required this check can be switched off in the Lists tab of the Preferences Dialog.

Migrating from a version prior to Version 9.2.0

HTML file changes

Prior to Version 9.2.0 the `qmlist()` location implementation used JavaScript to implement twisties. This has been replaced in 9.2.0 to use the HTML 5 tags `<details>` and `<summary>` to reduce reliance on JavaScript. File ***mqadminnav.html*** has been updated. If the browser you use to interact with MO71 web access pages does not support the `<details>` and `<summary>` tags, please see 27.2.1 Twisties not working on page 243 in the Troubleshooting section.

CMDSCOPE

In previous versions the CMDSCOPE field of an object dialog was initialised to the Queue Manager name if an object was selected from a list dialog. However, this prevents making changes to objects belonging to multiple Queue Managers in a single dialog. Since initialising the field offers no practical benefit it is no longer done in version 9.2.0.

Migrating from a version prior to Version 9.1.3

Compare differences

Prior to Version 9.1.3 the differences between object definitions were highlighted by greying out the definitions that were the same. In Version 9.1.3 the difference is shown by changing the background colour. If you wish the 'changed background' colour can be set in the ' Colour Dialog'. The prior behaviour can be re-instated by changing the 'Show compare difference using change colour' in the Lists tab of the Preferences Dialog.

AMQP Channels

New dialogs for AMQP Channel Status and the Channel Status list have been added. If you don't use AMQP channels then you can suppress dialogs from the command menu by unchecking the option in the location dialog in the Options tab.

Migrating from a version prior to Version 9.1.2

PUTDATE and PUTTIME Fields

Prior to version 9.1.2 you could say that you wanted to see these fields in 'local' time. However, Version 9.1.2 introduces Date/Time fields for all Date and Time combinations. So, now you should display the Put Date/Time field. This has the advantage that you can change the time format to your preferred date format.

Timezone

MO71 used to allow you specify a time-zone using a time offset. This has been improved to now be an actual Timezone. ie. One based on location rather than just a simple time offset. This is so that DST processing occurs correctly. MO71 will choose a time-zone which matches your previous time offset. However, there are often multiple Timezones with the same timezone bias and you may need to select the one with the correct DST settings manually.

Migrating from a version prior to Version 9.1.1

Main Window view-search keyboard shortcut

The keyboard shortcut to change whether the Queue Manager search field is shown in the main window has changed from Alt-s to Ctrl-Shift-s. This was done to avoid clashes with dialog action menus.

Weak SSL CipherSpec

Two more SSL Cipher Specs, TLS_RSA_WITH_AES_256_CBC_SHA and TLS_RSA_WITH_AES_128_CBC_SHA, are now marked as 'weak'.

Migrating from a version prior to Version 9.0.3

AUTHREC MQSC Values

Version 9.0.3 introduces support for exporting AUTHREC objects in MQSC format. Prior to this point it was only possible to export them in text, CSV and XML format. The reason for this was that MO71 supported exporting AUTHREC object before there even were MQSC commands for these objects. This meant that MO71 had to invent the MQSC keywords to be used. Unfortunately

not all of these keywords match the official keywords now that AUTHREC MQSC commands have been defined. This means that an XML export and AUTHREC filters can be slightly different that in previous versions of MO71.

Maximum field/column lengths

With Version 9.0.3 MO71 will now, by default, not necessarily show the entire field data. This helps to make the output more compact and easier to digest. The default sizes can be changed in the "Preferences Dialog'. If the data has been truncated MO71 will display ' … ' at the end of the field.

For example object lists will only show roughly the first 50 characters of any field. If the user wishes to see more they can make the column wider.

Similarly when using Web Access the HTML pages may not now, by default, contain all the field data. This means that web pages are more compact and more efficient. It does however mean that the user cannot see all of the data if required. If required the value can be set to a larger value or set to the value 0 which implies no limit should be used.

