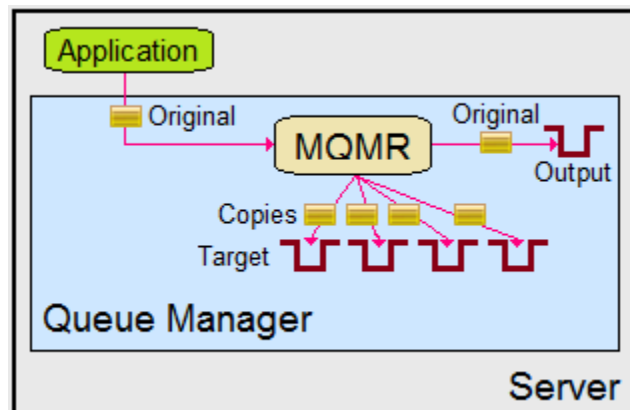


MQ Message Replication Installation and Operation Manual



Capitalware Inc.
Unit 11, 1673 Richmond Street, PMB524
London, Ontario N6G2N3
Canada
sales@capitalware.com
<https://www.capitalware.com>

Last Updated: January 2021.
© Copyright Capitalware Inc. 2015, 2021.

Table of Contents

1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 EXECUTIVE SUMMARY.....	2
1.3 CONTEXT DIAGRAM (LOGICAL VIEW).....	2
1.4 PREREQUISITES.....	3
1.4.1 <i>Operating System</i>	3
1.4.2 <i>IBM MQ</i>	4
1.4.3 <i>Windows 32-bit</i>	4
1.4.4 <i>Windows 64-bit</i>	4
2 INSTALLING MQ MESSAGE REPLICATION.....	5
2.1 WINDOWS INSTALLATION.....	5
2.2 LINUX 32-BIT INSTALLATION.....	5
2.3 UNIX AND LINUX 64-BIT INSTALLATION.....	6
2.4 IBM I INSTALLATION.....	7
2.4.1 <i>MQMR-GUI Installation</i>	8
3 CONFIGURING MQMR.....	9
3.1 API EXIT.....	10
3.1.1 <i>Windows</i>	10
3.1.2 <i>For Linux 32-bit</i>	13
3.1.3 <i>Unix and Linux 64-bit</i>	13
3.1.4 <i>IBM i</i>	14
3.2 FILE PATHS.....	15
3.2.1 <i>Absolute Path</i>	15
3.2.2 <i>Relative Path</i>	15
3.2.3 <i>Environment Variables</i>	16
3.3 MQMR-GUI.....	17
4 INIFILE KEYWORDS (GLOBAL VALUES).....	18
4.1 ACTIVE.....	18
4.2 EXITPATH.....	18
4.3 EXCLUDEAPPLICATIONS.....	19
4.4 EXCLUDEUSERIDS.....	20
4.5 LICENSEFILE.....	21
4.6 LICENSE KEY.....	21
4.7 LOGGING.....	22
4.8 ADDSOURCEINFO.....	23
5 MESSAGE REPLICATION BY QUEUE.....	24
5.1 WHAT VALUE IS TO BE USED FOR SECTION NAME/QUEUE NAME?.....	24
5.1.1 <i>Local Queue</i>	24
5.1.2 <i>Cluster Queue</i>	24
5.1.3 <i>Alias Queue</i>	24
5.1.4 <i>Remote Queue</i>	25
5.2 SECTION NAME FOR A QUEUE.....	26
5.2.1 <i>TargetQueues</i>	26

5.2.2 Context.....	26
5.2.3 AddSourceInfo.....	26
5.2.4 UseExpiry.....	27
5.2.5 Expiry.....	27
5.2.6 UseResolvedQueueName.....	27
5.2.7 ForMQPut1DoMQPut1.....	27
5.2.8 UseSyncPoint.....	28
5.2.9 CollapseMQMDE.....	28
5.2.10 ClearRO.....	28
5.2.11 ClearROCOA.....	28
5.2.12 ClearROCOD.....	28
5.2.13 ClearROPAN.....	29
5.2.14 ClearRONAN.....	29
5.2.15 ClearROException.....	29
5.2.16 ClearROExpiration.....	29
5.2.17 SkipCOA.....	29
5.2.18 SkipCOD.....	30
5.2.19 SkipPAN.....	30
5.2.20 SkipNAN.....	30
5.2.21 SkipExpiration.....	30
5.3 EXAMPLE.....	31
6 APPENDIX A – SUMMARY OF INIFILE.....	32
7 APPENDIX B – MQMR UPGRADE PROCEDURES.....	39
7.1.1 Windows Upgrade.....	39
7.1.2 Linux 32-bit Upgrade.....	39
7.1.3 Unix and Linux 64-bit Upgrade.....	40
7.1.4 IBM i Upgrade.....	40
8 APPENDIX C – CAPITALWARE PRODUCT DISPLAY VERSION.....	41
8.1 EXAMPLES.....	41
8.1.1 Windows.....	41
8.1.2 Linux 32-bit.....	41
8.1.3 Unix and Linux 64-bit.....	41
8.1.4 IBM i.....	41
9 APPENDIX D – SUPPORT.....	42
10 APPENDIX E – SUMMARY OF CHANGES.....	43
11 APPENDIX F – LICENSE AGREEMENT.....	45
12 APPENDIX G – NOTICES.....	47

1 Introduction

1.1 Overview

MQ Message Replication (MQMR) will clone messages being written (via MQPUT or MQPUT1 API calls) to an application's output queue and MQMR will write the exact same messages to 'n' target queues ('n' can be up to 100). When MQMR replicates a message both the message data and the message's MQMD structure will be cloned. This means that the fields of the MQMD structure (i.e. PutTime, MessageId, CorrelId, UserId, etc..) will be exactly the same as the original message's MQMD structure.

The MQAdmin can configure MQMR to perform message replication on any number of queues. The application messages put to each queue can be replicated and put to (up to) 100 target queues.

MQMR can clone messages being written to any MQ queue type. i.e. local, alias, remote and/or cluster queues (including transmission queues).

Existing applications do not need to be modified or updated nor is it necessary to set up Publish/Subscribe to use MQMR. MQMR is deployed and configured at the queue manager level; hence, no application changes are required.

MQMR is an MQ API Exit that operates with IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix, IBM i (OS/400) and Linux platforms.

On AIX, HP-UX, Linux, Solaris and Windows, MQMR can be configured and used with a non-default installation of MQ in a multi-install MQ environment.

Note: Raspberry Pi is a Linux ARM 32-bit OS (Operating System). Hence, simply follow the Linux 32-bit instructions for installing and using the solution on a Raspberry Pi.

New in MQMR v2.0.0, added 2 auxiliary programs:

- **MQ Queue To SQLite DB** (MQ2SDB) program will offload MQ messages to an SQLite database. See the **MQ2SDB Operation** manual for more information.
- **SQLite DB To MQ Queue** (SDB2MQ) program will load SQLite database rows into messages in an MQ queue. See the **SDB2MQ Operation** manual for more information.

1.2 Executive Summary

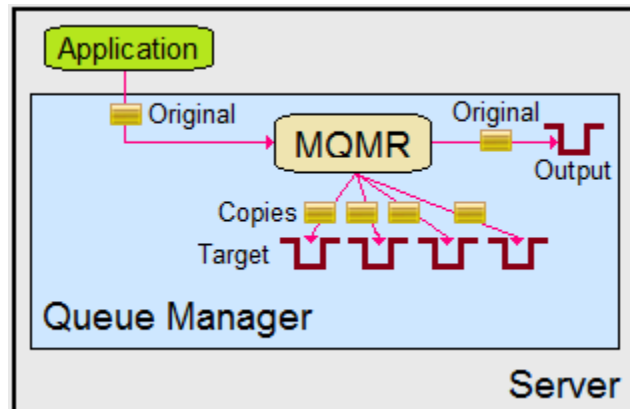
MQMR is an MQ API Exit. The MQ API Exit is available in 3 forms:

- Windows DLL
- Shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The major features of MQMR are as follows:

- All message data & MQMD fields written to the application's output queue will be cloned to a target queue(s)
- No application changes required
- Standard MQ feature, GET-with-Convert, is supported
- Provides high-level logging capability for message cloning

1.3 Context Diagram (Logical View)



1.4 Prerequisites

This section provides the minimum supported software levels. These prerequisites apply to server-side installations of MQ Message Replication.

