MQ Auditor Installation and Operation Manual

Capitalware Inc.

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

Last Updated: January 2022.

Table of Contents

1 Introduction

1.1 Overview

MQ Auditor (MQA) is a solution that allows a company to audit / track all MQ API calls performed by MQ applications that are connected to a queue manager. The API Exit operates with IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix, IBM i and Linux environments.

MQA audits the following MQ API calls:

- MQCONN, MQCONNX, MQOPEN, MQGET, MQPUT, MQPUT1, MQINQ, MQSET, MQCLOSE, MQDISC, MQBACK, MQBEGIN and MQCMIT.
- XASTART, XAEND, XAOPEN, XACLOSE, XACOMMIT, XACOMPLETE XAFORGET, XAPREPARE, XARECOVER, XAROLLBACK, AX_REG and AX_UNREG.
- MQCALLBACK, MQCB, MQCTL, MQSTAT, MQSUB and MQSUBRQ.

MQA is designed to provide the user with all of the information of an MQ API call in "human readable" format. Human readable implies that it will convert binary fields into their MQ defined names (i.e. ObjType=MQOT_Q). It will convert Command Server / PCF messages into human readable messages. The user can control the fields that are outputted for each of the following MQ structures: MQCNO, MQOD, MQGMO, MQPMO and MQMD. For MQ v7 & higher, the following fields are outputted: Message Properties, MQCBC, MQCBD, MQSTS and MQSD.

MQA handles the following MQ embedded message types: MQCIH, MQDH, MQDLH, MQIIH, MQRFH, MQRFH2, MQRMH, MQTM, MQWIH, MQXQH, MQHSAP, SMQBAD. The user can control the fields that are outputted for each of the MQ embedded messages.

MQA's default behavior is to log all MQ API calls by all applications (users). To override the default, the user can define 4 types of filters: Applications, UserIDs, Queues and Topics.

- Applications means that MQ Auditor will log all MQ API calls whose application name matches the filter value.
- UserIDs means that MQ Auditor will log all MQ API calls whose UserID matches the filter value.
- Queues means that MQ Auditor will log all MQ API calls whose queue name matches the filter value.
- Topics means that MQ Auditor will log all MQ API calls whose topic name matches the filter value.

MQA is designed to output 1 line per API call (1 long line). The output (audit) information is written to plain text CSV (Comma Separate Value) files. There are 2 Audit CSV file types: QMgr and Queue/Topic. Audit information for MQGET, MQPUT, MQPUT1, MQINQ and MQSET related to a queue is written to the Queue Audit file and audit information for all other calls is written to the QMgr Audit file.

The user can choose to have the output (audit) information written to a local or remote queue rather than to a file. From an MQA point of view, this is a very dangerous feature. The reason it is dangerous is that MQA audits / monitors all MQ API calls and for each call it generates audit data. If the audit data is written to a queue then this action can potentially cause an endless loop (MQA will audit itself, over and over again). Therefore, the "audit queue" is not audited / monitored nor is the transmit queue if remote queue is used. If the user is using a remote queue then it is strongly recommended that a separate channel and transmit queue be used so that the normal transmit queue can be audited / monitored. The user will need a program to read the messages from the 'audit queue' and to write the information somewhere (i.e. database).

Audit Queue Off Load (AQOL) is a new companion application for MQ Auditor. The purpose of AQOL is to retrieve audit records from the audit queue and write the audit records to plain text CSV (Comma Separate Value) files. For more information regarding the AQOL program, please read the ***AQOL Installation and Operation*** manual.

MQA can explicitly not write audit information for particular queues, topics, UserIds and applications. It can also explicitly not write information when MQ issues particular reason codes (i.e. 2033).

On AIX, HP-UX, Linux, Solaris and Windows, MQA 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.

1.2 Executive Summary

The MQ Auditor solution contains an API Exit.

The API Exit is available in 3 forms:

- Windows DLL (32-bit & 64-bit)
- Shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The major features of MQ Auditor are as follows:

- Audit / track all MQ API calls issued by an MQ application
- MQA is designed to provide the user with all of the information of an MQ API call in "human readable" format.
- MQA will process all standard MQ structures
- MQA will process all standard embedded message structures
- Provides the audit files in an easy to manage CSV (Comma Separated Value) format
- The user can define 4 types of filters: Applications, UserIDs, Queues & Topics
- The user can select to have the CSV data written to a queue rather than a file
- The user can select to not have audit information written for particular reason codes

1.3 Prerequisites

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

1.3.1 Operating System

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

1.3.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.3.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.3.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.3.1.4 Linux x86

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

1.3.1.5 Linux x86_64 (64-bit)

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

1.3.1.6 Linux on POWER

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

1.3.1.7 Linux on zSeries (64-bit)

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

1.3.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.3.1.9 Sun Solaris

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

1.3.1.10 Windows

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

1.3.2 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 or higher	32-bit & 64-bit
Solaris x86_64 v10 or higher	32-bit & 64-bit
Windows 2008, 2012, 2016, 7, 8, 8.1 & 10	32-bit & 64-bit

1.3.3 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

1.3.4 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 Auditor

This section describes how to install Capitalware's MQ Auditor.

2.1 API Exit

2.1.1 Windows Installation

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

The user may copy or ftp the mqa.dll, 64\mqa.dll, mqa_reg.bat, forfiles.exe, rotatelog.bat and mqa.ini files from one Windows server to another Windows server.

2.1.2 Linux 32-bit Installation

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

- **Linux_x86/mqa_linux.tar**
- **RaspberryPi_ARM/mqa_raspberrypi_arm.tar**

Steps to install the API Exit:

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

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

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

```
chmod +x setmqa.sh
./setmqa.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 mqa_licenses.ini  mqa.ini
```

2.1.3 Unix and Linux 64-bit Installation

To install the 64-bit version of MQA on Unix or Linux, first unzip the **mqa.zip** and then select the appropriate TAR file for the target platform. You will find 10 TAR files in the original ZIP file:

- **AIX/mqa_aix61_64.tar** for AIX v6.1 or higher
- **AIX/mqa_aix71_64.tar** for AIX v7.1 or higher
- **HPUX_IA64 /mqa_hpux64_ia64.tar**
- **Linux_x86_64/mqa_linux_x86_64.tar**
- **Linux_POWER/mqa_linux_power64.tar**
- **Linux_zSeries/mqa_linux_zseries64.tar**
- **Solaris_SPARC/mqa_solaris64.tar** for Solaris SPARC v8 and v9
- **Solaris_SPARC/mqa_solaris10_64.tar** for Solaris SPARC v10 or higher
- **Solaris_x86_64/mqa_solaris_x86_64.tar**

Steps to install the API Exit:

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

```
cd /var/mqm/
tar -xvf  mqa_xxx64.tar
```

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

```
chmod +x setmqa.sh
./setmqa.sh
```

Note: mqa_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 mqa_licenses.ini  mqa.ini
```

2.1.4 IBM i Installation

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

- **mqa.savf** is the IBM i 'Save File' that contains the library with the API exit.
- **mqa_iseries.tar** is the IBM i IFS TAR file that contains a sample initialization file for the API Exit and sample MQSC script to define MQ channels with the API exits.

Steps to install the API Exit:

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

   ```
   CRTSAVF FILE(QGPL/MQA)
   ```

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

   ```
   ftp -s:mqa_iseries.ftp  iseries_hostname
   ```

   ```
   your-IBM i-userid
   your-IBM i-password

   binary
   cd QGPL
   put mqa.savf

   quote SITE NAMEFMT 1

   cd /QIBM/UserData/mqm/
   put mqa_iseries.tar
   quit
   ```

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

   ```
   RSTLIB SAVLIB(MQA) DEV(*SAVF) SAVF(QGPL/MQA)
   CLRSAVF FILE(QGPL/MQA)
   CHGOBJOWN  OBJ(MQA) OBJTYPE(*LIB)  NEWOWN(QMQM)
   qsh
   cd /QIBM/UserData/mqm/
   tar -xvf mqa_iseries.tar
   chown -R QMQM mqa
   chmod -R 777 mqa
   rm mqa_iseries.tar
   ```

2.1.5 MQA-GUI Installation

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

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

2.1.5.1 MQA-GUI Installation on Windows

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

2.1.5.2 MQA-GUI Installation on Unix, Linux or macOS

To install MQA-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 mqagui.zip
```

3 MQ Auditor Files and Directories

This section describes the various files and directories used by MQA.

3.1 Audit Files

There are 2 Audit CSV file types: Queue Manager and Queue.

When the keyword ***OneFilePerConnection*** is set to 'Y' (Yes), all of the audit information is written to the Queue Manager Audit file (Queue Audit files are not used).

When the keyword ***SharedQueueAuditFile*** is set to 'Y' (Yes), only one queue audit file is used for all applications accessing the same queue. For Unix/Linux, make sure your file permissions are set correctly (i.e. world write-able).