Message Selection now based on Message Token

Previous versions of MO71 used either the message position and/or the message id to identify a particular message. This was not ideal because neither is guaranteed to be unique. Basing the message selection on Message token has the advantage that it is far faster and that message identification is guaranteed to be correct. Unfortunately there are three downsides with message tokens:

1. **They are not available on all Queue Managers.**
   However, they were introduced many releases ago, around MQ V7.0, so the vast majority of Queue Managers should now have this support.
2. **They are not maintained across Queue Manager restarts**
   If you cycle your Queue Managers then messages may get a different Message Token. It might be necessary to refresh any dialogs that spanned across the Queue Manager restart.
3. **No way of getting the previous message**
   Message Tokens can greatly increase the speed of browsing a queue. Consider a deep queue where you are looking at the message at position 100,000. MO71 can use the Message Token to refresh this dialog or get the next message and it will be very quick. However, unfortunately there is no way in MQ to say MQGMO_BROWSE_PREVIOUS. MO71 would have to browse the previous 99,999 messages to find the message you are after. If you have a Message list containing the message you are after then it would be far quicker just to double click on that list entry rather than hitting the 'Prev' button.

Location dialog option to select Cipher Specs renamed

The option on the Security tab of the location dialog that was previously named “Allow SSL Version 3 Cipher Specs” has been renamed to “Allow Weak Cipher Specs” in line with the changes in the way IBM refers to these deprecated algorithms. In addition, the set of Cipher Specs considered deprecated when this option is not selected is brought in line with recent changes to the IBM MQ product.

Queue Manager Object Dialog – some fields moved to a new Logging Tab

With the addition of a new tab called “Logging” on the queue manager object dialog, some of the fields previously shown in the “Status” tab have now been moved to this new tab.Migrating from a version prior to Version 9.0.2

Client connection to host list

If you define a location that may be located at different hosts then the behaviour may have changed. If you define the client channel connection list directly in MO71, rather than using a CCDT, then MO71 will, by default connect to each entry in the list one by one. This is so that MO71 can tell which machine is currently hosting the Queue Manager. You can switch this behaviour off if you prefer in the Connection tab of the Preferences Dialog. The advantage of knowing which machine is hosting the Queue Manager is that you can have MO71 indicate which server is active by following the host address with a text string. An example might be

```
host1(1414) “Primary”, host2(1415) “Secondary”
```

Migrating from a version prior to Version 9.0.0

New Menu Category

With the addition of two new object dialogs, for Buffer Pools and Page Sets, the z/OS storage objects have all been groups together in a new menu category called 'Media'. This category contains Buffer Pools, Page Sets, SMDS, CF Struct and Storage Classes.

Migrating from a version prior to Version 8.0.3

Location value

The location value that can be defined against a location has been increased from 49 characters to 99 characters.

SSL Version 3.0 Cipher Spec

It is recommended that the SSL Version 3.0 Cipher Specs are no longer used. For this reason, by default, they will not be shown in the channel definition panels. However, if for some reason you still need to define channels which use these old Cipher Specs then you can re-enable them in the security tab of the location dialog.

Client channel maximum message length

If, when defining a new location to connect to, you connect as a client you can choose to provide the client channel definition directly by pressing the 'Configure' button in the location dialog. This shows a client channel definition where you can choose which fields you wish to specify. If you don't specify a maximum message length value then the value used for this field was taken from the MQCD_DEFAULT constant which was 4MB. However, this means that you cannot display, move or copy messages larger than this value without modifying the definition. As a consequence this 4MB value has been increased to 100MB. This essentially means that the maximum message size which can be manipulated via this location definition is given by the SVRCONN channel definition at the other end of the connection. This is the effectively the same mechanism as is used by the MQSERVER environment variable.