1.4.1 Operating System

MQ Message Replication can be installed on any of the following supported servers:

1.4.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.4.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.4.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.4.1.4 Linux x86

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.4.1.5 Linux x86_64 (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.4.1.6 Linux on POWER

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.4.1.7 Linux on zSeries (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.4.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.4.1.9 Sun Solaris

- Solaris SPARC v10 or higher
- Solaris x86_64 v10 or higher

1.4.1.10 Windows

- Windows 2008, 2012 or 2016 Server (32-bit & 64-bit)
- Windows 7, 8, 8.1 or 10 (32-bit & 64-bit)

1.4.2 IBM MQ

- IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 (32-bit and 64-bit)

Operating System	MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
AIX v6.1 or higher	32-bit & 64-bit
HP-UX IA64 v11.23 or higher	32-bit & 64-bit
IBM i (OS/400)	64-bit
Linux x86	32-bit
Linux x86 64	32-bit & 64-bit
Linux on POWER	32-bit & 64-bit
Linux on zSeries	32-bit & 64-bit
Raspberry Pi ARM	32-bit
Solaris SPARC v10 & v11	32-bit & 64-bit
Solaris x86 64 v10 & v11	32-bit & 64-bit
Windows 2008, 2012, 2016, 7, 8, 8.1 & 10	32-bit & 64-bit

1.4.3 Windows 32-bit

The following is the software prerequisite for Windows 32-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist_x86.exe

1.4.4 Windows 64-bit

The following are the software prerequisite for Windows 64-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x64)
https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist_x64.exe

If local 32-bit applications connect in bindings mode to the queue manager then the following needs to be also installed:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist_x86.exe

2 Installing MQ Message Replication

This section describes how to install Capitalware's MQMR.

2.1 Windows Installation

To install MQMR on Windows, first unzip the **mqmr.zip** and then run the **mqmr_setup.exe** file. Follow the on-screen instructions and the API exit will be installed in the **C:\Capitalware\MQMR** directory (default installation).

The user may copy or ftp the **mqmr.dll**, **64\mqmr.dll**, **rotatelog.bat** and **mqmr.ini** files from one Windows server to another Windows server.

2.2 Linux 32-bit Installation

To install the 32-bit version of MQMR on Linux, first unzip the **mqmr.zip** and then select the appropriate TAR file for the target platform. You will find 2 TAR file in the original ZIP file:

- **Linux_x86/mqmr_linux.tar**
- **RaspberryPi_ARM/mqmr_raspberrypi_arm.tar**

Steps to Install MQMR:

1. ftp or copy the selected TAR file to the target platform to the **/var/mqm/** directory.
2. Un-tar the **mqmr_XXX.tar** file into the **/var/mqm/** sub-directory (xxx is either aix, hpux, solaris or linux)

```
cd /var/mqm/  
tar -xvf mqmr_XXX.tar
```

3. Change directory to **/var/mqm/exits/**
4. Next, do the following commands against **mqmr**:

```
chmod +x setmqmr.sh  
./setmqmr.sh
```

Note: It is important that both the License file and the IniFile are world-readable. Issue the following commands to change the file permissions:

```
cd /var/mqm/exits/  
chmod 644 mqmr_licenses.ini mqmr.ini
```

2.3 Unix and Linux 64-bit Installation

To install the 64-bit version of MQMR on Unix or Linux, first unzip the **mqmr.zip** and then select the appropriate TAR file for the target platform. You will find 7 TAR files in the original ZIP file:

- **AIX/mqmr_aix71_64.tar** for AIX v7.1 or higher
- **HPUX_IA64/mqmr_hpux64_ia64.tar**
- **Linux_x86_64/mqmr_linux_x86_64.tar**
- **Linux_POWER/mqmr_linux_power64.tar**
- **Linux_zSeries/mqmr_linux_zseries64.tar**
- **Solaris_SPARC/mqmr_solaris10_64.tar** for Solaris SPARC v10 or higher
- **Solaris_x86_64/mqmr_solaris_x86_64.tar**

Steps to Install MQMR:

1. ftp or copy the selected TAR file to the target platform to the **/var/mqm/** directory.
2. Un-tar the **mqmr_xxx.tar** file into the **/var/mqm/** sub-directory (xxx is either aix, hpux, solaris or linux)

```
cd /var/mqm/  
tar -xvf mqmr_xxx64.tar
```

3. Change directory to **/var/mqm/exits64/**
4. Next, do the following commands against **mqmr**:

```
chmod +x setmqmr.sh  
./setmqmr.sh
```

Note: mqmr_r shared library is not required for Solaris.

Note: It is important that both the License file and the IniFile are world-readable. Issue the following commands to change the file permissions:

```
cd /var/mqm/exits64/  
chmod 644 mqmr_licenses.ini mqmr.ini
```

2.4 IBM i Installation

To install the MQMR on IBM i, first unzip the **mqmr.zip** and then select the files in the IBM i (iSeries) directory.

- **mqmr.savf** is the IBM i 'Save File' that contains the library with the API exit.
- **mqmr_iseries.tar** is the IBM i IFS TAR file that contains a sample initialization file for the API Exit.

Steps to install the API Exit:

1. Log onto the target IBM i server and do the following command:

```
CRTSAVF FILE(QGPL/MQMR)
```

2. ftp the IBM i files to the IBM i server as follows:

```
ftp -s:mqmr_iseries.ftp iseries_hostname
```

The mqmr_iseries.ftp file contains the following ftp commands:

```
your-IBM i-userid  
your-IBM i-password  
  
binary  
cd QGPL  
put mqmr.savf  
  
quote SITE NAMEFMT 1  
  
cd /QIBM/UserData/mqm/  
put mqmr_iseries.tar  
quit
```

3. Log onto the target IBM i server and do the following commands:

```
RSTLIB SAVLIB(MQMR) DEV(*SAVF) SAVF(QGPL/MQMR)  
CLRSVF FILE(QGPL/MQMR)  
CHGOBJOWN OBJ(MQMR) OBJTYPE(*LIB) NEWOWN(QMQM)  
qsh  
cd /QIBM/UserData/mqm/  
tar -xvf mqmr_iseries.tar  
chown -R QMQM mqmr  
chmod -R 777 mqmr  
rm mqmr_iseries.tar
```

2.4.1 MQMR-GUI Installation

This section will describe how to install the MQMR-GUI. The user will find 2 files in the software package listed as follows:

- **MQMR-GUI/mqmrgui-wthjre.exe** (for Windows)
- **MQMR-GUI/mqmrgui.zip** (for Unix, Linux or macOS)

2.4.1.1 MQMR-GUI Installation on Windows

To install MQMR-GUI on Windows, run the **mqmrgui-wthjre.exe** file located in the MQMR-GUI directory. Follow the on-screen instructions and the program will be installed in the **C:\Capitalware\MQMR-GUI** directory (default installation).

2.4.1.2 MQMR-GUI Installation on Unix, Linux or macOS

To install MQMR-GUI on Unix or Linux, you will need to ftp or copy the selected TAR file to the target platform to the **/home/mqm/** directory. Next, one must telnet to the Unix, Linux or macOS server and 'cd' (change directory) to the **/home/mqm/** directory and unzip the archive file.

i.e. Do the following command:

```
unzip mqmrgui.zip
```

3 Configuring MQMR

This section describes how to configure the MQMR (MQ API Exit) on the support platforms.

Platform		Directory	Exit Module Name
Windows	32-bit	C:\Capitalware\MQMR\	mqmr.dll
Windows	64-bit	C:\Capitalware\MQMR\64\	mqmr.dll
Linux/Unix	32-bit	/var/mqm/exits/	mqmr
Linux/Unix	64-bit	/var/mqm/exits64/	mqmr

MQMR supports MQ's multi-install in a non-default directory.

After the user has configured the MQMR, the queue manager needs to be restarted.

3.1 API Exit

This section describes the necessary entries to enable the API Exit. The MQ Administrator will need to create a Local API Exit definition. This procedure differs depending on the platform.

Note: The value for the Data parameter cannot exceed a length of 32 characters.

3.1.1 Windows

On Windows, there are 2 ways to create the Local API Exit definition: via the command line or MQ Explorer. Note: Using MQ Explorer to add the API Exit is very easy.

