MA10: MQSeries MVS/ESA ISPF Utilities
Version 1.1.9

User Guide

Document: MA10
Issued: 23rd June, 1994
Revision Date: 6th September, 2000
Previous Revision Date: 25th July, 2000
Next Review: As required

Mark Butcher
IBM UK Laboratories Ltd.
Hursley Park
Take Note!

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


This edition applies to Version 1.1.9 of MQSeries for MVS/ESA ISPF Utilities, and to all subsequent releases and modifications until otherwise indicated in new editions.

A form for reader's comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM United Kingdom Laboratories
Transaction Systems Marketing Support (MP207)
Hursley Park
Hursley
Hampshire, SO21 2JN, England

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

© Copyright International Business Machines Corporation 1999. All rights reserved.
Note to U.S. Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.
## Contents

### Introduction
- Installation (1)
  - Enabling utilities (3)
  - Source code (4)

### MQBRO
- Limitations (5)
- Configuration/Components (7)

### MQCMD
- Sub-commands (12)
- Limitations (12)
- Configuration/Components (13)

### MQDLQ
- Sub-commands (15)
- Limitations (16)
- Configuration/Components (17)

### MQR2C
- How to use MQR2C (18)
- Some examples on how to invoke MQR2C (19)
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 been 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:

MQSeries
IBM
MVS
ESA
OS/390
MQM
## Summary of Changes

<table>
<thead>
<tr>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>24th June 1994</td>
<td>Initial release</td>
</tr>
<tr>
<td>28th June 1994</td>
<td>Clarify installation instructions.</td>
</tr>
<tr>
<td>24th January 1995</td>
<td>Add note in MQCMD chapter regarding queue security in response to APAR PN64646.</td>
</tr>
<tr>
<td>7th July 1995</td>
<td>Add source library</td>
</tr>
<tr>
<td>10th January 1995</td>
<td>Repackaged to allow WWW delivery</td>
</tr>
<tr>
<td>18th January 1996</td>
<td>Corrections to installation instructions</td>
</tr>
<tr>
<td>20th February 1996</td>
<td>Add instructions for transfer to host via ftp.</td>
</tr>
<tr>
<td>22nd May 1996</td>
<td>Add instructions for setting blocksize for transfer to host via ftp.</td>
</tr>
<tr>
<td>13th June 1996</td>
<td>Add source code for MQCMD.</td>
</tr>
<tr>
<td>11th October 1996</td>
<td>Add source code for MQBRO, Edit macro, Reason Code converter and sample JCL.</td>
</tr>
<tr>
<td>14th October 1996</td>
<td>Added section relating to compiling source code</td>
</tr>
<tr>
<td>28th January 1997</td>
<td>Ensure Working Storage cleared before use in modules MQBRO0, MQBRO1, MQBRO2. Source and Load modules updated.</td>
</tr>
<tr>
<td>21st May 1997</td>
<td>Message Descriptor (MD) is incorrectly cleared on refreshing the ISPF display when there is only one message on the queue. Messages &gt;32K cause the browser to loop.</td>
</tr>
<tr>
<td>5th August 1997</td>
<td>Correct and tidy up help panels</td>
</tr>
<tr>
<td>6th April 1999</td>
<td>Fixed storage problem. Moved the MQFB text from the COBOL code to the ISPF panel so they can be tailored.</td>
</tr>
<tr>
<td>26th May 1999</td>
<td>R2CMQA copybook has increased significantly in size for recent releases of MQSeries, causing MQ2RC to be updated. The documentation has been updated to highlight the need to generate the version of R2CMQA for the level of MQSeries appropriate to you.</td>
</tr>
<tr>
<td>17th June 1999</td>
<td>Improve instructions for preparation of R2CMQA copybook.</td>
</tr>
<tr>
<td>25th July 2000</td>
<td>Enable the browse utility to work with MQSeries for OS/390 V2.1. Added sample job for rebuilding COBOL utilities, and enabled MQR2C to handle MQFB reason codes as well as MQRC ones.</td>
</tr>
<tr>
<td>5th September 2000</td>
<td>Correct omission of entries in PREPCMQA for MQFB</td>
</tr>
</tbody>
</table>
Introduction

This document describes a number of utility programs (Applets) designed to perform small but useful functions with the IBM MQSeries for MVS/ESA product.

