MO72: MQSC Client for WebSphere MQ
User Guide

WebSphere MQ Development
MP211,
IBM UK Laboratories Ltd.
Hursley
Winchester
Hants, SO21 2JN
United Kingdom

Take Note!

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

Sixth Edition, May 2012

This edition applies to Version 1.9 of MQSC Client for WebSphere MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

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

Notices

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF 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.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not be submitted to any formal IBM 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 IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:
WebSphere MQ
IBM
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

.

Chapter 1. MQSC Client for WebSphere MQ

This document describes the functions available in the SupportPac.

Overview

Although the *MQSC Client for WebSphere MQ* provides an MQSC portal is has two quite distinct modes of operation:

1. For issuing MQSC commands to a Queue Manager through either a local or a client connection.

2. For issuing MQSC commands against the client channel table.
   This mode does not require a connection to a Queue Manager. All commands are processed directly by the program itself.

Both modes also provide a number of additional formatting options for displaying the results of the commands.

Installation

The program is provided in a zip file containing various directories, essentially a directory for each of the platforms. As well as the current version of the SupportPac I have included some of the previous program versions. This is for two reasons. Firstly, it is not always possible for me to build the latest version of the SupportPac on all the platforms. Secondly, I am often asked for previous versions of the SupportPac since the latest version may not run on earlier hardware. So, if the platform is missing in the current list please check previous versions. By all means send me an email if a platform you really need is missing and I'll see what I can do although clearly I can make no promises.

Once you have found the program you want to run, create a directory, say ***MQSC***, and copy the following files into it :-

- MQSC.EXE (On windows)
- mqsc (On Unix)

The *MQSC Client for WebSphere MQ* does not produce a configuration file of its own, but if you are an existing user of the SupportPac MO71: "WebSphere MQ for Windows GUI Administrator", and you run MQSC.EXE from the same directory as that program, it will use the MQMON.CFG configuration file produced by MQMONNTP.EXE. This allows you to reuse the same definitions that you have already created. There is an option to control this behaviour, detailed in *"Connecting using the MQMON.CFG configuration file"* on page 4.

Once the MQSC program is in the path it can be run just by typing 'mqsc' on the command line. No further set-up at this point is required. The program will dynamically load either the WebSphere MQ client or WebSphere MQ Queue Manager libraries as required. Note that to successfully connect to a Queue Manager at least one of these WebSphere MQ products must be installed. The WebSphere MQ client is available for free download at :-

WebSphere MQ Client Version 6

http://www-1.ibm.com/support/docview.wss?rs=171&context=SSFKSJ&dc=D400&uid=swg24019253&loc=en_US&cs=UTF-8&lang=en&rss=ct171websphere

WebSphere MQ Client Version 7

http://www-1.ibm.com/support/docview.wss?rs=171&context=SSFKSJ&dc=D400&uid=swg24019253&loc=en_US&cs=UTF-8&lang=en&rss=ct171websphere

WebSphere MQ Client Version 7.1

http://www-1.ibm.com/support/docview.wss?rs=171&context=SSFKSJ&dc=D400&uid=swg24019253&loc=en_US&cs=UTF-8&lang=en&rss=ct171websphere

Chapter 2. Getting Started

The *MQSC Client for WebSphere MQ* requires the command server to be running in order for it to operate. On z/OS the command server is automatically started when the queue manager starts, on other platforms use:-

```
strmqcsv QM1
```

You have several connection methods available to you with the *MQSC Client for WebSphere MQ*.

Modifying the client channel table

To modify the client channel table there is no need to connect to an actual Queue Manager, all that is required is to tell the program that you want to run in client only mode, this is done using the –n flag.

```
mqsc –n
```

Connecting via a Client

To connect to the Queue Manager via a client, use the following options on the command line if you are using the client channel table or the environment variable MQSERVER.

```
mqsc –m QM1 –l
```

The –l parameter is not specified the program will connect as a local application[1].

Connecting via a Client but targeting a different Queue Manager

To target a different Queue Manager than the one you're connected to specify the –M flag.

```
mqsc –m QM1 –l –M QM2
```

Connecting via a Client – specifying channel parameters

To connect to the Queue Manager via a client, use the following options on the command line if you want to directly specify the channel name and connection name.

```
mqsc –m QM1 –l –c SYSTEM.DEF.SVRCONN –h myhostname(1414)
```

Connecting directly to a Queue Manager

To connect directly to a Queue Manager, use the following options on the command line.

```
mqsc –m QM1
```

[1] On Unix this differs from previous versions of the SupportPac where connecting as a client was the default. For consistency with Windows all version will run as a local application unless the client flag is specified.