3.1.1.1 Windows Command Line

First, the user will need to manually edit the queue manager's `qm.ini` file to update the `ExitsDefaultPath` and `ExitsDefaultPath64` fields. The `qm.ini` file will be located at: ***C:\Program Files (x86)\IBM\IBM MQ\Qmgrs\{QMgrName}\qm.ini***

```
ExitPath:  
ExitsDefaultPath=C:\Capitalware\MQMR;C:\Program Files (x86)\IBM\IBM MQ\exits  
ExitsDefaultPath64=C:\Capitalware\MQMR\64;C:\Program Files (x86)\IBM\IBM MQ\exits64
```

Next, create the Local API definition via the command line using the MQ `amqmdain` program. The MQAdmin can issue the following commands to create the Local API Exit definition:

```
amqmdain reg {QMgrName} -c add -s ApiExitLocal\MQMR -v Name=MQMR  
amqmdain reg {QMgrName} -c add -s ApiExitLocal\MQMR -v Module=mqmr.dll  
amqmdain reg {QMgrName} -c add -s ApiExitLocal\MQMR -v Data=C:\Capitalware\MQMR\mqmr.ini  
amqmdain reg {QMgrName} -c add -s ApiExitLocal\MQMR -v Sequence=3  
amqmdain reg {QMgrName} -c add -s ApiExitLocal\MQMR -v Function=EntryPoint
```

Where *{QMgrName}* is the name of the Queue Manager.

As a convenience, we have included a batch file called `mqmr_reg.bat` that includes all of the `amqmdain` commands. `mqmr_reg.bat` is located in the MQMR install directory and it accepts one parameter (`QMgrName`).

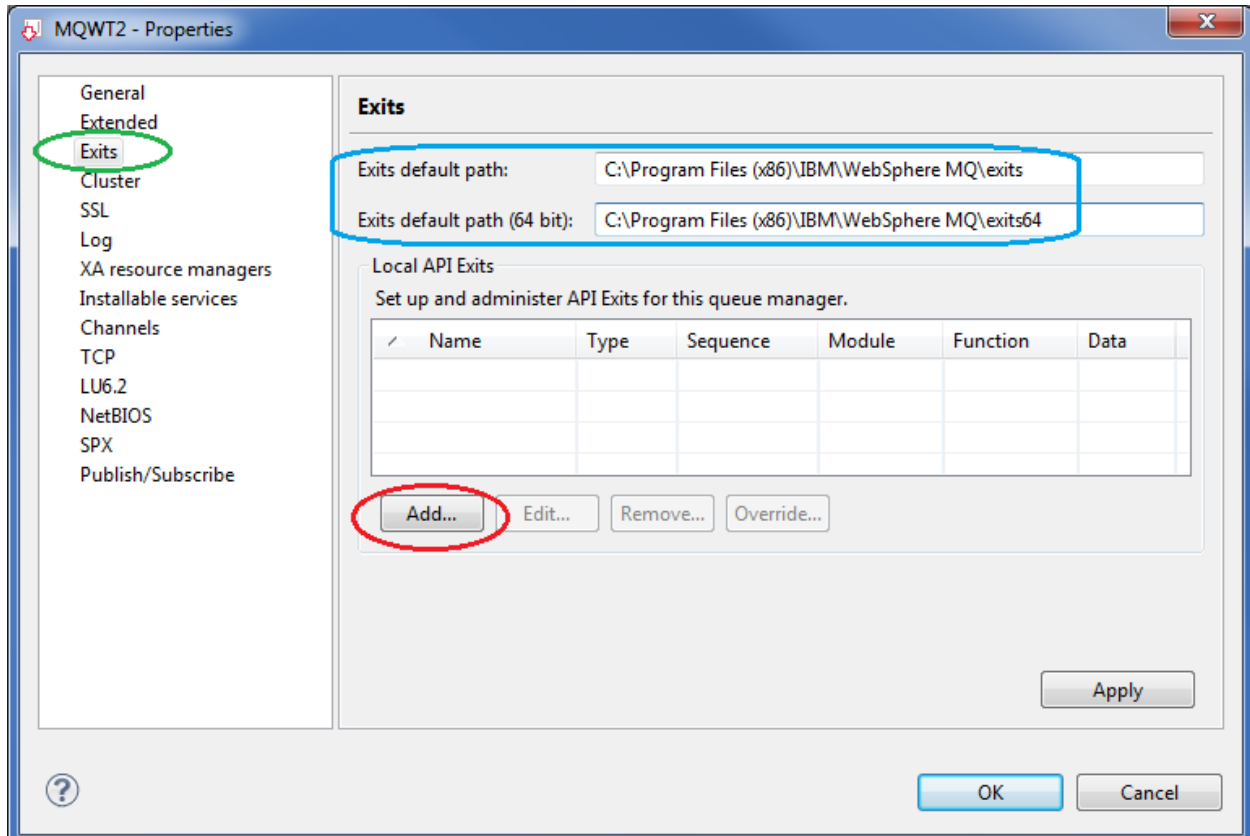
Open a Windows Command Prompt and issue the following commands to configure MQMR for a particular queue manager:

```
cd /D C:\Capitalware\MQMR\  
mqmr_reg.bat {QMgrName}
```

3.1.1.2 Windows MQ Explorer

To create a Local API Exit definition using MQ Explorer, do the following:

- Start MQ Explorer
- Right-click on the queue manager name
- Select **Properties** from the popup menu
- Select **Exits** in the left panel of the properties window
- Update the 'Exit default path' and 'Exit default path (64-bit)' with the MQMR install path. i.e. C:\Capitalware\MQMR and C:\Capitalware\MQMR\64
- Click the **Add** button.



The user is required to input the following values into the 5 fields of the Add API Exit window:

Field Name	Value
Name	MQMR
Function	EntryPoint
Module	mqmr.dll
Data	C:\Capitalware\MQMR\mqmr.ini
Sequence Number	3

The screenshot shows a dialog box titled "Edit API Exit 'MQMR'". It contains the following fields and controls:

- Name:** A text box containing "MQMR".
- Function:** A text box containing "EntryPoint".
- Module:** A text box containing "mqmr.dll" and a "Browse..." button to its right.
- Data:** A checked checkbox followed by a text box containing "C:\Capitalware\MQMR\mqmr.ini".
- Sequence number:** A spin box containing the number "3".
- At the bottom right, there are "OK" and "Cancel" buttons.
- At the bottom left, there is a help icon (a question mark in a circle).

Click **OK** when the information has been inputted.

3.1.2 For Linux 32-bit

Edit the *qm.ini* for the queue manager that MQMR is being applied to. The *qm.ini* file is located at `/var/mqm/qmgrs/{QMgrName}/qm.ini`

Make sure the ExitPath stanza exists in the qm.ini and then add the ApiExitLocal stanza as given below:

```
ExitPath:
  ExitsDefaultPath=/var/mqm/exits/

ApiExitLocal:
  Name=MQMR
  Sequence=3
  Function=EntryPoint
  Module=mqmr
  Data=/var/mqm/exits/mqmr.ini
```

Note: If the user has not installed MQMR in `/var/mqm/exits/` directory then the ExitsDefaultPath field will need to be updated with the path to the shared library.

3.1.3 Unix and Linux 64-bit

Edit the *qm.ini* for the queue manager that MQMR is being applied to. The *qm.ini* file is located at `/var/mqm/qmgrs/{QMgrName}/qm.ini`

Make sure the ExitPath stanza exists in the qm.ini and then add the ApiExitLocal stanza as given below:

```
ExitPath:
  ExitsDefaultPath=/var/mqm/exits/
  ExitsDefaultPath64=/var/mqm/exits64/

ApiExitLocal:
  Name=MQMR
  Sequence=3
  Function=EntryPoint
  Module=mqmr
  Data=/var/mqm/exits64/mqmr.ini
```

Note: If the user has not installed MQMR in `/var/mqm/exits/` and `/var/mqm/exits64/` directories then the ExitsDefaultPath and ExitsDefaultPath64 fields will need to be updated with the path to the shared library.

3.1.4 IBM i

On IBM i, edit the *qm.ini* for the queue manager that MQMR is being applied to. The *qm.ini* file is located at `/QIBM/UserData/mqm/qmgrs/{QMgrName}/qm.ini`

Add the `ApiExitLocal` stanza as given below:

```
ApiExitLocal:  
  Name=MQMR  
  Sequence=3  
  Function=EntryPoint  
  Module=MQMR/MQMR  
  Data=/QIBM/UserData/mqm/mqmr/mqmr.ini
```

3.2 File Paths

Data field of the ApiExitLocal stanza must NOT exceed 32 characters. In order to work with this limitation, MQMR supports 3 ways to specify an IniFile path: absolute path, relative path and environment variable.

Note: The IniFile path that is determined by MQMR API Exit will also be used for the following IniFile keywords (if no pathing is specified for these keywords): **LicenseFile** and **LogFile**.

3.2.1 Absolute Path

Absolute pathing (specifying the complete path) for the Data field works on Linux, Unix and Windows platforms.