3.1.1 Queue Manager Audit File

Audit information for the following MQ API calls is written to the Queue Manager Audit file:

- MQCONN, MQCONNX, MQOPEN, MQCLOSE, MQDISC, MQBACK, MQBEGIN and MQCMIT
- XASTART, XAEND, XAOPEN, XACLOSE, XACOMMIT, XACOMPLETE XAFORGET, XAPREPARE, XARECOVER, XAROLLBACK, AX_REG and AX_UNREG (MQ v6.0.2.7 & higher)
- MQCALLBACK, MQCB, MQCTL, MQSTAT, MQSUB and MQSUBRQ (MQ v7.0 & higher)

File naming convention for the Queue Manager Audit file is as follows:
QMgrName_ApplicationName_P#####_T@@@@@.csv

where

- **QMgrName** is the name of the queue manager
- **ApplicationName** is the name of the application connecting to the queue manager
- **P#####** is the process number
- **T@@@@@** is the thread number

Sample Queue Manager Audit File:

```
2010/09/06 18:32:05.656, MQXF_CONN , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
QMgrName=MQWT2, UserId=rlacroix, CNO_Options=MQCNO_SHARED_BINDING, PgmName=orkspace\MMX\Release\mmx.exe,
PgmType=MQXACT_EXTERNAL, Env=MQXE_OTHER

2010/09/06 18:32:05.656, MQXF_OPEN , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464,
OpenOptions=MQOO_BIND_AS_Q_DEF+MQOO_INPUT_EXCLUSIVE+MQOO_SAVE_ALL_CONTEXT+MQOO_FAIL_IF_QUIESCING,
OD_ObjQMgrName=MQWT2, OD_ObjName=TEST.Q10, OD_ObjType=MQOT_Q, OD_DynamicQNames=AMQ.*,

2010/09/06 18:32:05.656, MQXF_OPEN , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, OpenOptions=MQOO_BIND_AS_Q_DEF+MQOO_OUTPUT+MQOO_PASS_ALL_CONTEXT+MQOO_FAIL_IF_QUIESCING,
OD_ObjQMgrName=MQWT2, OD_ObjName=TEST.Q1, OD_ObjType=MQOT_Q, OD_DynamicQNames=AMQ.*,

2010/09/06 18:32:05.671, MQXF_OPEN , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370112, OpenOptions=MQOO_BIND_AS_Q_DEF+MQOO_OUTPUT+MQOO_PASS_ALL_CONTEXT+MQOO_FAIL_IF_QUIESCING,
OD_ObjQMgrName=MQWT2, OD_ObjName=TEST.Q2, OD_ObjType=MQOT_Q, OD_DynamicQNames=AMQ.*,

2010/09/06 18:32:07.812, MQXF_CMIT , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568
2010/09/06 18:32:09.468, MQXF_CMIT , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568
2010/09/06 18:32:13.093, MQXF_CMIT , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568

2010/09/06 18:32:13.093, MQXF_CLOSE, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568, HObj=-1,
CloseOptions=MQCO_NONE,

2010/09/06 18:32:13.093, MQXF_CLOSE, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568, HObj=-1,
CloseOptions=MQCO_NONE,

2010/09/06 18:32:13.093, MQXF_CLOSE, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568, HObj=-1,
CloseOptions=MQCO_NONE,

2010/09/06 18:32:13.093, MQXF_DISC , A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=-1
```

3.1.2 Queue Audit File

Audit information for MQGET, MQPUT, MQPUT1, MQINQ and MQSET MQ API calls related to a queue is written to the Queue Audit file.

3.1.2.1 When SharedQueueAuditFile is Set to 'N'

File naming convention for the Queue Manager Audit file is as follows:
`QMgrName_ApplicationName_P#####_T@@@@@_QueueName.csv`

where

- `QMgrName` is the name of the queue manager
- `ApplicationName` is the name of the application connecting to the queue manager
- `P#####` is the process number
- `T@@@@@` is the thread number
- `QueueName` is the name of the queue

Sample Queue Audit File:

```
2010/09/06 18:32:07.812, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.81, MD_MsgId=414D51204D5157543220202020202020206243E34B20000802,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=2836, MsgDataAsHex=50657465722E506F746B617940746865686617274666F72642E

2010/09/06 18:32:07.859, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.85, MD_MsgId=414D51204D5157543220202020202020206243E34B20000804,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=12, MsgDataAsHex=536D616C6C2066696C650D0A

2010/09/06 18:32:07.906, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.90, MD_MsgId=414D51204D5157543220202020202020206243E34B20000806,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=5779, MsgDataAsHex=3C3F786D6C2076657273696F6E3D22312E302220656E636F64
```

Sample Queue Audit File:

```
2010/09/06 18:32:07.812, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.81,
MD_MsgId=414D51204D5157543220202020202020206243E34B20000802, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=2836,
MsgDataAsHex=50657465722E506F746B617940746865686617274666F72642E

2010/09/06 18:32:07.859, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.85,
MD_MsgId=414D51204D5157543220202020202020206243E34B20000804, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=12,
MsgDataAsHex=536D616C6C2066696C650D0A

2010/09/06 18:32:07.906, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.90,
MD_MsgId=414D51204D5157543220202020202020206243E34B20000806, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=5779,
MsgDataAsHex=3C3F786D6C2076657273696F6E3D22312E302220656E636F64
```

3.1.2.2 When SharedQueueAuditFile is Set to 'Y'

File naming convention for the Queue Manager Audit file is as follows:
QMgrName_QueueName.csv

where

- **QMgrName** is the name of the queue manager
- **QueueName** is the name of the queue

Sample Queue Audit File:

```
2010/09/06 18:32:07.812, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.81, MD_MsgId=414D51204D5157543220202020202206243E34B20000802,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=2836, MsgDataAsHex=50657465722E506F746B617940746865686172746666F72642E

2010/09/06 18:32:07.859, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.85, MD_MsgId=414D51204D5157543220202020202206243E34B20000804,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=12, MsgDataAsHex=536D616C6C2066696C650D0A

2010/09/06 18:32:07.906, MQXF_GET, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24368464, GMO_Options=MQGMO_WAIT+MQGMO_NO_WAIT+MQGMO_FAIL_IF_QUIESCING+MQGMO_SYNCPOINT+MQGMO_CONVERT,
GMO_WaitInterval=-1, GMO_MatchOptions=MQMO_MATCH_MSG_ID+MQMO_MATCH_CORREL_ID, GMO_ResolvedQName=TEST.Q10,
MD_PutDate=2010/09/06, MD_PutTime=22:32:07.90, MD_MsgId=414D51204D5157543220202020202206243E34B20000806,
MD_MsgType=MQMT_DATAGRAM, MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2,
BufferLength=1024000, DataLength=5779, MsgDataAsHex=3C3F786D6C2076657273696F6E3D22312E302220656E636F64
```

Sample Queue Audit File:

```
2010/09/06 18:32:07.812, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.81,
MD_MsgId=414D51204D5157543220202020202206243E34B20000802, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=2836,
MsgDataAsHex=50657465722E506F746B617940746865686172746666F72642E

2010/09/06 18:32:07.859, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.85,
MD_MsgId=414D51204D5157543220202020202206243E34B20000804, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=12,
MsgDataAsHex=536D616C6C2066696C650D0A

2010/09/06 18:32:07.906, MQXF_PUT, A, PID=4192, TID=1, CC=0, RC=0, UserID=tester, HConn=8887568,
HObj=24370048, PMO_Options=MQPMO_SYNCPOINT+MQPMO_PASS_ALL_CONTEXT+MQPMO_FAIL_IF_QUIESCING,
PMO_ResolvedQMgrName=MQWT2, PMO_ResolvedQName=TEST.Q1, MD_PutDate=2010/09/06, MD_PutTime=22:32:07.90,
MD_MsgId=414D51204D5157543220202020202206243E34B20000806, MD_MsgType=MQMT_DATAGRAM,
MD_Persistence=MQPER_NOT_PERSISTENT, MD_UserId=rlacroix, MD_ReplyToQMgr=MQWT2, BufferLength=5779,
MsgDataAsHex=3C3F786D6C2076657273696F6E3D22312E302220656E636F64
```

3.2 Audit File Directories

MQA writes the Audit information to either a Queue Manager or a Queue audit file while the application is running. When the application disconnects from the queue manager, all of the audit files are move to the "archive" directory. By default, everything in the archive directory is deleted after 7 days (this is a user selected value).

3.2.1 Windows

The default Audit file directory on Windows is:
{Installation_Directory}\audit\

The default Audit file directory on Windows is:
{Installation_Directory}\audit\archive\

E.g. For queue manger MQWT1, the audit and archive directories would be
```
C:\Capitalware\MQA\audit\
C:\Capitalware\MQA\audit\archive\
```

3.2.2 Linux 32-bit

The default Audit file directory on Unix / Linux 32-bit is:
/var/mqm/exits/audit/

The default Audit file directory on Unix / Linux 32-bit is:
/var/mqm/exits/audit/archive/

E.g. For queue manger MQWT1, the audit and archive directories would be
```
/var/mqm/exits/audit/
/var/mqm/exits/audit/archive/
```

3.2.3 Unix and Linux 64-bit

The default Audit file directory on Unix / Linux 64-bit is:
/var/mqm/exits64/audit/

The default Audit file directory on Unix / Linux 64-bit is:
/var/mqm/exits64/audit/archive/

E.g. For queue manger MQWT1, the audit and archive directories would be
```
/var/mqm/exits64/audit/
/var/mqm/exits64/audit/archive
```

3.2.4 IBM i

The default Audit file directory on IBM i is:
/QIBM/UserData/mqm/mqa/audit/

The default Audit file directory on IBM i is:
/QIBM/UserData/mqm/mqa/audit/archive/

E.g. For queue manger MQWT1, the audit and archive directories would be
/QIBM/UserData/mqm/mqa/audit/
/QIBM/UserData/mqm/mqa/audit/archive/

4 Configuring MQ Auditor

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

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

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

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

4.1 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.

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

4.1.1 Windows

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

4.1.1.1 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\MQA;C:\Program Files (x86)\IBM\IBM MQ\exits
   ExitsDefaultPath64=C:\Capitalware\MQA\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 definition:

```
amqmdain reg {QMgrName} -c add  -s ApiExitLocal\MQAuditor -v Name=MQAuditor
amqmdain reg {QMgrName} -c add  -s ApiExitLocal\MQAuditor -v Module=mqa.dll
amqmdain reg {QMgrName} -c add  -s ApiExitLocal\MQAuditor -v Data=C:\Capitalware\MQA\mqa.ini
amqmdain reg {QMgrName} -c add  -s ApiExitLocal\MQAuditor -v Sequence=1
amqmdain reg {QMgrName} -c add  -s ApiExitLocal\MQAuditor -v Function=EntryPoint
```

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

As a convenience, we have included a batch file called ***mqa_reg.bat*** that includes all of the **amqmdain** commands. **mqa_reg.bat** is located in the MQA install directory and it accepts one parameter (the queue manager name).