Web Access

The default HTML files have been improved and simplified. In addition the style sheet file simple.css has been reorganised and the unused styles removed. The result is that the Web interface will show the MQ data in an iframe display where the Queue Manager links are in a navigation pane on the left and the actual MQ data is shown on the pane on the right. If you have modified the HTML yourself then these will continue to work as before. If you want to use the old style with a new installation then you can replace the mqmadminindex.html file with the mqmadminindex.html.old file.

Migrating from a version prior to Version 8.0.2

Predefined Dialog Refresh Rate

The predefined dialog refresh rate has been changed from a numeric field which was the number of seconds to a character field which is a formatted field such as [n Days] [n Hours] [n Minutes] [n Seconds]. This makes the field much easier to use and understand. However, sorting the predefined dialog list by refresh rate is now no-longer correct since it will do an alphabetical sort rather than a numeric sort. To enable correct sorting an additional numeric refresh rate has been added to the list of predefined dialog fields. This can be displayed in the list and sorted on or you can sort using a filter with something like 'sortd(ratesec)'.

Location name in Queue Manager dialog

Version 8.0.2 displays the MO71 configured location name in a Queue Manager dialog. This value will also be exported in Text, CSV and XML format. This means that exported files between MO71 V8.0.2 and previous versions will not generate identical files.

Filter sort of enumeration list fields

The sort order of enumeration types has changed from numeric to alphabetic. Consider, for example the filter 'sort(type)' applied toa channel list. Previously the records would be sorted numerically based on the channel type field. So, for example, MQCHT_SENDER (or the value 1) would be lowest in the list. However, from version 8.0.2 onwards this will sort the fields alphabetically so 'Client Connection' channels are lowest in the list. If you really want to revert to the numeric sort you can do so by using the filter 'sort(+type)'. This forces the parameter passed to the sort function into a numeric value.

Auto Export

The place where Auto-export of Dialog data is done has changed. You should not notice a difference in behaviour however if you make use of this feature then you are recommended to verify that it is still working as expect in version 8.0.2.

Auto Export Timestamp

The auto export timestamp was erroneously being output in GMT. This has been changed and the timestamp is now in local time.

Migrating from a version prior to Version 8.0.0

Save Objects

Version 8.0.0 introduces the concept of application monitoring. Application monitoring stores data in application and queue objects. As a consequence you should ensure that you have the save objects preference option set to ensure that this monitoring data is saved. Equally if you are transferring your configuration between machines you should copy the MQMON.DFN file as well as the normal MQMON.CFG file.

Network Colours

The colours used in the network display are now called Diagram colours since they are also used in the Application View Display.

Objects in main window

The default has changed so that objects are now shown in the main window by default.

Objects in main window performance

Filling the main window container could take considerable time since the windows control takes a relatively long time to initialise. Some users, with large number of Queue Managers and large numbers of MQ objects, have therefore switched off the objects in main window. Version 8.0.0 has changed the way in which the main window is initialised so that the records are only filled when required. This should mean that for all users the objects can be shown in the main window without a performance penalty.

Main Window Double-click

You can now double-click on the object titles to show lists of the object type. For example, double clicking on 'Queues' will show the list of queues for that Queue Manager.

Multi-object select on Network Display

Both the network diagram and the application view diagram allow you to select and move multiple objects. Either click on the object while holding the <Ctrl> button. Or click on the background, while holding the <Ctrl> button, to drag out a selection box.

Save Objects

Version 8.0.0 introduces the concept of application monitoring. Application monitoring stores data in application and queue objects.

Migrating from a version prior to Version 7.5.2

Licencing

Version 7.5.2 was the first version which requires a licence to run.

Tabbed Dialogs

The fields in the tabbed dialogs (Channel, Queue, Queue Manager, etc) have been reordered to form a more logical grouping. As a consequence the order of the fields will be exported has also changed. If you are comparing exported files then the files may be identified as different even though the only difference is the order of fields.

Invoking Windows Commands

To invoke the command interpreter the format of the command has changed slightly. Suppose we wish to invoke a batch file mybatch.bat. The command which should be used is 'cmd mybatch.bat'.