E.g. Windows

```
Data=C:\Capitalware\MQMR\mqmr.ini
```

Hence, MQMR will use the following path as the IniFile path:
C:\Capitalware\MQMR

3.2.2 Relative Path

Relative pathing for the Data field is supported on Linux, IBM i, Unix and Windows platforms. MQMR will extract the path from Data field and prefix it to the IniFile specified in the Data field in order to locate the IniFile.

For Windows:

```
C:\Capitalware\MQMR\
```

For IBM MQ 32-bit on Unix and Linux:

```
/var/mqm/exits/
```

For IBM MQ 64-bit on Unix and Linux:

```
/var/mqm/exits64/
```

For IBM MQ on IBM i:

```
/QIBM/UserData/mqm/mqmr/
```

E.g. Unix

```
Data=mqmr.ini
```

Hence, MQMR will use the following path as the IniFile path:

E.g. 32-bit

```
/var/mqm/exits/
```

E.g. 64-bit

```
/var/mqm/exits64/
```

3.2.3 Environment Variables

3.2.3.1 Global Environment Variable

MQMR supports the use of the MQMR_HOME environment variable which holds the directory path information. MQMR_HOME environment variable is supported on Linux, IBM i, Unix and Windows platforms.

e.g. Unix

```
export MQMR_HOME=/really/long/path/MQHA/QMgrName/data/
```

```
Data=mqmr.ini
```

Hence, MQMR will use the following path as the IniFile path:
/really/long/path/MQHA/QMgrName/data/

3.2.3.2 Queue Manager Specific Environment Variable

MQMR supports the use of the MQMR_HOME environment variable post-fixed with the queue manager name which holds the directory path information. MQMR_HOME environment variable post-fixed with the queue manager name is supported on Linux, IBM i, Unix and Windows platforms.

e.g. Unix with a queue manager name of MQA1

```
export MQMR_HOME_MQA1=/really/long/path/MQHA/QMgrName/data2/
```

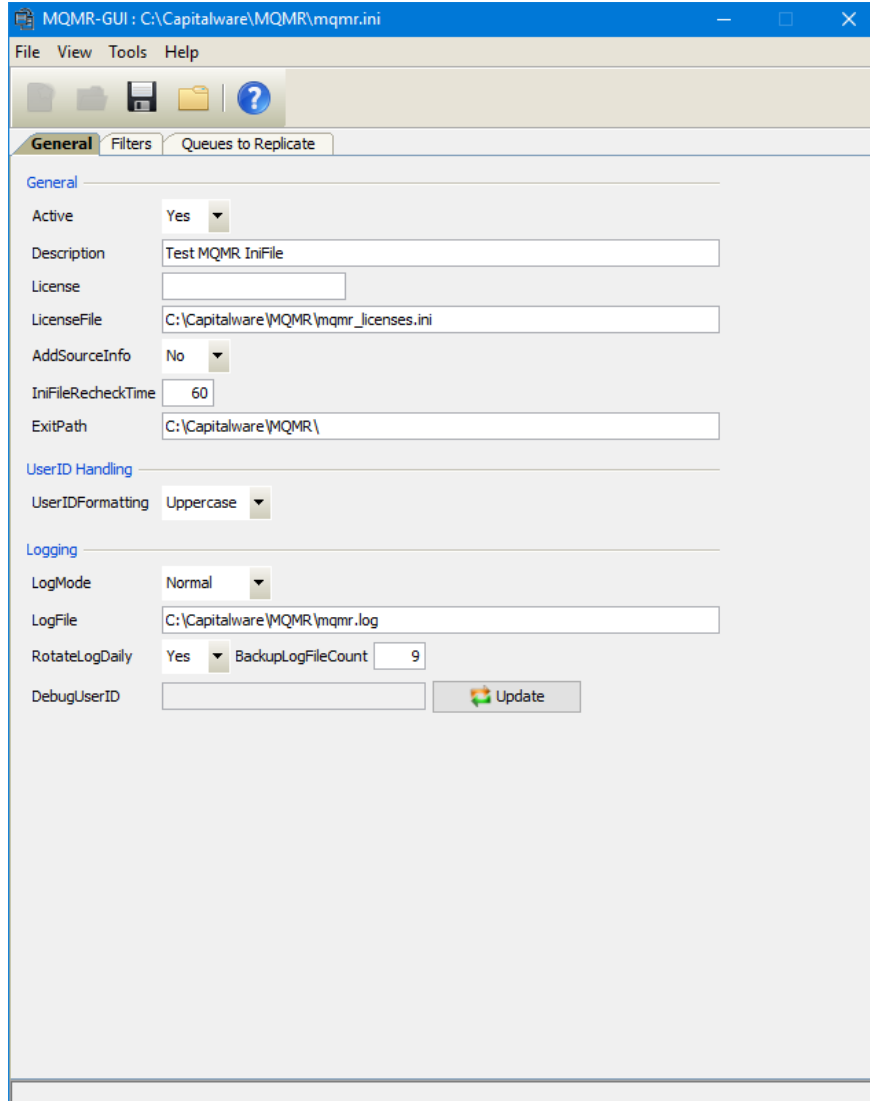
```
Data=mqmr.ini
```

Hence, MQMR will use the following path as the IniFile path:
/really/long/path/MQHA/QMgrName/data2/

Note: If both environment variables are specified then the queue manager specific environment variable will be used.

3.3 MQMR-GUI

This section briefly describes the new graphical program called MQMR-GUI. This program assists the user in creating and managing their MQMR IniFiles. For more information, please see the *MQMR-GUI User Guide* manual.



4 IniFile Keywords (Global Values)

This section describes IniFile keywords.

4.1 Active

MQMR can be enabled or disabled by the IniFile's property keyword 'Active'. When Active has a value of Y, MQMR is activated. The default value is Y.

```
Active=Y
```

4.2 ExitPath

This section describes the use of the keyword ExitPath to explicitly set the API exit path. By setting ExitPath to a particular path, the default value for ExitPath is overridden. The defaults are as follows:

For Windows:

```
ExitPath=C:\Capitalware\MQMR\
```

For IBM MQ 32-bit on Unix and Linux:

```
ExitPath=/var/mqm/exits/
```

For IBM MQ 64-bit on Unix and Linux:

```
ExitPath=/var/mqm/exits64/
```

For IBM MQ on IBM i:

```
ExitPath=/QIBM/UserData/mqm/mqmr/
```

```
ExitPath=D:\Vendor\Ccapitalware\MQMR\
```

4.3 ExcludeApplications

This section describes how to explicitly exclude applications from the encryption/decryption processing when an application has connected to the queue manager. MQMR will look up the regular expression patterns from the **ExcludeApplications** keyword in order to determine if the application name matches any of the specified expression patterns.

If there is a match, the application is excluded from encryption/decryption process. Each regular expression pattern is separated from the next pattern by a semi-colon (;).

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more)
- '?' matches any single character
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z, a-z)
- [SET] matches any of the characters in the specified set
- [!SET] or [^SET] matches any character except those specified in the set (negation).

A SET can be composed of characters or ranges. A range is in the form: 'character – character' (i.e. 0-9 or A-Z). Although this is the simplest range allowed in the [] pattern, more complex inclusive ranges such as [0-9a-zA-Z] are allowed. [0-9a-zA-Z] specifies that the character can be 0 through 9 **or** a through z **or** A through Z. Other characters are allowed (ie. 8 bit characters) if your system supports them.

In order to suppress the special syntactic significance of any of these characters '[] * ? # @ ! ^ - \', a backslash ('\') must precede the special character.

Note: *ExcludeApplications must NOT exceed 2048 characters.*

To authorize MQMR decryption by application name, you need 2 keywords in the IniFile:

- **UseExcludeApplications** activates this feature. If it has a value of Y, MQMR will exclude applications from the encryption/decryption process.
- **ExcludeApplications** specifies the list of application names that are to be excluded.

```
UseExcludeApplications=Y
ExcludeApplications=abc*;gadget;xyz*
```

4.4 ExcludeUserIDs

This section describes how to explicitly exclude UserIDs from the encryption/decryption processing when a UserID has connected to the queue manager. MQMR will look up the regular expression patterns from the **ExcludeUserIDs** keyword in order to determine if the UserID matches any of the specified expression patterns.

If there is a match, the UserID is excluded from encryption/decryption process. Each regular expression pattern is separated from the next pattern by a semi-colon (;).

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more)
- '?' matches any single character
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z, a-z)
- [SET] matches any of the characters in the specified set
- [!SET] or [^SET] matches any character except those specified in the set (negation).