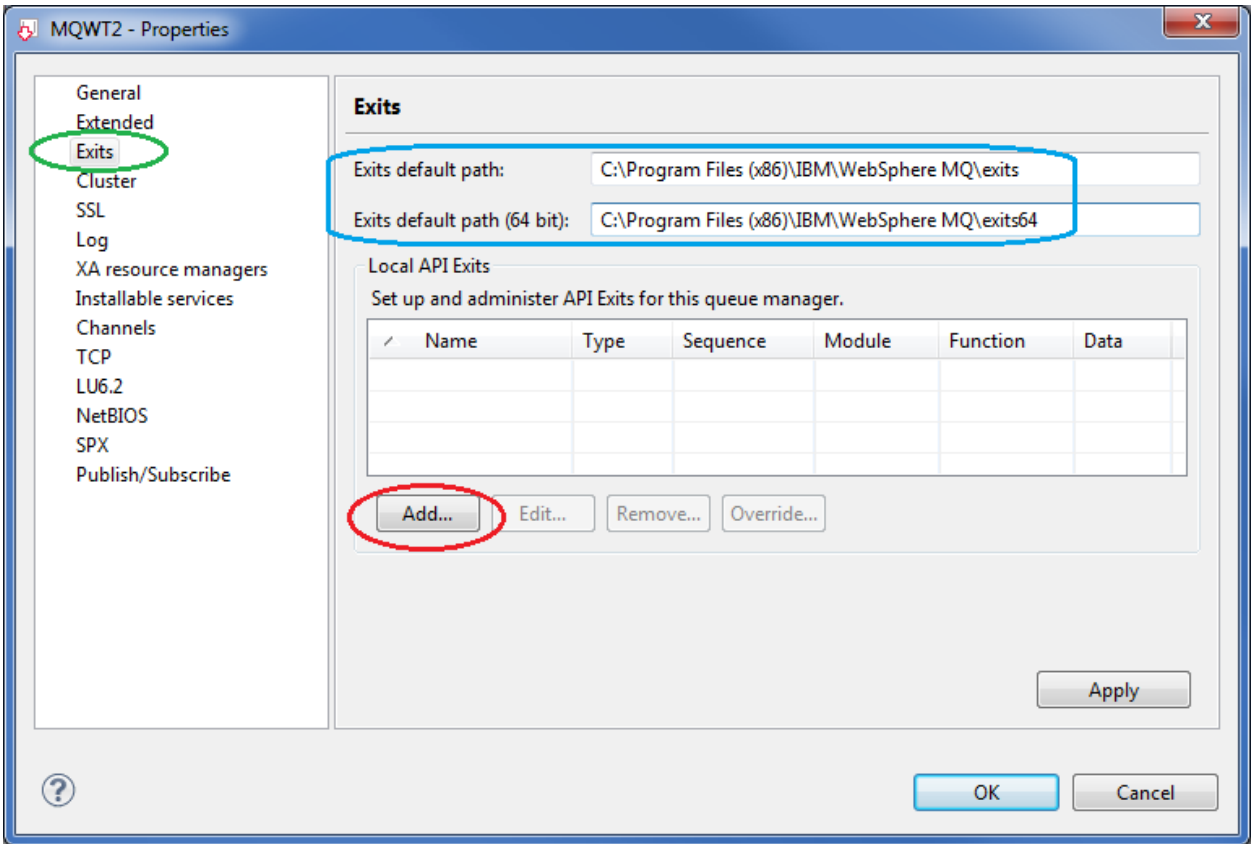
Open a Windows Command Prompt and issue the following commands:

```
cd /D C:\Capitalware\MQA\
mqa_reg.bat {QMgrName}
```

4.1.1.2 Windows MQ Explorer

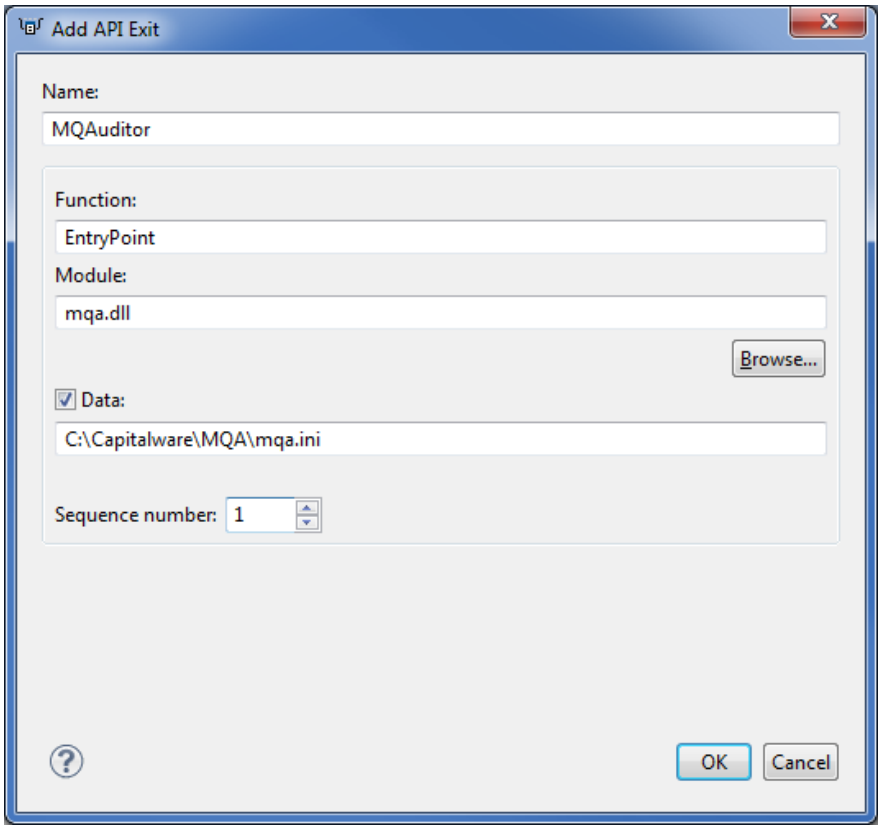
To create a Local API Exit definition using MQ Explorer, do the following:

- 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 MQA install path. i.e. C:\Capitalware\MQA and C:\Capitalware\MQA\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	MQAuditor
Function	EntryPoint
Module	mqa.dll
Data	C:\Capitalware\MQA\mqa.ini
Sequence Number	1



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

4.1.2 Linux 32-bit

On Unix and Linux, edit the ***qm.ini*** for the queue manager that MQA 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=MQAuditor
  Sequence=1
  Function=EntryPoint
  Module=mqa
  Data=/var/mqm/exits/mqa.ini
```

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

4.1.3 Unix and Linux 64-bit

On Unix and Linux, edit the ***qm.ini*** for the queue manager that MQ Auditor 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=MQAuditor
  Sequence=1
  Function=EntryPoint
  Module=mqa
  Data=/var/mqm/exits64/mqa.ini
```

Note: If the user has not installed MQA 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.

4.1.4 IBM i

On IBM i, edit the ***qm.ini*** for the queue manager that the user MQ Auditor 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=MQAuditor
  Sequence=1
  Function=EntryPoint
  Module=MQA/MQA
  Data=/QIBM/UserData/mqm/mqa/mqa.ini
```

4.2 File Paths

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

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

4.2.1 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\MQA\mqa.ini
```

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

4.2.2 Relative Path

Relative pathing for the Data field is supported on Linux, IBM i, Unix and Windows platforms. MQA will use the default 'platform' path and prefix it to the IniFile specified in the Data field in order to locate the IniFile.

For Windows:
C:\Capitalware\MQA

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/mqa/

E.g. Unix

```
Data=mqa.ini
```

Hence, MQA will use the following path as the IniFile path:
e.g. Unix 64-bit
/var/mqm/exits64/