Installation

Take the following actions to install the SupportPac from the MA10.ZIP file:

1. Use INFOZIP's UNZIP to unpack the MA10.ZIP file.
   
   This produces

   **readme** Brief introduction
   **ma10.cli** Clist library
   **ma10.loa** Load library
   **ma10.men** Message library
   **ma10.pen** Panel library
   **ma10.src** Source library
   **ma10.jcl** Sample JCL library

2. The library files need to be transferred to the destination TSO system as sequential binary files with a record format of FB 80. Use one of the following methods to accomplish this:
   
   - Use the Communications Manager/2 SEND commands below to send the files to TSO as a sequential binary files:
     
     send ma10.cli A:ma10.clistseq
     send ma10.loa A:ma10.loadseq
     send ma10.men A:ma10.menuseq
     send ma10.pen A:ma10.penuseq
     send ma10.src A:ma10.srcseq
     send ma10.jcl A:ma10.jclseq
     where A is the TSO session ID.

   - To send them via ftp ensure the BINARY option is set then use the following commands:
     
     site fixrecfm 80
     put ma10.cli ma10.clistseq
     put ma10.loa ma10.loadseq
     put ma10.men ma10.menuseq
     put ma10.pen ma10.penuseq
put ma10.src ma10.srcseq

put ma10.jcl ma10.jclseq

- With Personal Communications, use the “Send Files to Host” option under the Transfer menu item to transmit to TSO

  PC File   ma10.cli etc
  Host File  ma10.clis1seq etc
  Transfer Type  loadlib

  The Transfer type of **loadlib** may need to be correctly setup. To do this, use the “Setup.Define Transfer Types” option under the Transfer menu item and create the **loadlib** type with the Ascii, CRLF and Append checkboxes all unselected, the **Fixed** radio button selected and the **LRECL** set to **80**

3. On TSO, issue the following commands to unload these sequential files into TSO partitioned datasets:

- **receive indsname(ma10.clis1seq)**
  when prompted for a filename, reply
  dsn(ma10.clis1)

- **receive indsname(ma10.loadseq)**
  when prompted for a filename, reply
  dsn(ma10.load)

- **receive indsname(ma10.menuseq)**
  when prompted for a filename, reply
  dsn(ma10.ispmenu)

- **receive indsname(ma10.penuseq)**
  when prompted for a filename, reply
  dsn(ma10.isppenu)

- **receive indsname(ma10.srcseq)**
  when prompted for a filename, reply
  dsn(ma10.source)

- **receive indsname(ma10.jclseq)**
  when prompted for a filename, reply
  dsn(ma10.jcl)

4. Confirm that the datasets have been unloaded correctly by browsing the PDS and ensuring the appropriate members exist as shown in the following table.
Enabling utilities

To get the applets up and running perform the following tasks.

1. Update your logon proc (or equivalent):
   - Put MA10.ISPMENU into the ISPMLIB concatenation.
   - Put MA10.ISPPENU into the ISPPLIB concatenation.
   - Put MA10.CLIST into the SYSPROC concatenation.
   - Put MA10.LOAD (This library must not be APF authorised) and the MQM library SCSQAUTH into the ISPLLIB or STEPLIB concatenation.

   **Note:** Also ensure that the appropriate Language Environment or COBOL runtime libraries are available to the TSO user, and are in the correct order in your library concatenations. A number of customers have experienced problems when moving to LE, as there have been old COBOL libraries ahead of the LE libraries in their library concatenations, which have resulted in the utilities in this SupportPac abending with 0C4s. Therefore if you are using LE, please ensure that the SCEERUN library is ahead of any COBOL runtime libraries in your library concatenations, particularly in your logon procs.

2. Update your favourite 'Utility Selection Panel' to include an 'MQAPPS' option. This option should display panel MQAPPS. For example

   ```
   % Q +MQAPPS - MQM MVS/ESA Utilities
   :
   )PROC
   :
   Q,PANEL(MQAPPS)
   :
   ```

3. Log on using the updated Logon Proc and try out the MQAPPS panel!
Source code
The source for these utilities, including any copybooks which are not part of the MQSeries for MVS/ESA product, are supplied in the MA10.SOURCE library.