A SET can be composed of characters or ranges. A range is in the form: 'character – character' (i.e. 0-9 or A-Z). Although this is the simplest range allowed in the [] pattern, more complex inclusive ranges such as [0-9a-zA-Z] are allowed. [0-9a-zA-Z] specifies that the character can be 0 through 9 **or** a through z **or** A through Z. Other characters are allowed (ie. 8 bit characters) if your system supports them.

In order to suppress the special syntactic significance of any of these characters '[] * ? # @ ! ^ - \', a backslash ('\') must precede the special character.

Note: *ExcludeUserIDs must NOT exceed 2048 characters.*

To authorize MQMR decryption by UserID name, you need 2 keywords in the IniFile:

- **UseExcludeUserIDs** activates this feature. If it has a value of Y, MQMR will exclude UserIDs from the encryption/decryption process.
- **ExcludeUserIDs** specifies the list of UserID names that are to be excluded.

```
UseExcludeUserIDs=Y
ExcludeUserIDs=fred;barney;wilma;betty
```


4.5 LicenseFile

This section will describe how to have a file that contains all of the user's MQMR license keys.

The format of the LicenseFile is similar to an IniFile or properties file where each keyword has an associated value. Each keyword and its value are on a separate line. The format is as follows:

QMgrName = License_Key

Example:

```
MQA1 = 10M0-AAAA-BBBBBBBB  
MQB1 = 10M0-XXXX-CCCCCCCC
```

If the queue manager name is not found in the LicenseFile then the License keyword will be used to retrieve the license key value.

The following are the default values for LicenseFile:

For Windows:

LicenseFile=C:\Capitalware\MQMR\mqmr_licenses.ini

For IBM MQ 32-bit on Unix and Linux:

LicenseFile=/var/mqm/exits/mqmr_licenses.ini

For IBM MQ 64-bit on Unix and Linux:

LicenseFile=/var/mqm/exits64/mqmr_licenses.ini

For IBM MQ on IBM i:

LicenseFile=/QIBM/UserData/mqm/mqmr/mqmr_licenses.ini

4.6 License Key

This section will describe how to license MQ Message Replication to a particular queue manager.

Note: The License keyword is not required if the user has implemented the LicenseFile keyword or the License file actually exists in the default location.

Your license will look something like: 10M0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).

```
License=10M0-AAAA-BBBBBBBB
```

4.7 Logging

This section describes the necessary entries to enable MQMR to record log information. To enable and control logging, there are 4 keywords in the IniFile:

1. **LogMode** specifies what type of logging the user wishes to have. LogMode supports 4 values [Q/N/V/D] where Q is Quiet, N is Normal, V is Verbose and D is Debug. The default value is N.
2. **LogFile** specifies the location of the log file. The default values are as follows:

For Windows:

```
LogFile=C:\Capitalware\MQMR\mqmr.log
```

For IBM MQ 32-bit on Linux:

```
LogFile=/var/mqm/mqmr/mqmr.log
```

For IBM MQ 64-bit on Unix and Linux:

```
LogFile=/var/mqm/mqmr/mqmr.log
```

For IBM MQ on IBM i:

```
LogFile=/QIBM/UserData/mqm/mqmr/mqmr.log
```

Token Replacement for LogFile keyword:

- **%QM%** - Substitutes the name of the queue manager
- **%UID%** - Substitutes the UserID
- **%PID%** - Substitutes the Process ID
- **%TID%** - Substitutes the Thread ID

3. **RotateLogDaily** specifies whether or not the log files will be rotated on a daily basis. A Y value for 'RotateLogDaily' will activate this feature; otherwise, the log files will left as is. The default value is Y.

In other words, it is possible to keep up to 9 backup log files. The first connection request after midnight (and not at midnight) will cause it to roll/rotate the log files. If there are already 9 backup log files, the ninth backup log file will be deleted and 8 becomes 9, 7 becomes 8, etc...

4. **BackupLogFileCount** specifies the number of backup log files that should be kept by MQMR. The default value is 9. This keyword is only used if RotateLogDaily is set to 'Y'.

4.8 AddSourceInfo

AddSourceInfo specifies if MQMR is will add the source queue and queue manager name to the message as a message property. AddSourceInfo supports 2 values [Y / N]. The default value is N.

```
AddSourceInfo=Y
```

Message Properties Example:

```
CW.MQMR.Src.QMgrName = MQWT1  
CW.MQMR.Src.QName = TEST.Q1
```

5 Message Replication by Queue

This section describes IniFile keywords to activate MQMR message replication against queues of a queue manager.

5.1 What Value is to be used for Section Name/Queue Name?

The section will describe how to determine what value should be used for the Section Name/Queue Name. MQMR uses the "Resolved Queue" value when processing what queue to replicate.

5.1.1 Local Queue

For a local queue, simply use the local queue name for the Section Name/Queue Name.

5.1.2 Cluster Queue

For a cluster queue, simply use the cluster queue name for the Section Name/Queue Name.

5.1.3 Alias Queue

For an alias queue, if the client application opened the queue with:

- MQOD version 1 or 2, the user **MUST** use the '*QUEUE*' value from the alias queue definition for the Section Name/Queue Name.
- MQOD version 3 or higher, the user **MUST** use the '*TARGET*' value from the alias queue definition for the Section Name/Queue Name.

Example:

```
DISPLAY QALIAS(TEST.Q1.AL)
  1 : DISPLAY QALIAS(TEST.Q1.AL)
AMQ8409: Display Queue details.
  QUEUE(TEST.Q1.AL)
  ALTDATE(2012-12-04)
  TARGET(TEST.Q1)
  CLUSTER( )
  CLWLPRTY(0)
  CLWLPRTY(0)
  CUSTOM( )
  DEFPRTY(0)
  DEFPRSP(SYNC)
  DESCR( )
  PUT(ENABLED)
  PROPCTL(COMPAT)
  TARGTYPE(Queue)
  TYPE(QALIAS)
  ALTTIME(15.51.41)
  CLUSNL( )
  CLWLPRTY(0)
  CUSTOM( )
  DEFPRTY(0)
  DEFPRSP(SYNC)
  DESCR( )
  PUT(ENABLED)
  SCOPE(QMGR)
```

Note: UseResolvedQueueName keyword controls which value MQMR will use to match the queue name on.

5.1.4 Remote Queue

For a remote queue, if the client application opened the queue with:

- MQOD version 1 or 2, the user MUST use the '*QUEUE*' value from the alias queue definition for the Section Name/Queue Name.
- MQOD version 3 or higher, the user MUST use the '*RNAME*' value from the alias queue definition for the Section Name/Queue Name.

Example:

```
DISPLAY QREMOTE(TEST.Q2.RQ)
  1 : DISPLAY QREMOTE(TEST.Q2.RQ)
AMQ8409: Display Queue details.
  QUEUE(TEST.Q2.RQ)
  ALTDATE(2012-12-04)
  CLUSNL( )
  CLWLPRTY(0)
  CUSTOM( )
  DEFPRTY(0)
  DEFPRESP(SYNC)
  PUT(ENABLED)
  RNAME(TEST.Q2)
  XMITQ(MQWT2.XMIT)
  TYPE(QREMOTE)
  ALTTIME(15.51.41)
  CLUSTER( )
  CLWLRANK(0)
  DEFBIND(OPEN)
  DEFPSIST(NO)
  DESCR( )
  RQMNAME(MQWT2)
  SCOPE(QMGR)
```

Note: UseResolvedQueueName keyword controls which value MQMR will use to match the queue name on.

5.2 Section Name for a Queue

The section stanza must begin in column 1 with a left square bracket '[' followed by a "Q:" and end with a right square bracket ']'. Between the colon and the right square bracket, the user specifies the queue name. Note: No wild-cards are allowed.

```
[Q:TEST.Q01]
```

5.2.1 TargetQueues

TargetQueues specifies a list of queues that the clone messages will be written to. Separate each queue in the list with a semi-colon (;).

```
TargetQueues=TEST.Q10;TEST.Q11;TEST.Q12;TEST.Q13
```

5.2.2 Context

Context indicates how MQMR will handle the clone message's MQMD. Context supports 2 values [S/D]. The default value is S.