Connecting using the MQMON.CFG configuration file

To connect to the Queue Manager using the same parameters as used in MQMONNTP.EXE, use the following options on the command line to indicate the configuration file is to be used. The second example show the syntax for specifying the location of this file if it is not in the directory the MQSC.EXE resides in.

```
mqsc –m QM1 –g

mqsc –m QM1 –g –f c:\MQMON
```

Connecting to a z/OS Queue Manager

If the Queue Manager is running on z/OS, this must be indicated to the *MQSC Client for WebSphere MQ* by use of the **–z** option.

```
mqsc –m CSQ1 –l -z
```

Chapter 3. Issuing MQSC Commands

You can issue any of the commands in the *WebSphere MQ Script (MQSC) Reference* manual via the *MQSC Client for WebSphere MQ*. These commands may be typed in manually or issued via an input file. The output from the commands can either be displayed in the command window (and there are a number of ways to control the format of this output discussed in Chapter 5. Display Options on page 12) or the output can be sent to a file. MQSC.EXE will not wait indefinitely for commands to complete.

Specifying a Reply Queue

By default MQSC will use a reply queue of SYSTEM.DEFAULT.MODEL.QUEUE. To change this queue to another use the **–**r parameter.

```
mqsc –m QM1 –rREPLYQ –i Commands.txt
```

Specifying a Reply Prefix

If a model queue is used as the reply queue then the actual queue name will be of the form :-

MQMON.<*userid*>.<*Hex String*>

This may be changed if require using the –R command.

```
mqsc –m QM1 –RMQSC.* –i Commands.txt
```

This will cause the reply queue to be called something like **MQSC.4180E6A706EC0420.**

Specifying the Command Queue

By default MQSC target the command queue for that platform, either

SYSTEM.ADMIN.COMMAND.QUEUE or **SYSTEM.COMMAND.INPUT**

To direct commands to a different queue use the **–q** parameter.

```
mqsc –m QM1 –qMY_COMMAND_QUEUE –i Commands.txt
```

Issuing a single command from the command line

To run a single command directly from the command line enter a command line the following..

```
mqsc –m QM1 –C “display Q(Q1)”
```

Using an Input File

To input commands from a text file use one of the following options.

```
mqsc –m QM1 –i Commands.txt
```

```
mqsc –m QM1 < Commands.txt
```

Using an Output File

To send the output from your commands to a text file use one of the following options. It may be appropriate to combine this option with the *Command Echoing* option described on page 12.

```
mqsc –m QM1 –o Output.txt
```

```
mqsc –m QM1 > Output.txt
```

Controlling the wait time

The default time that the *MQSC Client for WebSphere MQ* will wait for a command to complete is 5 seconds. To change this, use the following parameter string.

```
mqsc –m QM1 –p wait=10
```

Chapter 4. Modifying the Client Channel table

To modify the client channel table you need to start the program in client only mode using the –n flag. You are then presented with a command prompt. At the command prompt you can enter normal MQSC commands for the manipulation of channels and authinfo objects. The commands supported are:

- DISPLAY, DEFINE, ALTER, DELETE CHANNEL
- DISPLAY, DEFINE, ALTER, DELETE AUTHINFO

The syntax and capabilities of the commands are essentially the same as those supported by Queue Managers but there are some key differences explained below. It is strongly recommended that a backup is taken of the client channel table before any commands to alter the file are issued.

Channel Table Location

There is a default name and location for channel table file or it can be given explicitly to the program using the **–t** parameter. The channel table name is constructed from two paths: [FILE PATH][FILE NAME]. These two parts are constructed as follows:

File Path

The path to the location is determined using the following values, in order of precedence:

1) Any path specified using the –t parameter
   A path will be considered to be included i if the value contains a directory separator character (or ':' for Windows). For example, **mqsc –n –t c:\tables\**

2) The value given by the MQCHLLIB environment variable

3) The installed path of WebSphere MQ (Windows Only)

4) Default location

   a) c:\mqm For Windows

   b) /var/mqm For Unix

File Name

The name of the file used is determined using the following values, in order of precedence:

1) Any file name given using the –t parameter
   The file name will be assumed to exist if the –t parameter **does not** end with a directory separator. For example, **mqsc –n –t myfile.dat**

2) The value given by the MQCHLTAB environment variable

3) AMQCLCHL.TAB

MQSC Syntax

Where ever possible the syntax of the MQSC commands supported on the Queue Managers has been reproduced so it should be possible to run the channel command scripts directly against the program. However, for usability some of features have been extended or restrictions removed.