Auto-config save

By default MO71 will now save it's configuration automatically whenever a significant configuration is changed. This behaviour can we switched off in the preferences dialog.

'Not authorised' message suppression

If a user issues a command such as 'DIS Q(*)' and doesn't have display authority to all the queues some versions of MQ send a not authorised message to inform that there are some queues the user isn't allowed to see. Later versions of MQ have removed this behaviour since it doesn't actually achieve anything. Previous version of MO71 used to display these 'Not authorised' messages, however, they are suppressed by version 7.5.2.

Allow browse of more than 10,000 messages

From a performance reason browsing deep queues is not recommended, especially over a client connection. However, there are some times when it is the simplest approach. As a consequence the limit of 10,000 messages has been increase to 30,000. However, the default setting in the preferences dialog is still 10,000 so will need to be changed if you wish to browse 30,000 messages at one time.

Browse message summary

In the message list display the prior versions of MO71 used to display the first 100 bytes of the message. However, version 7.5.2 introduces the concept of a message summary which tries to summarize the content of the message. It turns out that 120 bytes allows more formats to be interpreted so this is now the default. This does mean that slightly more data is transmitted in a browse under version 7.5.2. This values can be reduced to 108 in the preferences dialog although this will reduce the amount of information in the message summary.

Clipboard copy Ctrl-A operation

In previous versions of MO71 Ctrl-A was a short cut for 'Add Location' . However, MO71 now supports clipboard copy from the message dialog and MQSC window. These windows therefore support text selection and so Ctrl-A has been changed to the more usual meaning of 'select all'.

Search Colour

This version introduced a new colour for searched text. This is used, for example, for displaying found text in the MQSC window. It may be necessary to alter this colour to match your colour scheme.

MQTT Commands

Version 7.5.2 has added MQTT channel commands. However, MQTT channels are fairly specialised. If required these commands can be suppressed from the menu by changing the location options.

Filter ambiguity checking

The rules for determining whether a filter field is ambiguous has been tightened up. As a consequence it is possible that a truncated variable could now be regarded as ambiguous and the filter therefore fail. It is always recommended, certainly for saved filters, that the full field name be used. As fields are added to MQ objects there is no guarantee that a non-ambiguous field will not become ambiguous in the future.

Dialog Settings

Version 7.5.2 saves and restores many more dialog attributes. For example, the export file and all the export settings. Similarly the auto refresh export settings. Previously some of these fields were saved globally. It may then be necessary to initialise the fields with their initial values.

Queue Manager Version

This version of MO71 will warn the user if it detects a Queue Manager which is later than the version supported by the program. This should not be a problem at the time of release since the latest released version of MQ is 7.5 which is supported by MO71 7.5.2.

Separators

This version of MO71 introduces configurable separators for lists, indicators and integer numbers. Consequently the format of some exports will change. If required the separators can be changes back to a simple comma in the preferences dialog.

Subscription Topic Strings

When displaying a subscription, by default, this version of MO71 will remove any TOPIC object stem. If you prefer you can disable this in the preference dialog. However, it is the default because it more accurately reflects the subscription that was made. Clearly if you are exporting the subscription definition in MQSC then you must have TOPIC stem stripping enabled.

Migrating from a version prior to Version 7.5.1

Version 7.5.1 was essentially a re-branding exercise as MQGem Software took over the support and maintenance of MO71. All customer queries and support requests should now be made to MQGem Software at support@mqgem.com

Appendix A: Icons

These are the icons used in MO71, together with brief descriptions of their meaning.

A.1. Queue Manager Icons

	Queue Manager - unknown state
	Queue Manager - Available state
	Queue Manager - Unavailable state
	Queue Manager - Waiting reply
	Queue Manager Group - unknown state
	Queue Manager Group - No queue managers are unavailable
	Queue Manager Group - One or more queue managers are unavailable

A.2. Queue Icons

	Queue List
	Local queue - Unknown depth
	Local queue - less than 20% full
	Local queue - less than 40% full
	Local queue - less than 60% full
	Local queue - less than 80% full
	Local queue - over 80% full
	Alias queue
	Model queue
	Remote queue

A small ‘c’ character will be displayed on the icon if the queue is representing a cluster queue definition.