4.2.3 Environment Variables

4.2.3.1 Global Environment Variable

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

e.g. Unix

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

```
Data=mqa.ini
```

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

4.2.3.2 Queue Manager Specific Environment Variable

MQA supports the use of the MQA_HOME environment variable post-fixed with the queue manager name which holds the directory path information. MQA_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 MQA_HOME_MQA1=/really/long/path/MQHA/QMgrName/data2/
```

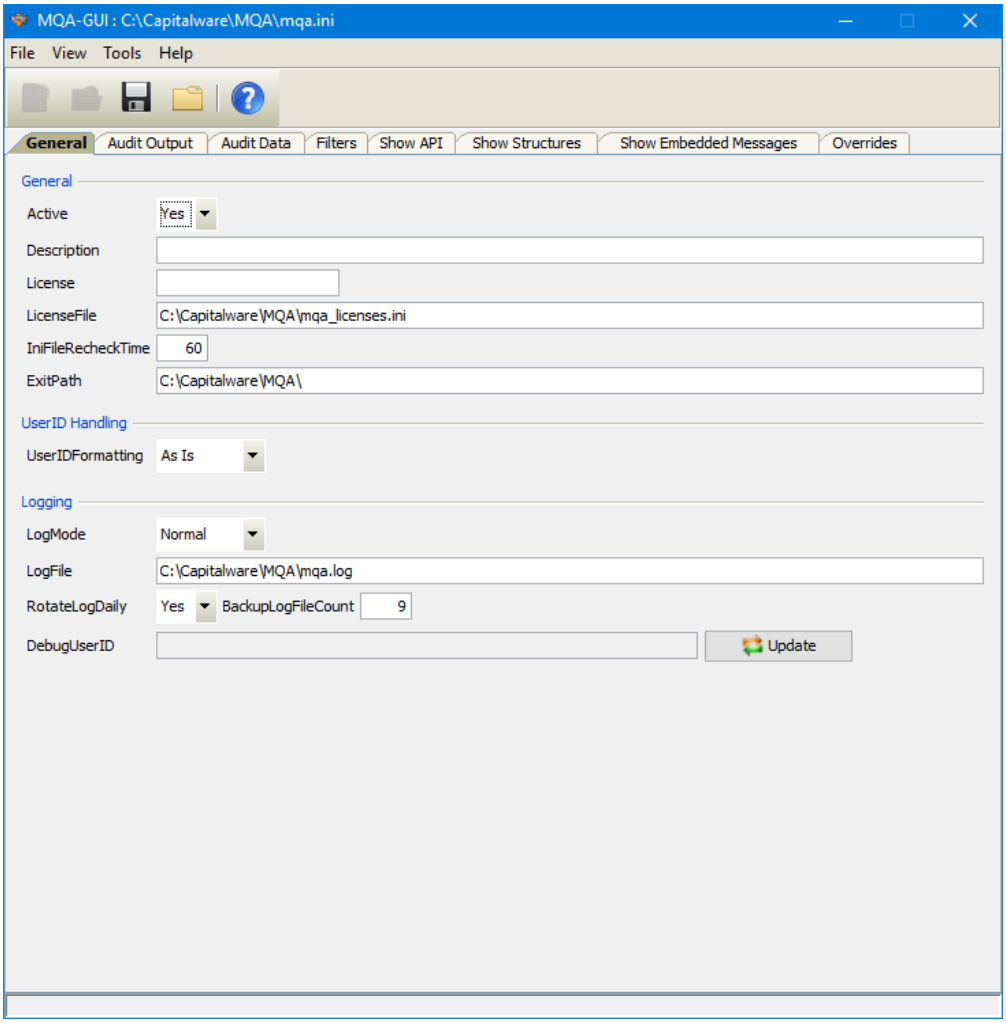
```
Data=mqa.ini
```

Hence, MQA 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.

4.3 MQA-GUI

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



5 IniFile Keywords (Global Values)

This section describes IniFile keywords.

5.1 Audit Files

This section provides a description for the 6 keywords related to Audit files. There are 76keywords related to Audit files:

1. **AuditPath** specifies path to the Audit files Setting this parameter will override the default value for AuditPath. The default is as follows:

   For Windows:
   `AuditPath=C:\Capitalware\MQA\audit\`

   For IBM MQ 32-bit on Unix and Linux:
   `AuditPath=/var/mqm/audit/`

   For IBM MQ 64-bit on Unix and Linux:
   `AuditPath=/var/mqm/audit/`

   For IBM MQ on IBM i:
   `AuditPath=/QIBM/UserData/mqm/mqa/audit/`

2. **OneFilePerConnection** specifies that the MQAdmin wishes to have all audit information outputted only to the Queue Manager Audit file. The default value is N. Setting 'OneFilePerConnection' to 'Y' (Yes) will cause MQA to output all audit information to the Queue Manager Audit file (the Queue Audit file will not be used).

3. **SharedQueueAuditFile** specifies that the MQAdmin wishes to have all audit information for a particular queue outputted to a single audit queue file. The default value is N. Setting 'SharedQueueAuditFile' to 'Y' (Yes) will cause MQA to output all audit queue information to a single Queue Audit file.

   If ArchiveCleanUp is set to 'Y' and SharedQueueAuditFile is set to 'Y', when the Audit file is moved to the archive directory, it is renamed from
   `QMgrName_QueueName.csv` to `QMgrName_QueueName_DYYYY_MM_DD.csv`

4. **UseRollingAuditFile** enables MQA to manage of the Audit (QMgr and Queue) file size. This is controlled by the IniFile's property keyword 'UseRollingAuditFile'. The default value is Y. Setting 'UseRollingAuditFile' to 'Y' (Yes) will activate this feature.

5. **AuditFileMaxSize** specifies how large an Audit file can become before it is moved to the archive directory. The value represents the maximum number of MB (MegaBytes) that the Audit file is to become before it is archived (moved to the Archive directory). The default value is 100. This keyword is only used if UseRollingAuditFile is set to 'Y'.

6. **StartUpCleanUp** specifies the cleanup of the audit directory on queue manager startup. This is controlled by the IniFile's property keyword 'StartUpCleanUp'. The default value is N. Setting 'StartUpCleanUp' to 'Y' (Yes) will activate this feature.

```
AuditPath=C:\Capitalware\MQA\audit\
OneFilePerConnection=N
SharedQueueAuditFile=N
UseRollingAuditFile=Y
AuditFileMaxSize=100
StartUpCleanUp=N
```

5.2 Audit Archives Files

This section provides a description for the keywords related to Audit Archives files. There are 3 keywords related to Audit Archives files:

1. **AuditArchivePath** specifies the path to the Audit Archives files Setting this parameter will override the default value for AuditArchivePath. The default is as follows:

   For Windows:
   AuditArchivePath=C:\Capitalware\MQA\audit\archive

   For IBM MQ 32-bit on Unix and Linux:
   AuditArchivePath=/var/mqm/audit/archive/

   For IBM MQ 64-bit on Unix and Linux:
   AuditArchivePath=/var/mqm/audit/archive/

   For IBM MQ on IBM i:
   AuditArchivePath=/QIBM/UserData/mqm/mqa/audit/archive/

2. **ArchiveCleanUp** specifies cleanup of the archive directory. The default value is Y. Setting 'ArchiveCleanUp' to 'Y' (Yes) will activate this feature.

3. **ArchiveDays** specifies the number of days the archive files should be kept for. The default value is 7. This keyword is only used if ArchiveCleanUp is set to 'Y'.

```
AuditArchivePath=H:\Backup\MQA\archive\
ArchiveCleanUp=Y
ArchiveDays=7
```

5.3 Audit Queue

This section provides a description for the keywords related to Audit Queue. From an MQA point of view, this is a very dangerous feature. The reason it is dangerous is that MQA audits/monitors all MQ API calls and for each call it generates audit data. If the audit data is written to a queue then this action can potentially cause an endless loop (MQA will audit itself, over and over again). Therefore, neither the "audit queue" nor the transmit queue is not audited / monitored if a remote queue is used. If the user is using a remote queue, it is strongly recommended that a separate channel and transmit queue be used so that the normal transmit queue can be audited / monitored.

The user can write their own program to read the messages from the 'audit queue' and to write the information somewhere (i.e. database) or use the ***Audit Queue Off Load*** (AQOL) program that is supplied with MQ Auditor. For more information regarding the AQOL program, please read the ***AQOL Installation and Operation*** manual.

There are 2 keywords related to Audit files:

1. **UseAuditQueue** controls whether AuditQueue is used or not. Set to Y to activate feature.

2. **AuditQueue** is either a local or remote queue containing the location where the audit data is to be outputted to.

```
UseAuditQueue=Y
AuditQueue=CAPITALWARE.AUDIT.QUEUE
```

5.4 Active

Another IniFile keyword is Active. This keyword is used to enable or disable MQA. The default value is Y. Setting 'Active' to 'Y' (Yes) will cause MQA to be active (enabled).

```
Active=Y
```

5.5 ExcludeRC

Another IniFile keyword is ExcludeRC. This keyword contains the MQ reason codes not to generate audit information for. The default value is blank. ExcludeRC supports up to 25 reason codes. Separate each reason code with a semi-colon ";".

```
ExcludeRC=2033;2009
```

5.6 MonitorType

Specifies what type of monitoring the user wishes to have. MonitorType supports 3 values [A / B / C] where A is After, B is Both and C is Conversion. By default, MonitorType has value is A. An 'A' value will generate audit information for each MQ API call AFTER it is completed. A 'C' value will generate audit information the same as 'A' except before and after conversion is outputted for MQGET.

```
MonitorType=A
```

5.7 MonitorInternal

This section describes the necessary entries to enable monitoring of internal MQ applications issuing MQ API calls. The default value is N. Setting 'MonitorInternal' to 'Y' (Yes) will cause MQA to generate audit information for internal MQ applications.

```
MonitorInternal=N
```

5.8 MsgDataAsHex

The MsgDataAsHex keyword controls how MQA will handle the message data when the audit information is generated. The default value is Y. A 'Y' value will cause the message data to be converted to hexadecimal format in the audit record.

```
MsgDataAsHex=Y
```

5.9 MsgDataLength

The MsgDataLength keyword controls how much of the message data will be included in the audit information. The default value is 25. If MsgDataLength is set to '-1' then all of the message data will be outputted.

Note: If MsgDataAsHex is set to 'Y' then the maximum amount of data outputted will be 1024.

```
MsgDataLength=25
```

5.10 Mask for Binary Characters

This section provides a description for the keywords related to 'Mask for Binary' characters. This feature will either replace a character with a different character remove the character from the message data. Note: Carriage Return (X'0D') and Line Feed (x'0A') are not handled in this feature. (See MaskCR and MaskLF features.)

There are 2 keywords related to 'Mask for Binary' characters:

1. **MaskBinaryChars** controls whether UseAsBinaryChar is used or not. Set to Y to activate feature.

2. **UseAsBinaryChar** specifies a character to be used when a binary character is found in the message data. Single quotes are required. There are 3 ways to specify a value for UseAsBinaryChar:
   - To remove the character from the message data, use '' (nothing between the single quotes)
   - To specify a replacement character, use '.' (a single character between the single quotes)
   - To specify a replacement in hexadecimal format, use X'2E' (a hexadecimal number)

```
MaskBinaryChars=Y
UseAsBinaryChar='.'
```

5.11 Mask for Carriage Return Character

This section provides a description for the keywords related to 'Mask for Carriage Return' (CR) character. This feature will either replace a character with a different character remove the character from the message data.

There are 2 keywords related to 'Mask for Carriage Return' character:

1. **MaskCR** controls whether UseAsCR is used or not. Set to Y to activate feature.

2. **UseAsCR** specifies a character to be used when a Carriage Return character is found in the message data. Single quotes are required. There are 3 ways to specify a value for UseAsCR:
   - To remove the character from the message data, use '' (nothing between the single quotes)
   - To specify a replacement character, use '.' (a single character between the single quotes)
   - To specify a replacement in hexadecimal format, use X'2E' (a hexadecimal number)

```
MaskCR=Y
UseAsCR='.'
```

5.12 Mask for Line Feed Character

This section provides a description for the keywords related to 'Mask for Line Feed' (LF) character. This feature will either replace a character with a different character remove the character from the message data.

There are 2 keywords related to 'Mask for Line Feed' character:

1. **MaskLF** controls whether UseAsLF is used or not. Set to Y to activate feature.

2. **UseAsLF** specifies a character to be used when a Line Feed character is found in the message data. Single quotes are required. There are 3 ways to specify a value for UseAsLF:
   - To remove the character from the message data, use '' (nothing between the single quotes)
   - To specify a replacement character, use '.' (a single character between the single quotes)
   - To specify a replacement in hexadecimal format, use X'2E' (a hexadecimal number)

```
MaskLF=Y
UseAsLF='.'
```

5.13 ExitPath

This section describes the necessary entries to explicitly set the ExitPath in order to override the default value for ExitPath. The default values are as follows:

For Windows:

```
ExitPath=C:\Capitalware\MQA\
```

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/mqa/
```

```
ExitPath=E:\software\MQA\
```

5.14 UserIDFormatting

This section describes the necessary entries on how to handle the incoming UserID. 'UserIDFormatting' supports 3 values [A / U / L]. ('As Is, Uppercase and Lowercase). The default value is A.

```
UserIDFormatting=U
```

5.15 Filter by Application Name

This section describes how to monitor particular applications connecting to the queue manager. The default value is '*' (monitor all applications).