Channel Type

Normal MQSC channel commands require that the channel type attribute is explicitly specified. Since this program is only concerned with channels with a CHLTYPE of CLNTCONN, this restriction is removed. A channel **can** be defined as simply as ***DEF CHL(TEST)*** and will be a CLNTCONN channel.

Generic names

Queue Manager MQSC commands allow a generic name to be used by specifying a wildcard character at the end of the name. For example, ***DIS CHL(T*)*** will display all the channels starting with the character 'T'. This has been extended in the program to allow two forms of wildcards which can be placed anywhere in the string:

'?'	matches a single character
'*'	matches zero of more characters

So, for example:

DIS CHL(*T)	Will display any channels ending with the character 'T'
DIS CHL(*QM1*)	Will display any channels with 'QM1' somewhere in the name
DIS CHL(????)	Will display any channels with a four character name

Where Clause

WebSphere MQ Version 6 introduced the WHERE clause on DISPLAY commands. It is now possible to display objects filtered by something other than their name. The WHERE clause is supported by the program regardless of the version of MQ which actually installed. So, for example:

- **DIS CHL(*) WHERE(HBINT GT 300)**
  Will display any channels with a heartbeat of greater than 300 seconds
- **DIS CHL(*) WHERE(CONNAME LK *1414*)**
  Will display any channels which have the string '1414' in the connection name

The WHERE clause has been extended in three ways.

1. **Multiple expressions**
   It is possible to link multiple expressions using the operators 'AND' and 'OR'. For example:

   a. **DIS CHL(*) WHERE(HBINT GT 300 AND CONNAME LK *1414*)**

2. **Object name can be used in the expressions**
   It is possible to use the object name itself in the where expression. For example

   a. **DIS CHL(*) WHERE(CHL GT m)**

3. **Operators**
   The operators LT,GT,EQ,NE,LE,GE can be replaced with <,>,=,!=,<=,>= respectively.

Where Clause on Alter

Queue Manager MQSC only supports the WHERE clause on the DISPLAY verb. However, there are times when it is useful to be able to make a 'selective' alter as well. For this reason I have allowed the WHERE clause to be specified on ALTER. For example:

- **ALTER CHL(*) WHERE(HBINT EQ 300) HBINT(400)**
  Will alter the heartbeat interval, of all the channels that currently have a heartbeat interval value of 300, to 400.

- **ALTER CHL(*) WHERE(CONNAME LK *host1*) CONNAME(host2)**
  Will alter all channels which are currently configured to connect to 'host1' to now connect to 'host2'.

Channel Definition Version

When a channel definition is written to a client channel table file is contains an inherent structure version. The version of structure dictates which versions of WebSphere MQ can read the definition. It is not possible to define a WebSphere MQ Version 7 client channel table file and have it read by a WebSphere MQ 6.0 client. A consequence of this is that client channel table definitions should be defined on a Queue Manager which is sufficiently old to include the versions of all the clients which will use the file.

This is not a practical solution for this SupportPac since it would require a different version of it for each WebSphere MQ release. Instead I allow the specification of the client channel versions in the file either on a global level using the –v parameter or on each client channel definition using the VERSION attribute.

By default, without using any of the mechanisms described below, the client channel definitions are written using the latest structure version known. In this release of the SupportPac this will mean at WebSphere MQ V7.0 level.

- **mqsc –n –v6.0**
  Write all channel records at the WebSphere MQ Version 6.0 level.
  Versions supported are : 5.0, 5.1, 5.2, 5.3, 6.0,7.0 and 7.1

- **mqsc –n –vs6**
  Write all channel records with version 6 structures. This clearly requires knowledge of which structure version is appropriate for each WebSphere MQ release. The mapping of which structure version is appropriate for a particular WebSphere MQ or MQSeries release or a particular platform, is documented in the Intercommunication product manual.

- **mqsc –n –v0**
  Write all channel records at the same version as they we read from the file. Any new channel definitions will be written in the latest version.

- **DIS CHL(TEST) VERSION**
  Display the version of a particular channel. Since this is not a Queue Manager MQSC supported attribute, it is not included when DIS CHL(TEST) ALL is used, but must be requested specifically, for example DIS CHL(TEST) ALL VERSION. The value shown in this attribute will be the WebSphere MQ or MQSeries product version number, for example,

VERSION(5.2). If the version of the client channel is earlier than the support versions, it will be shown as ‘s’ followed by the MQCD version number, for example, VERSION(s3).