A.3. Channel Icons

	Channel - unknown status
	Channel - STOPPED status
	Channel - RUNNING status
	Channel - other status (e.g. RETRYING)
	Cluster Channel - unknown status
	Cluster Channel - STOPPED status
	Cluster Channel - RUNNING status
	Cluster Channel - other status (e.g. RETRYING)
	Channel – SVRCONN
	Channel – SVRCONN – RUNNING status

A.4. Other Object Icon

	Bufferpool
	Pageset
	Process
	Empty Namelist
	Namelist with one name
	Namelist with multiple names
	Storage Class
	Topic
	Application
	Application (Active)
	Application (In Error)

A.5. Network View Icons

	Queue Manager definition list empty
	Queue Manager definition list populated
	Queue Manager definition list is populated but the definitions are ‘old’. The setting of what is regarded as old is set in the ‘Network’ tab of the preference dialog.
	Queue Manager definition list refreshing
	One or more channels running
	One or more channels retrying (none running)
	One of more channels stopped (none running or retrying)

A.6. Container Icons

	Sub tree expanded
	Sub tree expanding/contracting
	Sub tree contracted
	Option selected
	Option deselected

Figure 23:Icons

A.7. User Icons

The network and application views allow the user to specify their own icons to be displayed for Queue Managers and Applications. These icons can be any BMP file. A BMP file can be created very simply by a variety of drawing tools, including the standard Paint program that comes free with windows[51].

One key point to note for user bitmaps is that whatever colour of pixel is in the top-left most position will be regarded as the transparent colour. Wherever this top-left colour appears in your bitmap the background will show through.

[51]If you create a set of icons that you are particularly proud of and they are suitable for use by others then please feel free to send them to me and I may include them in a future version of MO71

Appendix B: Constants

Constants are integer values that have a given name and can be used in expressions. Constants can be used in either upper or lower case. Constants are grouped into various categories described below:

Boolean Values

true	The value '1' representing the true condition
false	The value '0' representing the false condition

Console Priorities

These constants are used in the csl() function to set the console priority

info	Lowest priority
low	
med or **medium**	
high	Highest priority

usrcol() values

The usrcol() function allow for a variety of different behaviours depending on which values are passed to the function.

alignl	Align the chart element to the left (if a bar) or the top (if a column)
alignr	Align the chart element to the right (if a bar) or the bottom (if a column)
avgl	Long term average. This option should be used where a long term average rather than the current value is needed. The equation used to calculate value is newval = 0.02*currentval + 0.98*oldval[52]
avgs	Short term average. This option should be used where a short term average rather than the current value is needed. The equation used to calculate value is newval = 0.1*currentval + 0.9*oldval[52]
bar	Display the value as a horizontal bar
bgtext	The background colour applies just to the text
bottom	Draw the text at the bottom of the cell
bytes	An option indicating that the value is in bytes. If the user has Units display active then the value will be displayed accordingly.
centre	Draw the text in the centre of the cell
col	Display the value as a vertical column
diff	Display the different between the current value and the last value
left	Draw the text on the left of the cell
ms	An option indicating that the value is in milliseconds. If the user has Units display active then the value will be displayed accordingly.
narrow	Display the bar or column slightly narrow of the cell
opaque	Draw the text with the background colour
percent	The text value displayed should be displayed as a percentage of the maximum
rate	This option indicates that the value should be considered a 'rate'. Often, but not always, used in conjunction with 'diff'. For example, consider the case where we wish to have a column showing how many bytes a channel is moving. We can have a user column that is the **diff+rate** of the bytes sent. MO71 will divide the value passed by the number of seconds since the last refresh thereby giving you a transfer rate.
reversed	Draw the column and bar charts in reversed mode. bar – bar drawn from the cell right edge col – columns drawn from top and bottom edges rather than centre
right	Draw the text of the right of the cell
secs	An option indicating that the value is in seconds. If the user has Units display active then the value will be displayed accordingly.
small	Draw the text using a smaller font
text	Draw the text value
thin	Display the bar or column thinner than the cell. Thin is thinner than narrow.
top	Draw the text at the top of the cell
vthin	Display the bar or column as a very thin chart
zero	Display the bar or column so that the zero point is in the centre
zeroline	Draw a line at the bar or column zero point
w10	Set minimum column width to 10 character wide
…	Constants **w10, w20** , **w30**, **w40** and so on up to **w150** . Each one increasing the minimum width of the column by 10 characters. For 'bar' type displays the column width automatically expands to fill the space available.
w150	Set minimum column width to 150 characters wide