To assemble and linkedit these programs, sample JCL is provided in the MA10.JCL library.

The version of R2CMQA shipped with this SupportPac is for MQSeries for MVS/ESA V1.1.1. If you are using a later version of MQSeries, then you must generate the most up-to-date version of copybook R2CMQA by running the supplied ISPF edit macro, PREPCMQA, against a copy of the CMQA dsect in library SCSQMACS, and renaming the resulting edited file as R2CMQA. See “MQR2C” on page 18 for detailed instructions for how to generate an appropriate version of R2CMQA.

The version of MQDLQ0 shipped with this version of the SupportPac has been built with MQSeries for MVS/ESA V1.1.4 and LE/370 V1.5. If you wish to rebuild it job MA10.JCL(COBOL) has been added as a sample.
MQBRO is an ISPF based tool that can be used to browse an MQM MVS/ESA queue. It uses the ISPF BRIF interface so the standard ISPF browse facilities are available e.g. HEX ON, FIND etc.

When MQBRO is started the MQM Queue Browse entry panel is displayed. On this panel the Queue Manager name and Queue name to be browsed must be entered. Also the MsgId and CorrelId of any messages to be displayed must also also be specified. If any message in the queue is to be browsed then a MsgId and CorrelId of hex zeroes must be entered in the 'Hex MsgId' and 'Hex CorrelId' fields as shown:

```
------------------------ MQM Queue Browse - Entry Panel -----------------------
COMMAND ===> _

Connect to MQM ===> VCSE
Queue Name ===> TEST.LOCAL.QUEUE1

Hex MsgId ===> 00
or
    ------+----1----+----2----+----3----+----4----+---
MsgId ===>

Hex CorrelId ===> 00
or
    ------+----1----+----2----+----3----+----4----+---
CorrelId ===>

MQMD Display ===> ON  Caps ===> ON
```

**Note:** You do not need to enter all 48 hex characters in the Hex MsgId and Hex CorrelId fields. Trailing Hex zeroes will be added. (If the MsgId or CorrelId are specified in the character form then it is padded with spaces.)

The 'MQMD Display' option is used to include the message descriptor associated with the message in the browse output. For example when set to ON each line of output in the browse screen is formatted as follows:

```
mqmd===>msg
```

Where mqmd is the message descriptor (it has not been formatted in any way, so you need to turn HEX ON if you want to see the hex values.) ===> is an eyecatcher and msg is the message put to the queue.

When 'MQMD Display' is set to OFF, only the message is output on the browse screen.

The CAPS field is used to control whether upper case translation takes place on the Msgid, correlid, queue name and queue manager names. The screen must be re-displayed before any change of this field is detected.

Once the queue details have been selected and the browse started a 'normal' ISPF browse screen will be shown.
As messages may be added or removed by other processes while MQBRO is browsing the queue, the line/message number shown by ISPF is not guaranteed. For example if the first screen is browsed MQBRO will still have a browse cursor open against the queue. If PF8 is entered and the next screen full of messages is displayed the line/message number will indicate for example LINE 0000024 even though the message is not necessarily the 24th message in the queue.

When browsing a large queue you will probably notice the performance degrade especially when scrolling backwards. This is due to the fact that you can only BROWSE_NEXT and BROWSE_FIRST via the MQI. This means that MQBRO has to start browsing the queue from the beginning when you scroll backwards.

If on the Browse entry panel you enter a MsgId and CorrelId that causes no matching messages to be retrieved, then a screen as follows will be displayed:

The MsgId and CorrelId used are displayed. You must enter the HEX ON command to display the MsgId and CorrelId inserts if they contains non-printable characters. In the above example a MsgId of Hex 0 and a CorrelId of 'QUERY-REQUEST00002333' were specified on the Browse entry screen. Unfortunately the Queue MY.REPLY.Q did not contain any messages which matched this selection criteria.
Limitations

When entering the ISPF BROWSE sub-command on the command line during an MQBRO browse, the ISPF BROWSE command is invoked. MQBRO is NOT called.

If an MQI error occurs, then depending on where the error occurs you will receive different types of error message, for example if the error occurs under the main MQBRO program the ISPF short and long message is set.