- **DEF CHL(TEST) ……VERSION(5.3)**
  Write this particular channel at WebSphere MQ 5.3 level. This means that this channel can be read by a WebSphere MQ client at version 5.3 or later.

- **ALTER CHL(*) VERSION(5.3)**
  Alter all channel definitions in the channel file to be written out in WebSphere MQ 5.3 level.
  This command is clearly useful for migrating channel definitions from one release to the other. Channel records can be altered both to later or earlier releases. Note, however, that when migrating to older levels that some of the information in the channel definition may be lost since the earlier release did not support that option. As usual it is recommended that a backup of the channel table is taken before any modifications are made.

Authentication Information Objects

The channel definition file doesn’t really contain AUTHINFO object definitions; what it contains is a list of Certificate Revocation List (CRL) locations in a single object definition. However, to manipulate this single object it is easier to think of it as multiple objects. The defined object in WebSphere MQ which maps to the information contained in the list is the AUTHINFO object. However, this is a mapping which, unfortunately, is not complete. The notable differences are:-

- **Object name**
  With discretely defined objects each one has a unique name. However, in the client channel file none of the CRL location entries has an associated name. The consequence of this is that the name has to be generated. The name used is merely the index in the list. So, for example the first entry in the list is given the name **‘1’**, the second **‘2’** and so on. Bear in mind that the name is not fixed for a particular entry. The same entry could have different names depending on where it currently exists in the list.

- **Description**
  Normally each AUTHINFO object is entitled to have its own description. However, since all AUTHINFO objects are part of the single object they can only have a single description which they share. Therefore changing the description of one AUTHINFO object will change it for all.

  A channel file which has come from the server will not contain a description.

- **Alter Time/Date**
  In a similar fashion to the description field above the alter time/date fields are shared by all the AUTHINFO objects.

The other requirement when manipulating a list of objects is having some control over the order of the list. In particular it is necessary to have the notion of “insert this definition at this point in the list”. This is done using the ‘INSERT’ keyword. This gives us the following options:

- **DEFINE AUTHINFO(1) CONNAME(HOST1)**
  This will define a new AUTHINFO object at the start (position 1) of the list but will fail if the list already contains a definition.

- **DEFINE AUTHINFO(1) CONNAME(HOST1) REPLACE**
  This will define a new AUTHINFO object at the start (position 1) of the list and will replace any definition currently in the first position.

- **DEFINE AUTHINFO(1) CONNAME(HOST1) INSERT**
  This will define a new AUTHINFO object at the start (position 1) of the list and will 'push' any current definitions further down the list. Note that this means the name by which these subsequent AUTHINFO objects are referred to, changes.

There are two other variations will allow you to define an entry at the end of the list

- **DEFINE AUTHINFO(999) CONNAME(HOST1)**
  Currently you can only define 10 definitions. Any number larger than this will be taken to mean 'at the end'. This command will therefore put this definition at the end of the list.

- **DEFINE AUTHINFO(myname) CONNAME(HOST1) INSERT**
  Although the name of the AUTHINFO should be a number the program will allow the use of a name. This is to maintain as much compatibility with Queue Manager MQSC as possible. Essentially a list of DEFINE AUTHINFO commands will be accepted. Note that the name, regardless of what it is, will be treated as 'end of the list'. So, a list of DEFINE commands will define the objects in the order of the DEFINE commands themselves. Also note that once the definition is made, the name itself is lost. The command DISPLAY AUTHINFO(myname) would therefore fail.

Chapter 5. Display Options

There are a number of options for controlling the display of the output from MQSC commands. These are discussed in this chapter. Many of these options are specified by means of a parameter string.

Command Echoing

To suppress the command issued being echoed to the output use the following option.

```
mqsc –m QM1 -e
```

Number of Columns

To configure the number of columns to be used for the output use the following parameter string. Without this option in effect, the number of columns is based on the width of the command window you issue the command in.

```
mqsc –m QM1 –p cols=2
```

```
AMQ8409: Display Queue details.
DESCR(Alias for local queue Q1)         TARGQ(Q1)
CLUSTER( )                              CLUSNL( )
QUEUE(Q1.ALIAS)                         ALTDATE(2004-08-30)
ALTTIME(20.43.51)                       GET(ENABLED)
PUT(ENABLED)                            DEFPRTY(0)
DEFPSIST(NO)                            SCOPE(QMGR)
DEFBIND(OPEN)                           TYPE(QALIAS)
```