[52]There is no guarantee that this equation will remain for all time. It is included to help the user understand the processing.

Colours

Various functions allow the specification of a colour such as **fg()**, **bg()**, **usrcol()** etc. There are two types of colour constants; fixed colours, which always represent the same colour, and scheme colours, whose values change depending on the colour scheme in use.

Fixed Colours

Fixed colours are just the appropriate red-green-blue (RGB) value which in the Windows Graphics system actually has the order BGR. For example, the colour blue is 0xFF0000 which is 16,711,680 in Decimal. The colour constants are far easier to remember than the numeric equivalent. However, you can just use a number if you wish to set an exact colour. As mentioned previously you can also use the scheme colours in the table following this one.

The meaning of most of these constants are self-explanatory so are not included.

black	
blue	
brown	
cyan	This is a kinda greenish-blue colour
dblue	Dark Blue
dcyan	Dark Cyan
dgray	Dark Gray
dgreen	Dark Green
dpink	Dark Pink
dred	Dark Red
gray	
green	
lblue	Light Blue
lcyan	Light Cyan (this is kinda lighter greenish-blue)
lgray	Light Gray
lgreen	Light Green
lpink	Light Pink
lred	Light Red
pink	
red	
white	
yellow	

Scheme Colours

Scheme colours are a way of referring to the colours being used in the current colour scheme. For example, suppose you wish to set the colour of something to the same colour that is being used for the Listbox titles. You can do so by using the appropriate constant below. There are even four colours, User Colour 1, User Colour 2 etc which can be set to any colour you like, using the Colour Dialog, and then referenced in your filters.