The ISPF BRIF interface specifies a maximum record length of 32760. This means using MQBRO you can only browse the first 32760 bytes of a MQM message. If you browse a queue which contains messages longer than 32760 bytes the message will be truncated.

```
------------------------ MQM Queue Browse - Entry Pan 0002 2085 MQOPEN
COMMAND ===> _
UNKNOWN_OBJECT_NAME       THIS.QUEUE.IS.NOT.DEFINED
    Connect to MQM ==> VCCI
    Queue Name => THIS.QUEUE.IS.NOT.DEFINED

```

In this example '0002 2085 MQOPEN' was set to the short ISPF error message. The '0002' is the MQI completion code, '2085' is the MQI reason code and 'MQOPEN' was the MQI call being issued at the time of the error. PF1 was then hit to get the long ISPF error message, and this contained 'UNKNOWN_OBJECT_NAME THIS.QUEUE.IS.NOT.DEFINED' which is the text version of the 2085 reason code (from MQR2C) and the name of the queue involved in the MQI request. (Depending on the MQI call being used the Queue or Queue Manager name may be given).

If the BRIF read routine encounters an error then the error message is TPUT to the screen.

```
0002 2001 MQOPEN ALIAS_BASE_Q_TYPE_ERROR MY.ALIAS.WITH.NO.BASE
***

```

Again this message indicates the MQI Completion Code, Reason code, MQI call, Text version of the reason code and the Queuename. In the above example the error occurred as the alias queue 'MY.ALIAS.WITH.NO.BASE' was defined, but the base (target) queue in the alias had not been defined.

If the error occurs during the Read Routine MQBRO usually tells ISPF that no messages are available or that an I/O error occurred. This may cause the Browse screen to be displayed with no data. (If this occurs you should have always received an error message as shown above).

Configuration/Components

The following components are required to run MQBRO.

- Panels allocated to the ISPF ISPPLIB concatenation:
  - MQBROP0
  - MQBROP0H
  - MQBROP1
  - MQBROP1H
- MQBROP2H

- Messages allocated to the ISPF ISPMLIB concatenation:
  - MQBRM00

- Clist used to start MQBRO which should be in a library allocated to the TSO SYSPROC concatenation:
  - MQBRO

As an alternative MQBRO could be put on an ISPF selection panel as shown in the following panel extract:

```plaintext
)BODY
  
  % B + MQBRO - Browse MQM MVS/ESA Queue
  
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD,'.') B,'PGM(MQBRO0) NEWAPPL(MQBR)'
  :

- Executable modules:
  - MQBRO0
  - MQBRO2

- Other programs called which should be in the ISPLLIB or STEPLIB concatenation:
  - MQR2C
MQCMD

MQCMD is an ISPF based tool that can be used to issue MQM commands in the 'script' format. (This is the type of command format that you use when entering an MQM command from the console!).

There are two main benefits of MQCMD over SDSF and the MQM MVS/ESA Operations and Control Panels:

1. Commands entered via SDSF have a quite short length restriction (often less than 80 bytes) and the output from the command may be truncated depending on the console output message limit. With MQCMD commands greater than 200 bytes may be issued and the output from the reply is not subject to the console reply limit. The reply does not also 'clog' up the SYSLOG.

2. If the user is familiar with the MQM commands, then using MQCMD is usually faster than the MQM MVS/ESA Operations and Control Panels. You can also issue commands which the Operations and Control Panels don't allow you to enter, for example DISPLAY THREAD(*).

MQCMD has one panel which is used for both input and output:

```
--- MQM MVS/ESA Command Interface ---
COMMAND ===> _  SCROLL ===> PAGE
  Connect to MQM ===> MQM1  Wait-For ===> 0003  Caps ===> ON
  SysCmd Queue ===> On MQM ===> MQM Command ===> DISPLAY THD(*)