MQA will look up the regular expression patterns from the **Applications** keyword in order to determine if the application name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: Applications must NOT exceed 2048 characters.

```
Applications=mq*;hr[0-9][a-f];abc??01
```

5.16 Filter by Queue Name

This section describes how to monitor a particular queue when an application has connected to the queue manager. The default value is '*' (monitor all queues).

MQA will look up the regular expression patterns from the **Queues** keyword in order to determine if the queue name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: Queues must NOT exceed 2048 characters.

```
Queues=TEST*;ABC[0-9][a-f]
```

5.17 Filter by Topic Name

This section describes how to monitor a particular topic when an application has connected to the queue manager. The default value is '*' (monitor all topics).

MQA will look up the regular expression patterns from the **Topics** keyword in order to determine if the topic name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: Topics must NOT exceed 2048 characters.

```
Topics=test/ABC*;test/XYZ[0-9][a-f]
```

5.18 Filter by UserIDs

This section describes how to monitor for particular UserIDs connecting to the queue manager. The default value is '*' (monitor all UserIDs).

MQA will look up the regular expression patterns from the **UserIDs** keyword in order to determine if the UserID matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: UserIDs must NOT exceed 2048 characters.

```
UserIDs=xyz*;user[0-9][a-f]
```

5.19 ExcludeApplications

This section describes how to not monitor (i.e. skip) particular applications when an application has connected to the queue manager.

MQA will look up the regular expression patterns from the **ExcludeApplications** keyword in order to determine if the application matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

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

5.20 ExcludeQueues

This section describes how to not monitor (i.e. skip) particular queues when an application has connected to the queue manager.

MQA will look up the regular expression patterns from the **ExcludeQueues** keyword in order to determine if the queue name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: ExcludeQueues must NOT exceed 2048 characters.

```
UseExcludeQueues=Y
ExcludeQueues=SYSTEM*;ABC[0-9][a-f]
```

5.21 ExcludeTopics

This section describes how to not monitor (i.e. skip) particular topics when an application has connected to the queue manager.

MQA will look up the regular expression patterns from the **ExcludeTopics** keyword in order to determine if the topic name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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: ExcludeTopics must NOT exceed 2048 characters.

```
UseExcludeTopics=Y
ExcludeTopics=test/ABC*;test/XYZ[0-9][a-f]
```

5.22 ExcludeUserIDs

This section describes how to not monitor (i.e. skip) particular queues when an application has connected to the queue manager.

MQA will look up the regular expression patterns from the **ExcludeUserIDs** keyword in order to determine if the queue name matches any of the specified expression patterns. 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 characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

```
UseExcludeUserIDs=Y
ExcludeUserIDs=flint*;rubble;sla*
```

5.23 ShowAPI

The keyword ShowAPI contains the MQ API calls that the user wants MQA to generate in the audit information. The default value is '*'. Separate each API function name by a semi-colon (;).

The API calls for queue managers running MQ v5.3 or higher are as follows:

- MQBACK
- MQBEGIN
- MQCLOSE
- MQCMIT
- MQCONN
- MQCONNX
- MQDISC
- MQGET
- MQINQ
- MQOPEN
- MQPUT
- MQPUT1
- MQSET

The API calls for queue managers running MQ v6.0.2.7 or higher are as follows:

- AX_REG
- AX_UNREG
- XACLOSE
- XACOMMIT
- XACOMPLETE
- XAEND
- XAFORGET
- XAOPEN
- XAPREPARE
- XARECOVER
- XAROLLBACK
- XASTART

The API calls for queue managers running MQ v7.0 or higher are as follows:

- MQCALLBACK
- MQCB
- MQCTL
- MQSTAT
- MQSUB
- MQSUBRQ

```
ShowAPI=*
```

5.24 Show Structures

5.24.1 ShowCNO

The keyword ShowCNO (MQCNO - Connect Options) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'Options'.

The following fields are available:

- ClientConnPtr
- ConnectionId (MQ v6 or higher)
- ConnTag
- Options (default)
- SSLConfigPtr
- SecurityParmsPtr (MQ v6 or higher)
- Version

```
ShowCNO=Options;Version
```

If the user does not want any fields generated, set the ShowCNO keyword equal to 'NONE'.

```
ShowCNO=NONE
```

Note: To generate audit information for all of the fields of MQCNO, set the ShowCNO keyword equal to '*'.

```
ShowCNO=*
```

5.24.2 ShowOD

The keyword ShowOD (MQOD - Object Descriptor) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'Options;ObjectName;ObjectQMgrName;ObjectType;DynamicQName;ResolvedQName;ResolvedQMgrName;AlternateUserId'.

The following fields are available:

- AlternateSecurityId
- AlternateUserId
- DynamicQName
- InvalidDestCount
- KnownDestCount
- ObjectName
- ObjectQMgrName
- ObjectRecPtr
- ObjectString
- ObjectType
- Options
- RecsPresent
- ResObjectString
- ResolvedQMgrName
- ResolvedQName
- ResolvedType
- ResponseRecPtr
- SelectionString
- UnknownDestCount
- Version

```
ShowOD=Options;ObjectName;ObjectQMgrName;ObjectType;DynamicQName;ResolvedQName;ResolvedQMgrName;AlternateUserId
```

If the user does not want any fields generated, set the ShowOD keyword equal to 'NONE'.

```
ShowOD=NONE
```

Note: To generate audit information for all of the fields of MQOD, set the ShowOD keyword equal to '*'.

```
ShowOD=*
```

5.24.3 ShowGMO

The keyword ShowGMO (MQGMO - Get Message Options) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'Options;WaitInterval;ResolvedQName;MatchOptions'.

The following fields are available:

- GroupStatus
- MatchOptions
- MsgHandle
- MsgToken
- Options
- ResolvedQName
- ReturnedLength
- Segmentation
- SegmentStatus
- Version
- WaitInterval

```
ShowGMO=Options;WaitInterval;ResolvedQName;MatchOptions
```

If the user does not want any fields generated, set the ShowGMO keyword equal to 'NONE'.

```
ShowGMO=NONE
```

Note: To generate audit information for all of the fields of MQGMO, set the ShowGMO keyword equal to '*'.

```
ShowGMO=*
```

5.24.4 ShowPMO

The keyword ShowPMO (MQPMO - Put Message Options) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'Options;ResolvedQName;ResolvedQMgrName'.

The following fields are available:

- Action
- Context
- InvalidDestCount
- KnownDestCount
- Options
- OriginalMsgHandle
- NewMsgHandle
- PubLevel
- PutMsgRecFields
- PutMsgRecPtr
- RecsPresent
- ResolvedQMgrName
- ResolvedQName
- ResponseRecPtr
- Timeout
- UnknownDestCount
- Version

```
ShowPMO=Options;ResolvedQName;ResolvedQMgrName
```

If the user does not want any fields generated, set the ShowPMO keyword equal to 'NONE'.

```
ShowPMO=NONE
```

Note: To generate audit information for all of the fields of MQPMO, set the ShowPMO keyword equal to '*'.

```
ShowPMO=*
```

5.24.5 ShowMD

The keyword ShowMD (MQMD - Message Descriptor) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'PutDate;PutTime;MsgType;Format;Persistence;MsgId;CorrelId;ReplyToQ;ReplyToQMgr;UserId'.

The following fields are available:

- AccountingToken
- ApplIdentityData
- ApplOriginData
- BackoutCount
- CCSID
- CorrelId
- Encoding
- Expiry
- Feedback
- Format
- GroupId
- MsgFlags
- MsgId
- MsgSeqNumber
- MsgType
- Offset
- OriginalLength
- Persistence
- Priority
- PutApplName
- PutApplType
- PutDate
- PutTime
- ReplyToQ
- ReplyToQMgr
- Report
- UserId
- Version

```
ShowMD=PutDate;PutTime;MsgType;Format;Persistence;MsgId;Correl
Id;ReplyToQ;ReplyToQMgr;UserId
```

If the user does not want any fields generated, set the ShowMD keyword equal to 'NONE'.

```
ShowMD=NONE
```

Note: To generate audit information for all of the fields of MQMD, set the ShowMD keyword equal to '*'.

```
ShowMD=*
```

5.24.6 ShowMP

The keyword ShowMP will cause MQA to output all message properties to the audit information. Requires MQ v7.0 or higher. The default value is '*'.

```
ShowMP=*
```

If the user does not want any fields generated, set the ShowMP keyword equal to 'NONE'.

```
ShowMP=NONE
```

5.24.7 ShowCBC

The keyword ShowCBC (MQCBC - Call Back Context) contains the fields that the user wants MQA to generate in the audit information. Requires MQ v7.0 or higher. Separate each field by using by a semi-colon (';'). The default value is 'Hobj;CompCode;Reason;BufferLength;DataLength'.

The following fields are available:

- BufferLength
- CallbackArea
- CallType
- CompCode
- ConnectionArea
- DataLength
- Flags
- Hobj
- Reason
- ReconnectDelay
- State
- Version

```
ShowCBC=Hobj;CompCode;Reason;BufferLength;DataLength
```

If the user does not want any fields generated, set the ShowCBC keyword equal to 'NONE'.

```
ShowCBC=NONE
```

Note: To generate audit information for all of the fields of MQCBC, set the ShowCNO keyword equal to '*'.

```
ShowCBC=*
```

5.24.8 ShowCBD

The keyword ShowCBD (MQCBD - Call Back Data Descriptor) contains the fields that the user wants MQA to generate in the audit information. Requires MQ v7.0 or higher. Separate each field by using by a semi-colon (';'). The default value is 'Options;MaxMsgLength;CallbackType;CallbackName'.