- **S** means that MQMR will **set** the context of the clone message's MQMD to be the same as the original message's MQMD.
- **D** means that MQMR will use the default context for the clone message's MQMD

```
Context=S
```

5.2.3 AddSourceInfo

AddSourceInfo specifies if MQMR is will add the source queue and queue manager name to the message as a message property. AddSourceInfo supports 2 values [Y / N]. The default value is N.

```
AddSourceInfo=Y
```

Message Properties Example:

```
CW.MQMR.Src.QMgrName = MQWT1  
CW.MQMR.Src.QName = TEST.Q1
```

5.2.4 UseExpiry

UseExpiry indicates if MQMR will overwrite the cloned/replicated message's Expiry value. The original message's Expiry value is NOT altered. UseExpiry supports 2 values [Y/N]. The default value is N.

```
UseExpiry=N
```

5.2.5 Expiry

Expiry value (in milliseconds) used by MQMR to set the cloned/replicated message's Expiry value. The original message's Expiry value is NOT altered. The default value is '-1' (MQEI_UNLIMITED).

```
Expiry=-1
```

5.2.6 UseResolvedQueueName

UseResolvedQueueName controls what value MQMR will use to match queue name on. If it is set to 'Y', MQMR will use the resolve queue values, see section 5.1 for more details. If it is set to 'N' then for alias queues and remotes queues, MQMR will use the object name and not TARGET (alias queue) or RNAME (remote queue) value. UseResolvedQueueName supports 2 values [Y / N]. The default value is Y.

```
UseResolvedQueueName=Y
```

5.2.7 ForMQPut1DoMQPut1

ForMQPut1DoMQPut1 indicates how MQMR will put a message to the target queues when an application uses MQPUT/1 to put a message to an application's output queue. ForMQPut1DoMQPut1 supports 2 values [Y/N]. The default value is Y.

```
ForMQPut1DoMQPut1=Y
```

5.2.8 UseSyncPoint

UseSyncPoint indicates that MQMR should perform the MQPUT of the cloned message to the targets queues under a SyncPoint (unit of Work). UseSyncPoint supports 2 values [Y / N]. The default value is N.

Note: Do **NOT** use this option if the client application is not using a SyncPoint (it must issue either MQBACK or MQCMIT).

```
UseSyncPoint=Y
```

5.2.9 CollapseMQMDE

CollapseMQMDE indicates that MQMR should change the MQMD version from 1 to 2 (of the cloned message), move the MQMDE values from the embedded MQMDE structure and then remove the embedded MQMDE structure. CollapseMQMDE supports 2 values [Y / N]. The default value is N.

```
CollapseMQMDE=Y
```

5.2.10 ClearRO

ClearRO indicates that MQMR set the Report field to 'MQRO_NONE'. ClearRO supports 2 values [Y / N]. The default value is N.

```
ClearRO=Y
```

5.2.11 ClearROCOA

ClearROCOA indicates that MQMR will clear the MQRO_COA value from the Report field. ClearROCOA supports 2 values [Y / N]. The default value is N.

```
ClearROCOA=Y
```

5.2.12 ClearROCOD

ClearROCOD indicates that MQMR will clear the MQRO_COD value from the Report field. ClearROCOD supports 2 values [Y / N]. The default value is N.

```
ClearROCOD=Y
```


5.2.13 ClearROPAN

ClearROPAN indicates that MQMR will clear the MQRO_PAN value from the Report field. ClearROPAN supports 2 values [Y / N]. The default value is N.

```
ClearROPAN=Y
```

5.2.14 ClearRONAN

ClearRONAN indicates that MQMR will clear the MQRO_NAN value from the Report field. ClearRONAN supports 2 values [Y / N]. The default value is N.

```
ClearRONAN=Y
```

5.2.15 ClearROException

ClearROException indicates that MQMR will clear the MQRO_EXCEPTION value from the Report field. ClearROException supports 2 values [Y / N]. The default value is N.

```
ClearROException=Y
```

5.2.16 ClearROExpiration

ClearROExpiration indicates that MQMR will clear the MQRO_EXPIRATION value from the Report field. ClearROExpiration supports 2 values [Y / N]. The default value is N.

```
ClearROExpiration=Y
```

5.2.17 SkipCOA

SkipCOA indicates that MQMR will not clone a message that has the Feedback field set to MQFB_COA. SkipCOA supports 2 values [Y / N]. The default value is N.

```
SkipCOA=Y
```

5.2.18 SkipCOD

SkipCOD indicates that MQMR will not clone a message that has the Feedback field set to MQFB_COD. SkipCOD supports 2 values [Y / N]. The default value is N.

```
SkipCOD=Y
```

5.2.19 SkipPAN

SkipPAN indicates that MQMR will not clone a message that has the Feedback field set to MQFB_PAN. SkipPAN supports 2 values [Y / N]. The default value is N.

```
SkipPAN=Y
```

5.2.20 SkipNAN

SkipNAN indicates that MQMR will not clone a message that has the Feedback field set to MQFB_NAN. SkipNAN supports 2 values [Y / N]. The default value is N.

```
SkipNAN=Y
```

5.2.21 SkipExpiration

SkipExpiration indicates that MQMR will not clone a message that has the Feedback field set to MQFB_EXPIRATION. SkipExpiration supports 2 values [Y / N]. The default value is N.

```
SkipExpiration=Y
```

5.3 Example

If the queue name is called “TEST.Q01” then the user needs to prefix “Q:”, so that the stanza looks like: [Q:TEST.Q01]. Here is an IniFile example with 3 different groups of queues:

```
[Q:TEST.Q01]
Context=S
TargetQueues=TEST.Q10;TEST.Q11;TEST.Q12;TEST.Q13

[Q:ABC.HR.Q]
Context=S
TargetQueues=ABC.HR.FRED;ABC.HR.BARNEY;ABC.HR.WILMA;ABC.HR.BETTY

[Q:BANK.DEPOSIT.CREDIT]
Context=S
TargetQueues=BANK.DEPOSIT.TEMP01;BANK.DEPOSIT.TEMP02;BANK.DEPOSIT.TEMP03
```

6 Appendix A – Summary of IniFile

The sample IniFile below is the mqmr.ini file supplied for Windows.

```
[default]
Active=Y
LogMode=N
LogFile=C:\Capitalware\MQMR\mqmr.log

[Q:TEST.Q01]
Context=S
TargetQueues=TEST.Q10;TEST.Q11;TEST.Q12;TEST.Q13
UseExpiry=N
```

Note: Keywords are case sensitive.

The IniFile supports the following keywords and their values:

Keyword	Description of Server-side keywords
Active	Active specifies if MQMR is enabled or disabled. Active supports 2 values [Y / N]. The default value is Y. e.g. Active=N
AddSourceInfo	AddSourceInfo specifies if MQMR is will add the source queue and queue manager name to the message as a message property. AddSourceInfo supports 2 values [Y / N]. The default value is N. e.g. AddSourceInfo=Y
BackupLogFileCount	BackupLogFileCount specifies the number of backup logfiles that MQMR will be keeping. The default value is 9. e.g. BackupLogFileCount=9
ClearRO	ClearRO specifies that MQMR is to set the Report field to 'MQRO_NONE'. ClearRO supports 2 values [Y / N]. The default value is N. e.g. ClearRO=Y

Keyword	Description of Server-side keywords
ClearROCOA	<p>ClearROCOA specifies that MQMR will clear the MQRO_COA value from the Report field. ClearROCOA supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearROCOA=Y</p>
ClearROCOD	<p>ClearROCOD specifies that MQMR will clear the MQRO_COD value from the Report field. ClearROCOD supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearROCOD=Y</p>
ClearROPAN	<p>ClearROPAN specifies that MQMR will clear the MQRO_PAN value from the Report field. ClearROPAN supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearROPAN=Y</p>
ClearRONAN	<p>ClearRONAN specifies that MQMR will clear the MQRO_NAN value from the Report field. ClearRONAN supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearRONAN=Y</p>
ClearROException	<p>ClearROException specifies that MQMR will clear the MQRO_EXCEPTION value from the Report field. ClearROException supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearROException=Y</p>
ClearROExpiration	<p>ClearROExpiration specifies that MQMR will clear the MQRO_EXPIRATION value from the Report field. ClearROExpiration supports 2 values [Y / N]. The default value is N.</p> <p>e.g. ClearROExpiration=Y</p>