Output Width

To configure the width of the output, use the following parameter string. If you do not specify the number of columns to use then that value will be tailored according to the width of the output. If the command output is being sent to a file, the default width is 80 columns. Otherwise, the width is set to the width of the Command Window.

```
mqsc –m QM1 –p width=30
```

Line Wrap

To output the result of a command in a single line wrapping must be switched off.

```
mqsc –m QM1 –p wrap=off
```

Compress Attributes

The command output from some platforms, particularly z/OS, returns space padded values. The default behaviour is to remove these extra spaces and compress the output. To override this default behaviour and see all the spaces, use the following parameter string.

```
mqsc –m QM1 –p comp=no
```

Output Separator

By default, each command issued is separated by a WebSphere MQ message before the command output, such as

```
AMQ8409: Display Queue details.
```

There are a number of other possibilities that can be used to separate the output from commands. The values are listed in *Table 1: Separator Options* below.

detmsg	**Numeric value**	**Separator**
default	0	WebSphere MQ message
none	1	No separator
blank	2	Blank line
dotted	3	Single line of dashes
solid	4	Single line of underscores

Table 1: Separator Options

These separators can either be before or after the output from the command using the values listed in *Table 2: Separator Position* below.

detsep	**Numeric value**	**Separator**
before	0	Before command output
after	1	After command output

Table 2: Separator Position

To use a specific separator use the following parameter string.

```
mqsc –m QM1 –p detmsg=dotted,detsep=after
```

Not authorised messages suppression

If an attempt is made to list objects and the user is not authorised to see all of those objects then the command server will send a ‘not authorised’ response. These responses can be distracting and so by default they are suppressed. They can be shown using the command.

```
mqsc –m QM1 –p notauthmsg=yes
```

Prompt

The prompt for entering commands in *MQSC Client for WebSphere MQ* can be tailored using the following parameter string, An insert of %m is available to insert the queue manager name into the prompt. The default prompt is the queue manager name followed by ‘>’.

```
mqsc –m QM1 –p prompt=”Commands for %m>”
```

```
Connected to Queue Manager 'QM1'

Commands for QM1>
```

The list of inserts which can be used in the prompt string are :-

- m
  The target Queue Manager unless the target Queue Manager is also used in the prompt in which case it is the connected Queue Manager.
- M
  The target Queue Manager
- ?
  Is used to specify part of the prompt which is only displayed if the connected and target queue managers are different. For example the prompt ***%M%?(via %m)>*** will display the part in parentheses only if the connected and target Queue Managers are different.

Chapter 6. Parameter String

A number of the behaviours already described are specified by means of a parameter string. There are a number of ways of specifying this parameter string. It can be done on the command line when you start the *MQSC Client for WebSphere MQ* as already described in the previous chapters. It can also be specified in other ways which are described in this chapter.

Run Time Changes

You can display the current set of parameter string options while the *MQSC Client for WebSphere MQ* is running by typing the following.

```
=?
```

This will give you output like the following.

```
>=?
Client Tabe Ver  : 7.1
Width            : [width ]     = 120
Cols             : [cols  ]     = 6
Wrap             : [wrap  ]     = off
Formatting       : [format]     = yes
Compress         : [comp  ]     = yes
Detail Messages  : [detmsg]     = dotted
     valid  values [default|none|blank|dotted|solid]
Detail Separator : [detsep      = before
Not Auth Messages: [notauthmsg] = no
Wait             : [wait  ]     = 5
Prompt           : [prompt]     = %m>
Environment Var. MQSC_PARMS = <unset>
```

You can then change any of the options using the same parameter name as described in earlier chapters, prefixed with an equals sign as shown in the following example.

```
=detmsg=solid
```

Environment variable

You can set these parameters by means of an environment variable named MQSC_PARMS. This allows you to set a default set of parameters so that you don’t have to configure it every time you run it. For example:

```
set MQSC_PARMS=cols=3,detmsg=dotted,wait=10
```

Chapter 7. History of Changes

Version 1.9

- Added WebSphere MQ V7.1 channel format support
- Added 'Client Table Version' to parameter output
- Added 'Not auth messages' suppression.

Version 1.8

- Added WebSphere MQ V7.0 channel format support
- Added **–C** single command parameter
- Added **wrap=off** option
- Added SecurityExit and SecurityData parameters

Version 1.7

- Added support for client channel file manipulation
- Improved usability of display format commands
- Allowed new –M parameter to target a different Queue Manager than the one connected to

Version 1.2 (July 2005 edition)

- Added HP Support

Version 1.2

- Added –R parameter
- Added support for ***Via*** Queue Managers

Version 1.1

- Internal only version

Version 1.0

- Original version.