The following fields are available:

- CallbackArea
- CallbackFunction
- CallbackName
- CallbackType
- MaxMsgLength
- Options
- Version

```
ShowCBD=Options;MaxMsgLength;CallbackType;CallbackName
```

If the user does not want any fields generated, set the ShowCBD keyword equal to 'NONE'.

```
ShowCBD=NONE
```

Note: To generate audit information for all of the fields of MQCBD, set the ShowCBD keyword equal to '*'.

```
ShowCBD=*
```

5.24.9 ShowSD

The keyword ShowSD (MQSD - Subscription Descriptor) contains the fields that the user wants MQA to generate in the audit information. Requires MQ v7.0 or higher. Separate each field by using by a semi-colon (';'). The default value is 'Options;ObjectName;SubCorrelId;PubApplIdentityData;SubName;SubLevel;SubUserData'.

The following fields are available:

- AlternateSecurityId
- AlternateUserId
- ObjectName
- ObjectString
- Options
- PubAccountingToken
- PubApplIdentityData
- PubPriority
- ResObjectString
- SelectionString
- SubCorrelId
- SubExpiry
- SubLevel
- SubName
- SubUserData
- Version

```
ShowSD=Options;ObjectName;SubCorrelId;PubApplIdentityData;SubN
ame;SubLevel;SubUserData
```

If the user does not want any fields generated, set the ShowSD keyword equal to 'NONE'.

```
ShowSD=NONE
```

Note: To generate audit information for all of the fields of MQSD, set the ShowSD keyword equal to '*'.

```
ShowSD=*
```

5.24.10 ShowSTS

The keyword ShowSTS (MQSTS - Status Information Record) contains the fields that the user wants MQA to generate in the audit information. Requires MQ v7.0 or higher. Separate each field by using by a semi-colon (';'). The default value is 'CompCode;Reason;ObjectName;ObjectQMgrName;OpenOptions'.

The following fields are available:

- CompCode
- ObjectName
- ObjectQMgrName
- ObjectString
- ObjectType
- OpenOptions
- PutFailureCount
- PutSuccessCount
- PutWarningCount
- Reason
- ResolvedObjectName
- ResolvedQMgrName
- SubName
- SubOptions
- Version

```
ShowSTS=CompCode;Reason;ObjectName;ObjectQMgrName;OpenOptions
```

If the user does not want any fields generated, set the ShowSTS keyword equal to 'NONE'.

```
ShowSTS=NONE
```

Note: To generate audit information for all of the fields of MQSTS, set the ShowSTS keyword equal to '*'.

```
ShowSTS=*
```

5.25 Show Embedded Message Structures

5.25.1 ShowCIH

The keyword ShowCIH (MQCIH - CICS Information Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format;OutputDataLength;AbendCode;TransactionId'.

The following fields are available:

- AbendCode
- ADSDescriptor
- AttentionId
- Authenticator
- CancelCode
- CodedCharSetId
- CompCode
- ConversationalTask
- CursorPosition
- Encoding
- ErrorOffset
- Facility
- FacilityKeepTime
- FacilityLike
- Flags
- Format
- Function
- GetWaitInterval
- InputItem
- LinkTypeNextTransactionId
- OutputDataLength
- Reason
- RemoteSysId
- RemoteTransId
- ReplyToFormat
- ReturnCode
- StartCode
- StrucLength
- TaskEndStatus
- TransactionId
- UOWControl
- Version

```
ShowCIH=CodedCharSetId;Encoding;Format;OutputDataLength;AbendCode;TransactionId
```

If the user does not want any fields generated, set the ShowCIH keyword equal to 'NONE'.

```
ShowCIH=NONE
```

Note: To generate audit information for all of the fields of MQCIH, set the ShowCIH keyword equal to '*'.

```
ShowCIH=*
```

5.25.2 ShowDH

The keyword ShowDH (MQDH - Distribution Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format'.

The following fields are available:

- CodedCharSetId
- Encoding
- Flags
- Format
- ObjectRecOffset
- PutMsgRecFields
- PutMsgRecOffset
- RecsPresent
- StrucLength
- Version

```
ShowDH=CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowDH keyword equal to 'NONE'.

```
ShowDH=NONE
```

Note: To generate audit information for all of the fields of MQDH, set the ShowDH keyword equal to '*'.

```
ShowDH=*
```

5.25.3 ShowDLH

The keyword ShowDLH (MQDLH - Dead Letter Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';') The default value is 'Reason;DestQName;DestQMgrName;Format;PutApplType;PutApplName'.

The following fields are available:

- CodedCharSetId
- DestQMgrName
- DestQName
- Encoding
- Format
- PutApplName
- PutApplType
- PutDate
- PutTime
- Reason
- Version

```
ShowDLH=Reason;DestQName;DestQMgrName;Format;PutApplType;PutAp
plName
```

If the user does not want any fields generated, set the ShowDLH keyword equal to 'NONE'.

```
ShowDLH=NONE
```

Note: To generate audit information for all of the fields of MQDLH, set the ShowDLH keyword equal to '*'.

```
ShowDLH=*
```

5.25.4 ShowIIH

The keyword ShowIIH (MQIIH - IMS Information Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format;LTermOverride;TranInstanceId'.

The following fields are available:

- Authenticator
- CodedCharSetId
- CommitMode
- Encoding
- Flags
- Format
- LTermOverride
- MFSMapName
- ReplyToFormat
- SecurityScope
- StrucLength
- TranInstanceId
- TranState
- Version

```
ShowIIH=CodedCharSetId;Encoding;Format;LTermOverride;TranInsta
nceId
```

If the user does not want any fields generated, set the ShowIIH keyword equal to 'NONE'.

```
ShowIIH=NONE
```

Note: To generate audit information for all of the fields of MQIIH, set the ShowIIH keyword equal to '*'.

```
ShowIIH=*
```

5.25.5 ShowRFH

The keyword ShowRFH (MQRFH - Rules and Formatting Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format'.

The following fields are available:

- CodedCharSetId
- Encoding
- Flags
- Format
- NameValue
- StrucLength
- Version

```
ShowRFH=CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowRFH keyword equal to 'NONE'.

```
ShowRFH=NONE
```

Note: To generate audit information for all of the fields of MQRFH, set the ShowRFH keyword equal to '*'.

```
ShowRFH=*
```

5.25.6 ShowRFH2

The keyword ShowRFH2 (MQRFH2 - Rules and Formatting Header 2) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format;Folders'.

The following fields are available:

- CodedCharSetId
- Encoding
- Flags
- Folders
- Format
- NameValueCCSID
- StrucLength
- Version

```
ShowRFH2=CodedCharSetId;Encoding;Format;Folders
```

If the user does not want any fields generated, set the ShowRFH2 keyword equal to 'NONE'.

```
ShowRFH2=NONE
```

Note: To generate audit information for all of the fields of MQRFH2, set the ShowRFH2 keyword equal to '*'.

```
ShowRFH2=*
```

5.25.7 ShowRMH

The keyword ShowRMH (MQRMH - Reference Message Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format'.

The following fields are available:

- CodedCharSetId
- DataLogicalLength
- DataLogicalOffset2
- DataLogicalOffset
- DestEnvLength
- DestEnvOffset
- DestNameLength
- DestNameOffset
- Encoding
- Flags
- Format
- ObjectInstanceId
- ObjectType
- SrcEnvLength
- SrcEnvOffset
- SrcNameLength
- SrcNameOffset
- StrucLength
- Version

```
ShowRMH=CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowRMH keyword equal to 'NONE'.

```
ShowRMH=NONE
```

Note: To generate audit information for all of the fields of MQRMH, set the ShowRMH keyword equal to '*'.

```
ShowRMH=*
```

5.25.8 ShowTM

The keyword ShowTM (MQTM - Trigger Message) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'QName;ProcessName'.

The following fields are available:

- ApplId
- ApplType
- EnvData
- ProcessName
- QName
- TriggerData
- UserData
- Version

```
ShowTM=QName;ProcessName
```

If the user does not want any fields generated, set the ShowTM keyword equal to 'NONE'.

```
ShowTM=NONE
```

Note: To generate audit information for all of the fields of MQTM, set the ShowTM keyword equal to '*'.

```
ShowTM=*
```

5.25.9 ShowWIH

The keyword ShowWIH (MQWIH - Work Information Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format'.

The following fields are available:

- CodedCharSetId
- Encoding
- Flags
- Format
- MsgToken
- ServiceName
- ServiceStep
- StrucLength
- Version

```
ShowWIH=CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowWIH keyword equal to 'NONE'.

```
ShowWIH=NONE
```

Note: To generate audit information for all of the fields of MQWIH, set the ShowWIH keyword equal to '*'.

```
ShowWIH=*
```

5.25.10 ShowXQH

The keyword ShowXQH (MQXQH - Transmission Queue Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'RemoteQName;RemoteQMgrName'.

The following fields are available:

- MsgDesc
- RemoteQMgrName
- RemoteQName
- Version

```
ShowXQH=RemoteQName;RemoteQMgrName
```

If the user does not want any fields generated, set the ShowXQH keyword equal to 'NONE'.

```
ShowXQH=NONE
```

Note: To generate audit information for all of the fields of MQXQH, set the ShowXQH keyword equal to '*'.

```
ShowXQH=*
```

5.25.11 ShowHSAP

The keyword ShowHSAP (MQHSAP - SAP R/3 Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CodedCharSetId;Encoding;Format'.

The following fields are available:

- Client
- CodedCharSetId
- Encoding
- Flags
- Format
- HostName
- Language
- Password
- StrucLength
- SystemNumber
- UserId
- Version

```
ShowHSAP=CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowHSAP keyword equal to 'NONE'.

```
ShowHSAP=NONE
```

Note: To generate audit information for all of the fields of MQHSAP, set the ShowHSAP keyword equal to '*'.

```
ShowHSAP=*
```

5.25.12 ShowSBAD

The keyword ShowSBAD (SMQBAD - SAP R/3 Bad Message Header) contains the fields that the user wants MQA to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'ErrorType;Reason;CodedCharSetId;Encoding;Format'.

The following fields are available:

- CodedCharSetId
- Encoding
- ErrorType
- Format
- PutApplName
- PutApplType
- PutDate
- PutTime
- Reason
- Version

```
ShowSBAD=ErrorType;Reason;CodedCharSetId;Encoding;Format
```

If the user does not want any fields generated, set the ShowSBAD keyword equal to 'NONE'.