Keyword	Description of Server-side keywords
CollapseMQMDE	<p>CollapseMQMDE indicates that MQMR should change the MQMD version from 1 to 2 (of the cloned message), move the MQMDE values from the embedded MQMDE structure and then remove the embedded MQMDE structure. CollapseMQMDE supports 2 values [Y / N]. The default value is N.</p> <p>e.g. CollapseMQMDE=Y</p>
Context	<p>Context indicates how MQMR will handle the clone message's MQMD. Context supports 2 values [S/D]. The default value is S.</p> <ul style="list-style-type: none"> • S means that MQMR will set the context of the clone message's MQMD to be the same as the original message's MQMD. • D means that MQMR will use the default context for the clone message's MQMD <p>e.g. Context=S</p>
DebugUserID	<p>DebugUserID specifies a list of UserID which should have debug logging turned on for.</p> <p>e.g. DebugUserID=fred;barney</p>
ExcludeApplications	<p>ExcludeApplications specifies which particular applications should not be included.</p> <p>e.g. ExcludeApplications=test*;gadget</p> <p>Note: Only used if UseExcludeApplications is set to 'Y'.</p>
ExcludeUserIDs	<p>ExcludeUserIDs specifies which particular UserIDs should not be included.</p> <p>e.g. ExcludeUserIDs=fred;barney</p> <p>Note: Only used if UseExcludeUserIDs is set to 'Y'.</p>

Keyword	Description of Server-side keywords
Expiry	<p>Expiry value (in milliseconds) used by MQMR to set the cloned/replicated message's Expiry value. The original message's Expiry value is NOT altered. The default value is '-1' (MQEI_UNLIMITED).</p> <p>e.g. Expiry=5000</p>
ForMQPut1DoMQPut1	<p>ForMQPut1DoMQPut1 indicates how MQMR will put a message to the target queues when an application uses MQPUT1 to put a message to an application's output queue. ForMQPut1DoMQPut1 supports 2 values [Y/N]. The default value is Y.</p> <p>e.g. ForMQPut1DoMQPut1=Y</p>
IniFileRecheckTime	<p>IniFileRecheckTime specifies the amount, in seconds, before the IniFile is checked whether it has changed or not. The default value is 60.</p> <p>e.g. IniFileRecheckTime=60</p>
License	<p>License specifies the queue manager's license key. Your license key will look something like: 10S0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).</p> <p>e.g. License=10M0-AAAA-BBBBBBBB</p>

Keyword	Description of Server-side keywords
LicenseFile	<p>LicenseFile specifies the location of License file that contains all of the customer's license keys.</p> <p>The default values for LicenseFile are as follows:</p> <p>For Windows: LicenseFile=C:\Capitalware\MQMR\mqmr_licenses.ini</p> <p>For IBM MQ 32-bit on Unix and Linux: LicenseFile=/var/mqm/exits/mqmr_licenses.ini</p> <p>For IBM MQ 64-bit on Unix and Linux: LicenseFile=/var/mqm/exits64/mqmr_licenses.ini</p> <p>For IBM MQ on IBM i: LicenseFile=/QIBM/UserData/mqm/mqmr/mqmr_licenses.ini</p> <p>e.g. LicenseFile=/var/mqm/exits64/mqmr_licenses.ini</p>
LogFile	<p>LogFile specifies the location of the log file. The defaults are as follows:</p> <p>For Windows: LogFile=C:\Capitalware\MQMR \mqmr.log</p> <p>For IBM MQ 32-bit on Unix and Linux: LogFile=/var/mqm/audit/mqmr.log</p> <p>For IBM MQ 64-bit on Unix and Linux: LogFile=/var/mqm/audit/mqmr.log</p> <p>For IBM MQ on IBM i: LogFile=/QIBM/UserData/mqm/mqmr/mqmr.log</p>
LogMode	<p>LogMode specifies what type of logging the user wishes to have. LogMode supports 4 values [Q / N / V / D] where Q is Quiet, N is Normal, V is Verbose and D is Debug. The default value is N.</p> <p>e.g. LogMode=N</p>
RotateLogDaily	<p>RotateLogDaily specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y.</p> <p>e.g. RotateLogDaily=Y</p>

Keyword	Description of Server-side keywords
SkipCOA	<p>SkipCOA specifies that MQMR will not clone a message that has the Feedback field set to MQFB_COA. SkipCOA supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SkipCOA=Y</p>
SkipCOD	<p>SkipCOD specifies that MQMR will not clone a message that has the Feedback field set to MQFB_COD. SkipCOD supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SkipCOD=Y</p>
SkipPAN	<p>SkipPAN specifies that MQMR will not clone a message that has the Feedback field set to MQFB_PAN. SkipPAN supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SkipPAN=Y</p>
SkipNAN	<p>SkipNAN specifies that MQMR will not clone a message that has the Feedback field set to MQFB_NAN. SkipNAN supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SkipNAN=Y</p>
SkipExpiration	<p>SkipExpiration specifies that MQMR will not clone a message that has the Feedback field set to MQFB_EXPIRATION. SkipExpiration supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SkipExpiration=Y</p>
TargetQueues	<p>TargetQueues specifies a list of queues that the clone messages will be written to.</p> <p>e.g. TargetQueues=TEST.Q10;TEST.Q11;TEST.Q12;TEST.Q13</p>
UseExcludeApplications	<p>UseExcludeApplications specifies whether or not ExcludeApplications should be activated. UseExcludeApplications supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UseExcludeApplications=Y</p>

Keyword	Description of Server-side keywords
UseExcludeUserIDs	<p>UseExcludeUserIDs specifies whether or not ExcludeQueues should be activated. UseExcludeUserIDs supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UseExcludeUserIDs=Y</p>
UseExpiry	<p>UseExpiry indicates if MQMR will overwrite the cloned/replicated message's Expiry value. The original message's Expiry value is NOT altered. UseExpiry supports 2 values [Y/N]. The default value is N.</p> <p>e.g. UseExpiry=Y</p>
UseResolvedQueueName	<p>UseResolvedQueueName controls what value MQMR will use to match queue name on. If it is set to 'Y', MQMR will use the resolve queue values, see section 5.1 for more details. If it is set to 'N' then for alias queues and remotes queues, MQME will use the object name and not TARGET (alias queue) or RNAME (remote queue) value. UseResolvedQueueName supports 2 values [Y / N]. The default value is Y.</p> <p>e.g. UseResolvedQueueName=Y</p>
UseSyncPoint	<p>UseSyncPoint specifies if MQMR should perform the MQPUT of the cloned message to the targets queues under a SyncPoint (unit of Work). UseSyncPoint supports 2 values [Y / N]. The default value is N.</p> <p>Note: Do NOT use this option if the client application is not using a SyncPoint (it must issue either MQBACK or MQCMIT).</p> <p>e.g. UseSyncPoint=Y</p>

7 Appendix B – MQMR Upgrade Procedures

To upgrade an existing installation of MQMR from an older version to a newer version, please do the following:

7.1.1 Windows Upgrade

- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQMR API Exit
- Backup all MQMR IniFiles in the MQMR install directory
- If MQMR was installed using the Windows Installer then
 - Click the **Start -> All Programs -> Control Panel -> Add or Remove Programs**, select MQMR from the list and click the **Remove** button then follow the prompts to remove it
 - Run the **mqmr-setup.exe** file from the **Windows** directory to install the new version
- Otherwise, copy the following files (latest version) to the MQMR install directory:
 - mqmr.dll
 - mqmr_reg.bat
 - rotatelog.bat
- Restore the MQMR IniFiles if they were altered/deleted.
- Restart the queue manager
- Restart your MQ applications and any monitoring tools

7.1.2 Linux 32-bit Upgrade

- Log in under the mqm account
- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQMR API Exit
- Backup all MQMR IniFiles in the MQMR install directory
- Copy the appropriate tar file to the **/var/mqm/** directory
- Un-tar the contents of the tar file.
e.g. For Linux, do the following command:
tar -xvf mqmr_linux_x86.tar
- Run the script as follows:
./setmqmr.sh
- Restore the MQMR IniFiles if they were altered / deleted.
- Delete the MQMR tar file
- Restart the queue manager
- Restart your MQ applications and any monitoring tools

7.1.3 Unix and Linux 64-bit Upgrade

- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQMR API Exit
- Backup all MQMR IniFiles in the MQMR install directory
- Copy the appropriate tar file to the */var/mqm/* directory
- Un-tar the contents of the tar file.
e.g. For AIX 5.3, do the following command:
tar -xvf mqmr_aix53_64.tar
- Run the script as follows:
./setmqmr.sh
- Restore the MQMR IniFiles if they were altered / deleted.
- Delete the MQMR tar file
- Restart the queue manager
- Restart your MQ applications and any monitoring tools

7.1.4 IBM i Upgrade

- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQMR API Exit
- Backup all MQMR IniFiles in the MQMR install directory
- ftp the IBM i files to the IBM i server using the following command:

```
ftp -s:mqmr_iseries.ftp iseries_hostname
```

```
your-IBM i-userid  
your-IBM i-password  
  
binary  
cd QGPL  
put mqmr.savf  
quote SITE NAMEFMT 1  
quit
```