--- Command Results: ---
*************************************************************************
```

The following is a description of the input fields:

**Connect to MQM**   This field is used to identify the name of the local Queue Manager you want to connect (MQCONN) to.

**Wait-For**         This field is used to set the wait interval in seconds that MQCMD uses whilst waiting for the replies from the command server.

You can set this value depending on whether you are sending commands to the local Queue Manager or a remote Queue Manager and whether the command server has a lot of work to do.

A wait interval of 5 is usually plenty when sending commands to the local Queue Manager.
If you set the 'Wait-For' time to 9999 then MQCMD will wait 'forever'... (Or until it gets a reply)

**Caps**

This field is used to control the upper case translation of the following fields on this panel:

- Connect to MQM
- SysCmd Queue
- On MQM
- MQM Command

This field may be set to ON or OFF. When set to ON the characters entered in the above field are folded to upper case. When set to OFF the above fields are NOT folded to upper case.

**Note:** MQM resources are case sensitive, for example Queue BOB, Bob and bob are three different queues!

**Note:** Don't forget that when specifying the MQM Command with Caps set to OFF, you must enclose the lower case data in quotes ('). For example with Caps OFF:

```
define qlocal(this.will.be.folded.to.uppercase)
```

will define queue called THIS.WILL.BE.FOLDED.TO.UPPERCASE, whereas

```
define qlocal('this.is.a.lower.case.queuename')
```

will define a queue called this.is.a.lower.case.queuename.

**Note:** A change of the Caps setting will not take place until the screen is re-displayed.

**SysCmd Queue**

This field is used when you want MQCMD to put the command on a queue other than SYSTEM.COMMAND.INPUT. For example, if you want to send commands to a remote queue manager, you could on the local Queue Manager define a remote queue to its command input queue. i.e. On MQM1 define a QREMOTE called MQM2.SYSCMDQ with a RemoteName of SYSTEM.COMMAND.INPUT and a RemoteQMName of MQM2.

**On MQM**

If this field is specified, then 'SysCmd Queue' must also be non-blank. It is used when you want to specify the Queue Manager name of the Queue specified in the 'SysCmd Queue' field.

**MQM Command**

This is the MQM command to be issued in 'script' format.

You must not include the MQM Subsystem Recognition Character (SRC) in the command. The target MQM is determined by the Queue manager being connected to and the destination Queue.

The results from the command are displayed in a scrollable window in the last half of the screen. For example, the output from a DIS THD(*) command may look like:
The 'normal' ISPF scroll commands may be used except for LEFT and RIGHT. (for example, you can scroll up and down or alter the 'SCROLL ===>’ field etc).

When the response from the command server is longer than the width of the screen, the remainder of the message is displayed as a second line. No parsing is done on the messages displayed so some of the replies can look quite messy. For example:
### Sub-commands

MQCMD supports only one sub-command: RETCMD. RETCMD is used to retrieve the last MQM Command entered in the 'MQM Command' field. Up to 10 previous MQM commands are retained.

It's usually best to set up RETCMD on a PFKEY!

### Limitations

MQCMD is written in COBOL II. Both the VS Cobol II and LE/370 COBOL run-time libraries do not currently provide multi-tasking support for COBOL. This means the MQCMD can NOT be run on two ISPF split screens at once. If you attempt to run MQCMD on both sides of a split screen, the results are unpredictable (you will usually get an ISPF abend).
Configuration/Components

The following components are required to run MQCMD.

- COBOL II run time.

- MQCMD uses a dynamic queue for it's reply-to-queue. It uses the Model Queue 'SYSTEM.COMMAND.REPLY.MODEL' to create the dynamic queue. The Dynamic Queue used is called 'userid.MQCMD.*'.

  **Note:** If your userids contain characters that are illegal in a queue name (for example, £ or #) you may need to modify and recompile the program. The COBOL II source is provided.

- Panels allocated to the ISPF ISPPLIB concatenation:
  - MQCMDP0
  - MQCMDP0H
  - MQCMDP1H
  - MQCMDP2H

- Clist used to start MQCMD which should be in a library allocated to the TSO SYSPROC concatenation:
  - MQCMD

As an alternative MQCMD could be put on an ISPF selection panel as shown in the following panel extract:

```plaintext
)BODY

%C + MQCMD - Issue MQM MVS/ESA Command

)PROC

&ZSEL = TRANS( TRUNC (&ZCMD,'.'))
  C, 'PGM(MQCMD0) NEWAPPL(MQCM')