BUTTONS_BG	Buttons Background
BUTTONS_FG	Buttons Foreground
CLR1	User Colour 1 User colours are solely for the use of colours within filter expressions. Other parts of the user interface will not use these colours. They allow the user to define a filter which can then be coloured using the Colour Dialog described on page 229.
CLR2	User Colour 2
CLR3	User Colour 3
CLR4	User Colour 4
CONTAINER_BG	Container Background
CONTAINER_FG	Container Foreground
CONTAINER_FOCUS	Container Focus
CONTAINER_LINK_FG	Container Link Foreground
DATA_BG	Data Background
DATA_FG	Data Foreground
DATA_HEADER_BG	Data Titles Background
DATA_HEADER_FG	Data Titles Foreground
DATA_SEARCH_BG	Data Search Background
DATA_SEARCH_FG	Data Search Foreground
DFT_OBJECT_BG	Default Object Background
DFT_OBJECT_FG	Default Object Foreground
DFT_OBJECT_TITLE_BG	Default Object Title Background
DFT_OBJECT_TITLE_FG	Default Object Title Foreground
DIAGRAM_BG	Diagram Background
DIAGRAM_FG	Diagram Foreground
DIAGRAM_FOCUS_BG	Diagram Focus Background
DIAGRAM_FOCUS_FG	Diagram Focus Foreground
DIAGRAM_FRAME_BG	Diagram Frame Background
DIAGRAM_LINK_FG	Diagram Link Foreground
DIAGRAM_LINK_HI_FG	Diagram Link Highlight Foreground
DIALOG_BG	Dialog Background
ENTRY_BG	Entryfield Background
ENTRY_FG	Entryfield Foreground
FIELD_BG	Field Background
FIELD_BG_RO	Protected Field Background
FIELD_FG	Field Foreground
FIELD_FG_RO	Protected Field Foreground
FIELD_TITLE_BG	Field Title Background
FIELD_TITLE_BG_RO	Protected Field Title Background
FIELD_TITLE_FG	Field Title Foreground
FIELD_TITLE_FG_RO	Protected Field Title Foreground
FILTER_BG	Filter Background
FILTER_FG	Filter Foreground
LIST_BG	List Background
LIST_FG	List Foreground
LIST_LOWLIGHT	List Lowlight Foreground
LIST_SELECT_BG	List Select Background
LIST_SELECT_FG	List Select Foreground
LIST_STRIPE_BG	List Stripe Background
LIST_TITLES_BG	List Titles Background
LIST_TITLES_FG	List Titles Foreground
MENU_BG	Menu Background
MENU_FG	Menu Foreground
PRINT_FG	Print Foreground
STATUS_BG	Status Background
STATUS_FG	Status Foreground
TRCM_CHANNEL_ARROW	Trace Message Channel Arrow
TRCM_CHANNEL_LABEL_BG	Trace Message Channel Label Background
TRCM_CHANNEL_LABEL_FG	Trace Message Channel Label Foreground
TRCM_QMGR_BG	Trace Message Queue Manager Background
TRCM_QMGR_FG	Trace Message Queue Manager Foreground
TRCM_RESOLVE_ARROW_BG	Trace Message Resolve Arrow Background
TRCM_RESOLVE_ARROW_FG	Trace Message Resolve Arrow Foreground

Appendix C: BB Codes

The followng codes can be used in the Media Titles in the Network UI tab of the Preferences Dialog on page 204.

The purpose of the BB codes are to allow greater flexibility in the way the data is displayed. Note that these codes are only loosely based subset of the standard BB codes and many of the codes are not supported.

Open Tag	Close Tag	Meaning and Example
[b]	**[/b]**	[b]This is in bold[/b]
[i]	**[/i]**	[i]This is in italics[/i]
[ul]	**[/ul]**	[ul]This text is underlined[/ul]
[code]	**[/code]**	[code]This is text shown in a monospaced font[/code]
[size=*fontsize*]	**[/size]**	[size=large]This text shown larger[/size]
[fg=*colour*]	**[/fg]**	[fg=red]This text is shown in red[/fg]
[bg=*colour*]	**[/bg]**	[bg=yellow]The backgound colour of this text is yellow[/bg]

- **Colours**
  The list of colours that can be specified are: **red**, **blue**, **cyan**, **pink**, **green**, **white**, **yellow**
  In addition there are six special values

  ctr_fg - The currently configured container foreground colour

  ctr_bg - The currently configured container background colour

  clr1 - User Colour 1, please see the Colour Dialog described on page 229.for setting this colour.

  clr2 - User Colour 2 (see above)

  clr3 - User Colour 3 (see above)

  clr4 - User Colour 4 (see above)

- **Font sizes**
  This list of font sizes which can be specified are: **small**, **large**, **vlarge**

BB codes can be nested. So, for example to get large, blue, bold text you can specify:

[b][size=large][fg=blue]Big Blue Heading[/fg][/size][/b]

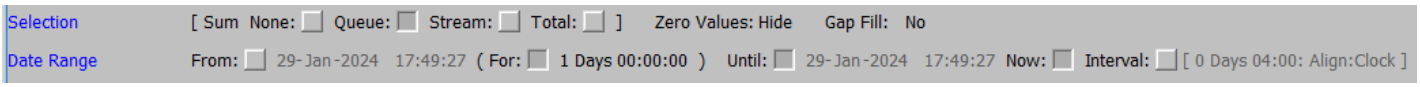
Note that if you are nesting tags that you must specify the closing tag in reverse order.