- Log onto the target IBM i server and do the following commands:

```
RSTLIB SAVLIB(MQMR) DEV(*SAVF) SAVF(QGPL/MQMR)  
CLRSVF FILE(QGPL/MQMR)  
CHGOBJOWN OBJ(MQMR) OBJTYPE(*LIB) NEWOWN(QMQM)
```

- Restore the MQMR IniFiles if they were altered / deleted.
- Restart the queue manager
- Restart your MQ applications and any monitoring tools

8 Appendix C – Capitalware Product Display Version

MQMR includes a program to display the product version number. The command to display the product version number is:

cwdspver

8.1 Examples

8.1.1 Windows

To use the cwdspver program on Windows, open a Command prompt and change the directory to **C:\Capitalware\MQMR** and type the following:

```
cwdspver.exe
```

8.1.2 Linux 32-bit

To use the cwdspver program on Linux for MQ 32-bit, open a shell prompt and change directory to **/var/mqm/exits/** and type the following:

```
./cwdspver
```

8.1.3 Unix and Linux 64-bit

To use the cwdspver program on Unix/Linux for MQ 64-bit, open a shell prompt and change directory to **/var/mqm/exits64/** and type the following:

```
./cwdspver
```

8.1.4 IBM i

To use the cwdspver program on IBM i, issue the following command on the Command Prompt:

```
CALL MQMR/CWDSPVER
```

9 Appendix D – Support

The support for MQ Message Replication can be found at the following location:

By email at:

support@capitalware.com

By regular mail at:

Capitalware Inc.
Attn: MQMR Support
Unit 11, 1673 Richmond Street, PMB524
London, Ontario N6G2N3
Canada

10 Appendix E – Summary of Changes

- MQ Message Replication v2.2.0
 - Fixed a bug handling MQPUT1 calls not correctly opening target queues
 - Fixed a bug handling MQPUT1 calls and ForMQPut1DoMQPut1 = Y not setting ObjectName correctly.
 - Fixed an issue in the subroutine that removes trailing blanks
 - Fixed an issue with the logging framework
 - Enhanced the code for dumping the pointers passed into exit.
 - Fixed a bug in mq2sdb not correctly resetting prepared statement after use.
 - Add 2 new columns to the IBM_MQ_MESSAGES table: Source_QMgr_Id & Source_Queue_Id.
 - Add 2 new tables to the database: SOURCE_QMGR_NAMES & SOURCE_QUEUE_NAMES.
 - Added new keyword 'IncludeSourceInfo' for mq2sdb that will cause the source QMgr & queue names to be written to database.
 - Added new keyword 'UseSelect', 'SelectQMgrName' & 'SelectQName' for sdb2mq that will control which messages are loaded into the queue.
 - Fixed issue when an invalid or expired license key is used
 - Fixed an issue with default exit path

- MQ Message Replication v2.1.0
 - Erase ExitUserArea field on exiting.
 - Added code to append trailing slash for ExitPath if it is missing.
 - Fixed issue with UserIDFormatting
 - Tuned the logging code

- MQ Message Replication v2.0.0
 - Added UseSyncPoint keyword to enable synpoint for cloned messages.
 - Added the ability to clear the MQMD Report Options field - new keywords: ClearRO, ClearROCOA, ClearROCOD, ClearROPAN, ClearRONAN, ClearROException & ClearROExpiration
 - Added the ability to skip messages with the MQMD Feedback field set to particular values - new keywords: SkipCOA, SkipCOD, SkipPAN, SkipNAN & SkipExpiration
 - Fixed an issue with the handling of reloading the IniFile
 - Fixed an issue with determining the application name.
 - Fixed an issue in the logging framework where a constant was being modified.
 - Added 2 auxiliary programs:
 - **MQ Queue To SQLite DB** (MQ2SDB) program will offload MQ messages to an SQLite database.
 - **SQLite DB To MQ Queue** (SDB2MQ) program will load SQLite database rows into messages in an MQ queue.

- MQ Message Replication v1.2.0

- Added the ability to exclude applications - new keywords: UseExcludeApplications & ExcludeApplications
 - Added the ability to exclude UserIDs - new keywords: UseExcludeUserIDs & ExcludeUserIDs
 - Added UserIDFormatting flag to force lowercase/uppercase/as_is UserID formatting on all platforms
 - Enhanced logging - the LogFile keyword now supports the following tokens: %QM%, %UID%, %PID% & %TID%
 - Added code to clear hostname field before use
 - Fixed memory allocation when MQOPEN has a non-zero reason code
- MQ Message Replication v1.1.0
- Added new keyword AddSourceInfo which will add the source queue manager name and queue name to the cloned message as message properties
 - Fixed an issue on Windows with freeing environment variable memory (error with FreeEnvironmentStrings Windows API call)
 - Fixed an issue with the Ini Processor not finding next section
 - Fixed an issue with using "size_t" variable type when it should have been "int"
- MQ Message Replication v1.0.0
- Initial release.

11 Appendix F – License Agreement

This is a legal agreement between you (either an individual or an entity) and Capitalware Inc. By opening the sealed software packages (if appropriate) and/or by using the SOFTWARE, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, promptly return the disk package and accompanying items for a full refund.

SOFTWARE LICENSE

1. **GRANT OF LICENSE.** This License Agreement (License) permits you to use one copy of the software product identified above, which may include user documentation provided in on-line or electronic form (SOFTWARE). The SOFTWARE is licensed as a single product, to an individual queue manager, or group of queue managers for an Enterprise License. This Agreement requires that each queue manager of the SOFTWARE be Licensed, either individually, or as part of a group. Each queue manager's use of this SOFTWARE must be covered either individually, or as part of an Enterprise License. The SOFTWARE is in use on a computer when it is loaded into the temporary memory (i.e. RAM) or installed into the permanent memory (e.g. hard disk) of that computer. This software may be installed on a network provided that appropriate restrictions are in place limiting the use to registered queue managers only. Each licensed queue manager will be provided with a perpetual license key and the licensee may continue to use the SOFTWARE, so long as the licensee is current on the Yearly Maintenance Fee. If the licensee stops paying the Yearly Maintenance Fee, then the SOFTWARE must be removed from all systems at the end of the current maintenance period.

2. **COPYRIGHT.** The SOFTWARE is owned by Capitalware Inc. and is protected by United States Of America and Canada copyright laws and international treaty provisions. You may not copy the printed materials accompanying the SOFTWARE (if any), nor print copies of any user documentation provided in on-line or electronic form. You must not redistribute the registration codes provided, either on paper, electronically, or as stored in the files mqmr.ini, mqmr_licenses.ini or any other form.

3. **OTHER RESTRICTIONS.** The registration notification provided, showing your authorization code and this License is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent or lease the SOFTWARE, but you may transfer your rights under this License on a permanent basis, provided you transfer this License, the SOFTWARE and all accompanying printed materials, retain no copies, and the recipient agrees to the terms of this License. You may not reverse engineer, decompile, or disassemble the SOFTWARE, except to the extent the foregoing restriction is expressly prohibited by applicable law.

LIMITED WARRANTY

LIMITED WARRANTY. Capitalware Inc. warrants that the SOFTWARE will perform substantially in accordance with the accompanying printed material (if any) and on-line documentation for a period of 365 days from the date of receipt.

CUSTOMER REMEDIES. Capitalware Inc. entire liability and your exclusive remedy shall be, at Capitalware Inc. option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet this Limited Warranty and that is returned to Capitalware Inc.

with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

NO OTHER WARRANTIES. To the maximum extent permitted by applicable law, Capitalware Inc. disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE and any accompanying written materials.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall Capitalware Inc. be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use the SOFTWARE, even if Capitalware Inc. has been advised of the possibility of such damages.

12 Appendix G – Notices

Trademarks:

AIX, IBM, MQSeries, OS/2 Warp, OS/400, iSeries, MVS, OS/390, WebSphere, IBM MQ and z/OS are trademarks of International Business Machines Corporation.

HP-UX is a trademark of Hewlett-Packard Company.

Intel is a registered trademark of Intel Corporation.

Java, J2SE, J2EE, Sun and Solaris are trademarks of Sun Microsystems Inc.

Linux is a trademark of Linus Torvalds.

Mac OS X is a trademark of Apple Computer Inc.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation.

UNIX is a registered trademark of the Open Group.

WebLogic is a trademark of BEA Systems Inc.