)ENDPROC
```

- Executable modules:
  - MQCMD0

- Other programs called which should be in the ISPLLIB or STEPLIB concatenation:
  - MQR2C
MQDLQ

MQDLQ is an ISPF based tool that can be used to browse 'DEAD' messages on an MQM MVS/ESA Dead-Letter queue. Messages on the DLQ can also have their Message Descriptor's displayed as well as being deleted.

When starting MQDLQ it must be passed the name of the Queue Manager that you want to process. When MQDLQ is started, a screen as follows is displayed:

-------- MQM MVS/ESA Dead Letter Queue Browse --------
COMMAND ===> _
DLQName ===> MQMX.DEAD.QUEUE
MsgLen ===> 00000231 CurDepth ===> 00000029 ( 00000001 )
StrucId ===> DLH Version ===> +00000001
Reason ===> +00002085 MQRC_UNKNOWN_OBJECT_NAME
DestQ ===> AN.UNDEFINED.QUEUE
DestQmgr ===> MQMX
Encoding ===> +00000785 CodedCharSetId ===> +00000500
Format ===> MQCMD1 PutApplType ===> +00000007 QMGR
PutApplName ===> MQMX
PutDate ===> 19940503 PutTime ===> 10515261

-------------------------------- Dead Message: --------------------------------
CSQN2051 COUNT= 29, RETURN=00000000, REASON=00000000

This screen can be divided up into 3 sections:

1. Queue Details:
   - DLQName - The name of the Dead Letter Queue
   - MsgLen - The length of the message being displayed (Including the length of the DLQ header)
   - CurDepth - Current Depth of the Dead Letter Queue. The number in brackets after CurDepth shows an approximate indication of where in the queue you are.

2. Message Details:
   All of these fields are taken from the Dead Letter Header of the message being browsed. For an explanation of these fields, see the description of MQDLH in the MQSeries Application Programming Reference.
   Some of these output fields have been interpreted. For example, the Reason field shows 2085 and MQRC_UNKNOWN_OBJECT_NAME and PutApplType shows 7 and QMGR.

3. Message Data:
   The last 8 lines of the screen contain data from the 'Dead' message being browsed. You cannot scroll this information.
The usual ISPF scroll commands are NOT active on this screen. To browse the next message just hit enter. To go to a previous message use the FIRST or GO commands.

**Sub-commands**

MQDLQ supports a number of sub-commands:

- **DEL** The DEL subcommand deletes the message currently being browsed.
  
  **Note:** NO confirm delete operation is required. If you DEL a message and hit enter and the command does not fail, the message will be deleted!

  After you have deleted a message MQDLQ does not scroll to the next message, just hit enter to scroll past the message you have just DEL'd.

- **FIRST** The FIRST subcommand can be used to restart the browse of the Dead Letter Queue from the first available message.

- **GO** The GO subcommand requires a message number parameter. For example, GO 75 will cause MQDLQ to display the 75th message in the Dead Letter Queue.

- **MD** The MD subcommand is used to display the message descriptor (MQMD) of the message currently being browsed. A scrollable screen, as follows, will be shown:
### MQM MVS/ESA MQMD Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrucId</td>
<td>+00000001</td>
</tr>
<tr>
<td>Report</td>
<td>+00000000 COPY-MSGID-TO-CORRELID NEW-MSGID NONE</td>
</tr>
<tr>
<td>Msgtype</td>
<td>+00000002 REPLY</td>
</tr>
<tr>
<td>Feedback</td>
<td>+00000000 MQFB-NONE</td>
</tr>
<tr>
<td>Expiry</td>
<td>-00000001</td>
</tr>
<tr>
<td>CodedCharSetId</td>
<td>+0000500</td>
</tr>
<tr>
<td>Priority</td>
<td>+00000000</td>
</tr>
<tr>
<td>Format</td>
<td>MQDEAD</td>
</tr>
<tr>
<td>Persistence</td>
<td>+00000000 NOT-PERSISTENT</td>
</tr>
<tr>
<td>Msgid</td>
<td>CSQ MQMX Z Ú fnQê</td>
</tr>
<tr>
<td>Hex</td>
<td>CED4ECCC4444444444A3514994</td>
</tr>
<tr>
<td></td>
<td>328053390000000000098794585</td>
</tr>
<tr>
<td>Correlid</td>
<td>CSQ MQMX Z Ú fnQê</td>
</tr>
<tr>
<td>Hex</td>
<td>CED4ECCC4444444444A3514994</td>
</tr>
<tr>
<td></td>
<td>328053390000000000098794585</td>
</tr>
<tr>
<td>BackoutCount</td>
<td>+00000000</td>
</tr>
<tr>
<td>ReplyToQ</td>
<td>MQMX</td>
</tr>
<tr>
<td>ReplyToQMgr</td>
<td>MQMX</td>
</tr>
</tbody>
</table>

**Context Section:**

- **UserIdentifier**: GRANT02
- **AccountingToken**: 207715
- **Hex**: 62077150000000000000000000000000
- **ApplIdentityData**: +00000007 QMGR
- **ApplOriginData**: MQMX
- **PutDate**: 19940503
- **PutTime**: 10515261

For an explanation of these fields see the description of MQMD in the MQSeries Application Programming Reference.

Some of these output fields have been interpreted. For example, the Msgtype field shows 2 and REPLY; the Persistence field shows 0 and NOT-PERSISTENT.

Use 'END' to quit the MD display.

---

**Limitations**

MQDLQ is written in COBOL II. Both the VS Cobol II and LE/370 COBOL run-time libraries, currently, do not provide multi-tasking support for COBOL. This means that MQDLQ can NOT be run on two ISPF split screens at once. (Or MQDLQ on one split screen with MQCMD on the other) If you attempt to run MQDLQ on both sides of a split screen, the results are unpredictable (you will usually get an ISPF abend).
Configuration/Components

The following components are required to run MQDLQ:

- COBOL II run time.

- Panels allocated to the ISPF ISPPLIB concatenation:
  - MQDLQP0
  - MQDLQP1
  - MQDLQP0H
  - MQDLQP1H
  - MQDLQP2H

- Clist used to start MQDLQ which should be in a library allocated to the TSO SYSPROC concatenation:
  - MQDLQ

As an alternative MQDLQ could be put on an ISPF selection panel as shown in the following panel extract:

```plaintext
)BODY
  )BODY
  % D + MQDLQ - Process Dead Letter Queue on MQM%===>_DLQM+
  )PROC
    &ZSEL = TRANS( TRUNC (ZCMD,'.'))
    D,'PGM(MQDLQ0) NEWAPPL(MQDL) PARM(&DLQM)'
  )PROC
  )BODY
```

- Executable modules:
  - MQDLQ0

- Other programs called which should be in the ISPPLIB or STEPLIB concatenation:
  - MQR2C
MQR2C

MQR2C is a small utility program that is used to convert the MQI reason codes to a text equivalent. For example, if you have an MQI call that fails with a 2033 reason code, you can call MQR2C and it will return in an output area 'NO_MSG_AVAILABLE'. You can then use this textual equivalent of the reason code in error messages to end users etc.

**Note:** MQR2C includes the R2CMQA copybook, which is generated from the CMQA copybook shipped in the SCSQMACS library of MQSeries for MVS/ESA. The version of R2CMQA shipped with this SupportPac is for MQSeries for MVS/ESA V1.1.1. If you have a later release of MQSeries, then you must generate an up to date version of R2CMQA, as follows:

1. Add the MA10 CLIST library to your SYSPROC ISPF concatenation, by doing one of the following:
   - Adding it to the SYSPROC concatenation defined in your logon proc, in which case you will have to logoff and logon again to pick up the change, or
   - Adding it to the SYSPROC concatenation defined in a program that you run to setup your own ISPF environment, or
   - Using the TSO CONCATD command
2. Copy the CMQA macro from the MQM SCSQMACS library into a PDS which you can update.
3. While editing the copy of the CMQA macro execute the PREPCMQA macro, by entering PREPCMQA on the editor command line.
4. Once PREPCMQA is complete, save the member as R2CMQA.
5. Edit the provided job ASMMQR2C in the MA10 JCL library, tailoring it to your system, and adding your library containing the new version of R2CMQA to the front of the SYSLIB concatenation of the ASM step.
6. Submit the job, and it should assemble and link MQR2C cleanly.

**How to use MQR2C**

MQR2C is written in assembler. It requires no working storage of its own (It uses the MVS/ESA linkage stack) so it can be called in the MVS Batch/TSO/CICS and IMS environments. It uses standard OS linkage and requires to be passed two parameters:

1. The address of a fullword which contains the MQI reason code to be converted.
2. The address of a 30 byte area where you want the converted reason code to be stored.

If the specified reason code can not be found, the program ends with a return code of 4, and the first 4 bytes of where you wanted the converted reason code to be kept are set to '????'.

MQR2C may be called either statically or dynamically. Obviously, dynamically is probably best for most situations (so that you do not need to re-link your programs if a new reason code is added to the MQI).

If the program is to be dynamically called under CICS, you must add a PPT entry for it. For example