Appendix D: Filter function 'set' - Field names

When using the 'set' function in a filter, the first parameter to the function is a field name. This field name is the MQSC spelling of the field you wish to reference. It is case insensitive. Normally it will be easy to determine the field name, and mostly you can hover your mouse pointer over the appropriate column in the list to find the field name.

There are a few exceptions to this where the field names are not obvious, and those are listed in the table below.

This table shows the spinner fields that are found in several list dialogs, namely the MQEV Event, Accounting and Statistics data lists. The 'set' function can directly address the individual parts of the spinner as needed. The **MQEV** User Guide should be consulted for the full set of values that can be supplied in these fields.



Spinner	Spinner section	Field name	set function example
Selection	Sum	`sum`	`set(sum,queue)`
	Zero Values	`zerovals`	`set(zerovals,show)`
	Gap Fill	`gapfill`	`set(gapfill,inner)`
Date Range	From	`from`	`set(from,"29-Jan-2024 09:30")`
	For	`from` **OR** `to`	`set(from,"-2days")` **OR** `set(to,"4days")`
	Until	`to`	`set(to,"29-Jan-2024 17:30")`
	Now	`to`	`set(to,"now")`
	Interval	`intvl`	`set(intvl,"4h30m")`
	Align	`intvlalgn`	`set(intvlalgn,mq)`

Notes:

1. When using `set(sum, ...)` if you wish to enable more than one check-box in the sum spinner, provide the names in a comma-separated list, e.g. `set(sum,queue,stream)`

   You cannot combine `set(sum,none)` or `set(sum,total)` with any other names. Only a single value can be used for those cases.

2. `'until'` is allowed as a synonym for the keyword `'to'` as that is what you see on the screen.

Appendix E: Display National Language Characters

In displaying a formatted message there are 3 codepages involved.

- The codepage of the message stated by the CCSID
- The codepage for the program reading and displaying the message, depending on the language of Windows
- The codepage used by the font to display the contents of a message, depending on the locale settings

In most cases when only ASCII characters (characters < x'80') are used these settings are not critical, however when other characters, like umlauts and the Euro-sign, are used these setting become more critical. As an extra handicap, with the introduction of the Euro Microsoft simply added the Euro to their codepages while IBM created new codepages.

The necessary translations from the program to the display are controlled by the locale Preferences field, selecting the correct language will be sufficient in most cases. The translation between the message and the program is controlled by the Browse CCSID Preference option and Convert option.The table shows the recommended setting which will suffice in most cases.

Language	Codepage	
	Windows NT/2000[53]	Browse CCSID[54]
Arabic	1256	5352
Brazilian	1252	5348
Chinese (Simplified)	936	936
Chinese (Traditional)	950	950
Chinese (Hong Kong)[55]	950	950
Czech	1250	5346
Danish	1252	5348
Dutch	1252	5348
English	1252	5348
Finnish	1252	5348
French	1252	5348
German	1252	5348
Greek[3)]	1253	5349
Hebrew	1255	5351
Hungarian	1250	5346
Italian	1252	5348
Japanese	932	932
Korean	949	949
Norwegian	1252	5348
Polish	1250	5346
Portuguese[55]	1252	5348
Russian	1251	5347
Spanish	1252	5348
Swedish	1252	5348
Turkish	1254	5350

Figure 24:Codepages

[53]For non-Western European language versions of Windows NT/2000 (codepage <> 1252), this value can be changed during installation of Windows NT/2000 and can be different from the value shown here.
This information is derived from www.microsoft.com/globaldev/reference/oslocversion.mspx

[54]This information is derived from www-3.ibm.com/software/ts/mqseries/support/euro/eurotabl.html

[55]These languages are not supported in Windows NT

End of document