```
ShowSBAD=NONE
```

Note: To generate audit information for all of the fields of SMQBAD, set the ShowSBAD keyword equal to '*'.

```
ShowSBAD=*
```

5.26 LicenseFile

This section will describe how to have a file that contains all of the user's MQA 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 = 10U0-AAAA-BBBBBBBB
MQB1 = 10U0-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\MQA\mqa_licenses.ini
```

For IBM MQ 32-bit on Unix and Linux:

```
LicenseFile=/var/mqm/exits/mqa_licenses.ini
```

For IBM MQ 64-bit on Unix and Linux:

```
LicenseFile=/var/mqm/exits64/mqa_licenses.ini
```

For IBM MQ on IBM i:

```
LicenseFile=/QIBM/UserData/mqm/mqa/mqa_licenses.ini
```

5.27 License Key

This section will describe how to license MQ Auditor 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: 10U0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).

```
License=10U0-AAAA-BBBBBBBB
```

5.28 Logging

This section describes the necessary entries to enable MQA 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 is as follows:

   For Windows:
   ```
   LogFile=C:\Capitalware\MQA\mqa.log
   ```

   For IBM MQ 32-bit on Linux:
   ```
   LogFile=/var/mqm/audit/mqa.log
   ```

   For IBM MQ 64-bit on Unix and Linux:
   ```
   LogFile=/var/mqm/audit/mqa.log
   ```

   For IBM MQ on IBM i:
   ```
   LogFile=/QIBM/UserData/mqm/mqa/mqa.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 MQA. The default value is 9. This keyword is only used if RotateLogDaily is set to 'Y'.

6 IniFile Keywords (Overriding Values)

This section describes IniFile keywords that override the global IniFile values. To explicitly override the global IniFile values by Application Name, UserID or Queue Name, the user can create an Inifile stanza to hold those specific values.

6.1 Application Name

This section describes the necessary entries to explicitly override the global IniFile values by a specific Application Name. The section stanza must begin in column 1 with a left square bracket '[' followed by a "A:" and end with a right square bracket ']'. Between the colon and the right square bracket, the user can explicitly specify an application name or a regular expression pattern.

The MQA will look up the application name against the regular expression pattern in order to determine if the application name matches any of the specified expression.

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

6.1.1 Overridden Fields

The following IniFile values can be overridden:

• AuditFileMaxSize • ExcludeQueues • ExcludeRC • ExcludeTopics • MaskBinaryChars • MaskCR • MaskLFMonitorType • MsgDataAsHex • MsgDataLength • OneFilePerConnection • Queues • SharedQueueAuditFile • ShowAPI • ShowCBC (MQ v7 or higher) • ShowCBD (MQ v7 or higher) • ShowCIH • ShowCNO • ShowDH • ShowDLH • ShowGMO • ShowHSAP	• ShowIIH • ShowMD • ShowOD • ShowPMO • ShowRFH • ShowRFH2 • ShowRMH • ShowSBAD • ShowSD (MQ v7 or higher) • ShowSTS (MQ v7 or higher) • ShowTM • ShowWIH • ShowXQH • Topics • UseAsBinaryChar • UseAsCR • UseAsLF • UseExcludeQueues • UseExcludeTopics • UseRollingAuditFile

6.1.2 Example

Note: Under Windows, MQA strips the file extension of “exe” from the file name before looking up the stanza name in the IniFile.

If the connecting application is called “testpgm” then the user needs to uppercase the name and prefix “A:”, so that the stanza looks like: [A:TESTPGM]

```
[A:TESTPGM]
MsgDataAsHex=N
ShowAPI=MQGET;MQPUT;MQPUT1


[A:TEST*]
MsgDataAsHex=Y
MsgDataLength=100
ShowMD=*
```

6.2 Queue Name

This section describes the necessary entries to explicitly override the global IniFile values by a specific Queue Name. 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 can explicitly specify a queue name or a regular expression pattern.

The MQA will look up the queue name against the regular expression pattern in order to determine if the queue name matches any of the specified expression.

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

6.2.1 Overridden Fields

The following IniFile values can be overridden:

• MaskBinaryChars • MaskCR • MaskLFMonitorType • MsgDataAsHex • MsgDataLength • ShowCIH • ShowDH • ShowDLH • ShowGMO • ShowHSAP • ShowIIH • ShowMD	• ShowOD • ShowPMO • ShowRFH • ShowRFH2 • ShowRMH • ShowSBAD • ShowTM • ShowWIH • ShowXQH • UseAsBinaryChar • UseAsCR • UseAsLF

6.2.2 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]

```
[Q:TEST.Q01]
MsgDataAsHex=N
MsgDataLength=1000
ShowMD=*


[Q:TEST.HR*]
MsgDataAsHex=Y
MsgDataLength=100
ShowMD=*
```

6.3 Topic Name

This section describes the necessary entries to explicitly override the global IniFile values by a specific Topic Name. The section stanza must begin in column 1 with a left square bracket '[' followed by a "T:" and end with a right square bracket ']'. Between the colon and the right square bracket, the user can explicitly specify a topic name or a regular expression pattern.

The MQA will look up the topic name against the regular expression pattern in order to determine if the topic name matches any of the specified expression.

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

6.3.1 Overridden Fields

The following IniFile values can be overridden:

• MaskBinaryChars • MaskCR • MaskLFMonitorType • MsgDataAsHex • MsgDataLength • ShowCIH • ShowDH • ShowDLH • ShowGMO • ShowHSAP • ShowIIH • ShowMD	• ShowOD • ShowPMO • ShowRFH • ShowRFH2 • ShowRMH • ShowSBAD • ShowTM • ShowWIH • ShowXQH • UseAsBinaryChar • UseAsCR • UseAsLF

6.3.2 Example

If the topic name is called “TEST.TOPIC” then the user needs to prefix “T:”, so that the stanza looks like: [T:TEST.TOPIC]

```
[T:TEST.TOPIC ]
MsgDataAsHex=N
MsgDataLength=1000
ShowMD=*


[T:TEST.HR*]
MsgDataAsHex=Y
MsgDataLength=100
ShowMD=*
```

6.4 UserID

This section describes the necessary entries to explicitly override the global IniFile values by a specific UserID. The section stanza must begin in column 1 with a left square bracket '[' followed by a "U:" and end with a right square bracket ']'. Between the colon and the right square bracket, the user can explicitly specify a UserID or a regular expression pattern.

The MQA will look up the UserID against the regular expression pattern in order to determine if the UserID matches any of the specified expression.

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more characters)
- '?' matches any single character (either numeric or alphabetic)
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z or 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 a range of characters. 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.

6.4.1 Overridden Fields

The following IniFile values can be overridden:

• AuditFileMaxSize • ExcludeQueues • ExcludeRC • ExcludeTopics • MaskBinaryChars • MaskCR • MaskLFMonitorType • MsgDataAsHex • MsgDataLength • OneFilePerConnection • Queues • SharedQueueAuditFile • ShowAPI • ShowCBC (MQ v7 or higher) • ShowCBD (MQ v7 or higher) • ShowCIH • ShowCNO • ShowDH • ShowDLH • ShowGMO • ShowHSAP	• ShowIIH • ShowMD • ShowOD • ShowPMO • ShowRFH • ShowRFH2 • ShowRMH • ShowSBAD • ShowSD (MQ v7 or higher) • ShowSTS (MQ v7 or higher) • ShowTM • ShowWIH • ShowXQH • Topics • UseAsBinaryChar • UseAsCR • UseAsLF • UseExcludeQueues • UseExcludeTopics • UseRollingAuditFile

6.4.2 Example

If the connecting application has a UserID of "abcxyz" then the user needs to uppercase the name and prefix "U:", so that the stanza looks like: [U:ABCXYZ]

```
[U:ABCXYZ]
MsgDataAsHex=N
ShowAPI=MQGET;MQPUT;MQPUT1
ShowMD=*

[U:MQ*]
MsgDataAsHex=N
ShowAPI=*
ShowMD=PutDate;PutTime;Format;Persistence;MsgId;CorrelId;UserId
```

7 Appendix A – Summary of IniFile

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