```
CEDA DEF PROG(MQR2C) LANG(ASSEMBLER) DATALOC(ANY)
```

Don't forget to also ensure that the MQR2C module is in a library in the CICS DFHRPL concatenation.
Some examples on how to invoke MQR2C

Static S/370 Assembler - TSO/Batch/CICS or IMS

CALL MQR2C,(REASON,DESC),VL,MF=(E,CALLLST)

REASON DS F
DESC DS CL30
CALLLST CALL ,(0,0),VL,MF=L

Note: Don't forget to include MQR2C with your module at link edit time!

Dynamic S/370 Assembler - TSO/Batch or IMS

LINK EP=MQR2C,PARAM=(REASON,DESC),MF=(E,LINKLST)

REASON DS F
DESC DS CL30
LINKLST DS 2F

Dynamic S/370 Assembler - CICS

EXEC CICS LOAD PROGRAM('MQR2C') ENTRY(R5) RESP(RESP)
LR R15,R5 set entry point address ..
CALL (R15),(REASON,DESC),VL,MF=(E,CALLLST)
EXEC CICS RELEASE PROGRAM('MQR2C') RESP(RESP)

REASON DS F
DESC DS CL30
CALLLST CALL ,(0,0),VL,MF=L

Dynamic COBOL - TSO/Batch/CICS or IMS

WS-CONVERT PIC X(8) VALUE 'MQR2C'.
WS-REASON PIC S9(9) BINARY VALUE ZERO.
WS-DESC PIC X(30) VALUE SPACES.

CALL WS-CONVERT USING WS-REASON WS-DESC.

Static COBOL - TSO/Batch/CICS or IMS

WS-REASON PIC S9(9) BINARY VALUE ZERO.
WS-DESC PIC X(30) VALUE SPACES.

CALL 'MQR2C' USING WS-REASON WS-DESC.

Note: Don't forget to include MQR2C with your module at link edit time!

Static C/370 - TSO/Batch/CICS or IMS
#pragma linkage(MQR2C,OS);
MQLONG reason;
char desc ??(30)??;

MQR2C( MQLONG *reason, char *desc);
MQR2C(&reason,desc);

**Note:** Don't forget to include MQR2C with your module at link edit time!
Sending your comments to IBM

MA10: MQSeries MVS/ESA ISPF Utilities
Version 1.1.9

MA10

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers’ Comment Form.
- By fax:
  - From outside the U.K., after your international access code use 44 1962 841409
  - From within the U.K., use 01962 841409
- Electronically, use the appropriate network ID:
  - IBMLink: IBMGB(AIMPACS)
  - Internet: aimpacs@uk.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.
MA10
Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

_______________________________  ________________________________
Name  Address

_______________________________
Company or Organization

_______________________________  ________________________________
Telephone  Email
MA10: MQSeries for MVS/ESA ISPF Utilities
MA10

You can send your comments POST FREE on this form from any one of these countries:

Australia  Finland  Iceland  Netherlands  Singapore  United States
Belgium    France    Israel    New Zealand  Spain       United States of America
Bermuda    Germany   Italy     Norway      Sweden      Switzerland
Cyprus     Greece    Luxembourg Portugal    Switzerland
Denmark    Hong Kong Monaco   Republic of Ireland  United Arab Emirates

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO

From:  Name
       Company or Organization
       Address
       EMAIL
       Telephone

IBM

REponce PAYee
GRANDE-BRETAGNE

IBM United Kingdom Laboratories Limited
Information Development Department (MP 095)
Hursley Park
WINCHESTER, Hants
SO21 2ZZ  United Kingdom