```
[default]
Active=Y
LogMode=N
LogFile=C:\Capitalware\MQA\mqa.log
UserIDs=*
Queues=*
Applications=*
```

Note: Keywords are case sensitive.

The IniFile supports the following keywords and their values:

Keyword	Description of Server-side keywords
Active	**Active** specifies if MQA is enabled or disabled. **Active** supports 2 values [Y / N]. The default value is Y. e.g. **Active**=N
Applications	**Applications** specifies how to selectively monitor applications. The default value is '*' meaning all applications are monitored. e.g. **Applications**=*
ArchiveCleanUp	**ArchiveCleanUp** turns on cleanup of the archive directory. **ArchiveCleanUp** supports 2 values [Y / N]. The default value is Y. e.g. **ArchiveCleanUp**=N
ArchiveDays	**ArchiveDays** specifies the number of days that the archive files will be kept. The default value is 7. e.g. **ArchiveDays**=7 Note: Only used if **ArchiveCleanUp** is set to 'Y'.

Keyword	Description of Server-side keywords
AuditArchivePath	**AuditArchivePath** specifies the path to the Audit Archives files. Setting this parameter will override the default value for **AuditArchivePath**. The default values are as follows: For Windows: **AuditArchivePath**=C:\Capitalware\MQA\audit\archive\ For IBM MQ 32-bit on Unix and Linux: **AuditArchivePath**=/var/mqm/audit/archive/ For IBM MQ 64-bit on Unix and Linux: **AuditArchivePath**=/var/mqm/audit/archive/ For IBM MQ on IBM i: **AuditArchivePath**=/QIBM/UserData/mqm/mqa/audit/archive/ e.g. **AuditArchivePath**=C:\Capitalware\MQA\audit\archive\
AuditFileMaxSize	**AuditFileMaxSize** specifies how large an Audit file can become before it is moved to the archive directory. The default value is 100. The value is in MB (MegaBytes). e.g. **AuditFileMaxSize**=100
AuditPath	**AuditPath** specifies the path to the Audit files. Setting this parameter will override the default value for **AuditPath**. The default values are as follows: For Windows: **AuditPathC**:\Capitalware\MQA\audit\ For IBM MQ 32-bit on Unix and Linux: **AuditPath**=/var/mqm/audit/ For IBM MQ 64-bit on Unix and Linux: **AuditPath**=/var/mqm/audit/ For IBM MQ on IBM i: **AuditPath**=/QIBM/UserData/mqm/mqa/audit/ e.g. **AuditPath**=C:\Capitalware\MQA\audit\

Keyword	Description of Server-side keywords
AuditQueue	**AuditQueue** specifies the name of the queue that MQA will write the audit information to. e.g. **AuditQueue**=CAPITALWARE.AUDIT.QUEUE Note: Only used if **UseAuditQueue** is set to 'Y'.
BackupLogFileCount	**BackupLogFileCount** specifies the number of backup logfiles that the API exit will be keeping. The default value is 9. e.g. **BackupLogFileCount**=9
DebugUserID	**DebugUserID** specifies a list of UserID which should have debug logging turned on for. e.g. **DebugUserID**=fred;barney
ExcludeApplications	**ExcludeApplications** specifies which particular applications should not be tracked. e.g. **ExcludeApplications**=test*;gadget Note: Only used if **UseExcludeApplications** is set to 'Y'.
ExcludeQueues	**ExcludeQueues** specifies which particular queues should not be tracked. e.g. **ExcludeQueues**=TEST.Q01 Note: Only used if **UseExcludeQueues** is set to 'Y'.
ExcludeTopics	**ExcludeTopics** specifies which particular topics should not be tracked. e.g. **ExcludeTopics**=test/ABC;test/XYZ Note: Only used if **UseExcludeTopics** is set to 'Y'.
ExcludeUserIDs	**ExcludeUserIDs** specifies which particular UserIDs should not be tracked. e.g. **ExcludeUserIDs**=fred;barney Note: Only used if **UseExcludeUserIDs** is set to 'Y'.

Keyword	Description of Server-side keywords
ExcludeRC	**ExcludeRC** specifies the MQ reason codes to not generate audit information. e.g. **ExcludeRC**=2033;2009
ExitPath	**ExitPath** specifies how to explicitly set the ExitPath in order to override the default values for ExitPath. The default values are as follows: For Windows: **ExitPath**=C:\Capitalware\MQA\ 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/mqa/ e.g. **ExitPath**=C:\Capitalware\MQA\
IniFileRecheckTime	**IniFileRecheckTime** specifies the amount, in seconds, before the IniFile is checked whether it has changed or not. The default value is 60. e.g. **IniFileRecheckTime**=60
License	**License** specifies the queue manager's license key. Your license will look something like: 10S0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work). e.g. **License**=10U0-AAAA-BBBBBBB

Keyword	Description of Server-side keywords
LicenseFile	**LicenseFile** specifies the location of License file that contains all of the customer's license keys. The default values for LicenseFile are as follows: For Windows: **LicenseFile**=C:\Capitalware\MQA\mqa_licenses.ini For IBM MQ 32-bit on Unix and Linux: **LicenseFile**=/var/mqm/exits/mqa_licenses.ini For IBM MQ 64-bit on Unix and Linux: **LicenseFile**=/var/mqm/exits64/mqa_licenses.ini For IBM MQ on IBM i: **LicenseFile**=/QIBM/UserData/mqm/mqa/mqa_licenses.ini e.g. **LicenseFile**=/var/mqm/exits64/mqa_licenses.ini
LogFile	**LogFile** specifies the location of the log file. The default values are as follows: For Windows: **LogFile**=C:\Capitalware\MQA\mqa.log For IBM MQ 32-bit on Unix and Linux: **LogFile**=/var/mqm/audit/mqa.log For IBM MQ 64-bit on Unix and Linux: **LogFile**=/var/mqm/audit/mqa.log For IBM MQ on IBM i: **LogFile**=/QIBM/UserData/mqm/mqa/mqa.log
LogMode	**LogMode** specifies what type of logging the user wishes to use. 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. e.g. **LogMode**=V
MaskBinaryChars	**MaskBinaryChars** specifies whether binary characters in the message data should be replaced or removed. **MaskBinaryChars** supports 2 values [Y / N]. The default value is N. e.g. **MaskBinaryChars**=N

Keyword	Description of Server-side keywords
MaskCR	**MaskCR** specifies whether the carriage return (CR) character in the message data should be replaced or removed. **MaskCR** supports 2 values [Y / N]. The default value is N. e.g. **MaskCR**=N
MaskLF	**MaskLF** specifies whether line feed (LF) character in the message data should be replaced or removed. **MaskLF** supports 2 values [Y / N]. The default value is N. e.g. **MaskLF**=N
MonitorInternal	**MonitorInternal** specifies whether monitoring of internal MQ applications issuing MQ API calls is enabled or not. **MonitorInternal** supports 2 values [Y / N]. The default value is N. e.g. **MonitorInternal**=N
MonitorType	**MonitorType** specifies the type of monitoring the user wishes to use. **MonitorType** supports 3 values [A / B / C] where A is After, B is Both and 'C' is Conversion. The default value is A. e.g. **MonitorType**=B
MsgDataAsHex	**MsgDataAsHex** controls how MQA will handle the message data as hexadecimal when the audit information is generated. **MsgDataAsHex** supports 2 values [Y / N]. The default value is Y. e.g. **MsgDataAsHex**=N
MsgDataLength	**MsgDataLength** controls how much of the message data will be included in the audit information. The default value is 25. e.g. **MsgDataLength**=50
OneFilePerConnection	**OneFilePerConnection** specifies that the MQAdmin wishes to have all audit information outputted only to the Queue Manager Audit file. **OneFilePerConnection** supports 2 values [Y / N]. The default value is N. e.g. **OneFilePerConnection**=Y

Keyword	Description of Server-side keywords
Queues	**Queues** specifies how to selectively monitor queues. The default value is '*' meaning all queues are monitored. e.g. **Queues**=*
RotateLogDaily	**RotateLogDaily** specifies whether or not daily log file rotation should take place. **RotateLogDaily** supports 2 values [Y / N]. The default value is Y. e.g. **RotateLogDaily**=Y
SharedQueueAuditFile	**SharedQueueAuditFile** specifies whether or not queue audit file sharing should take place. **SharedQueueAuditFile** supports 2 values [Y / N]. The default value is N. If ArchiveCleanUp is set to 'Y' and SharedQueueAuditFile is set to 'Y', when the Audit file is moved to the archive directory, it is renamed from `QMgrName_QueueName.csv` to `QMgrName_QueueName_DYYYY_MM_DD.csv` e.g. **SharedQueueAuditFile**=Y
ShowAPI	**ShowAPI** contains the MQ API calls that the user wants MQA to generate in the audit information. The default value is '*' meaning all MQ API calls will be monitored. e.g. **ShowAPI**=*
ShowCBC	**ShowCBC** (Call Back Context) contains the fields of the CallBack API call that the user wants MQA to generate in the audit information. MQ v7.0 or higher is required. The default value is 'Hobj;CompCode;Reason;BufferLength;DataLength'. e.g. **ShowCBC**= Hobj;CompCode;Reason
ShowCBD	**ShowCBD** (Call Back Data Descriptor) contains the fields of the CallBack API call that the user wants MQA to generate in the audit information. MQ v7.0 or higher is required. The default value is 'Options;MaxMsgLength;CallbackType;CallbackName'. e.g. **ShowCBC**=Options;CallbackType;CallbackName

Keyword	Description of Server-side keywords
ShowCNO	**ShowCNO** (Connect Options) contains the fields of the MQCONN / MQCONNX API calls that the user wants MQA to generate in the audit information. The default value is 'Options'. e.g. **ShowCNO**=Options
ShowCIH	**ShowCIH** (CICS Information Header) contains the fields of the MQCIH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format;OutputDataLength;AbendCode;TransactionId'. e.g. **ShowCIH**=CodedCharSetId;Encoding;Format;OutputDataLength;AbendCode;TransactionId
ShowDH	**ShowDH** (Distribution Header) contains the fields of the MQDH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format'. e.g. **ShowDH**=CodedCharSetId;Encoding;Format
ShowDLH	**ShowDLH** (Dead Letter Header) contains the fields of the MQDLH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'Reason;DestQName;DestQMgrName;Format;PutApplType;PutApplName'. e.g. **ShowDH**=Reason;DestQName;DestQMgrName;Format;PutApplType;PutApplName
ShowGMO	**ShowGMO** (Get Message Options) contains the fields of the MQGET API call that the user wants MQA to generate in the audit information. The default value is 'Options;WaitInterval;ResolvedQName;MatchOptions'. e.g. **ShowGMO**=Options;WaitInterval;MatchOptions
ShowHSAP	**ShowHSAP** (SAP R/3 Header) contains the fields of the MQHSAP embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format'. e.g. **ShowHSAP**=CodedCharSetId;Encoding;Format

Keyword	Description of Server-side keywords
ShowIIH	**ShowIIH** (IMS Information Header) contains the fields of the MQIIH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format;LTermOverride;TranInstance Id'. e.g. **ShowIIH**=CodedCharSetId;Encoding;Format;LTermOverride;TranInstanceId
ShowMD	**ShowMD** (Message Descriptor) contains the fields of the MQGET, MPUT and MQPUT1 API call that the user wants MQA to generate in the audit information. The default value is 'PutDate;PutTime;MsgType;Format;Persistence;MsgId;CorrelId; ReplyToQ;ReplyToQMgr;UserId'. e.g. **ShowMD**=PutDate;PutTime;Format;MsgId;CorrelId;UserId
ShowMP	**ShowMP** will cause MQA to output message properties to the audit information. MQ v7.0 or higher is required. The default value is '*'. e.g. **ShowMP**= *
ShowOD	**ShowOD** (Object Descriptor) contains the fields of the MQOPEN and MQPUT1 API calls that the user wants MQA to generate in the audit information. The default value is 'Options;ObjectName;ObjectQMgrName;ObjectType;DynamicQ Name;ResolvedQName;ResolvedQMgrName;AlternateUserId'. e.g. **ShowCBD**=Options;ObjectName;ObjectQMgrName;DynamicQ Name;ResolvedQName;ResolvedQMgrName;
ShowPMO	**ShowPMO** (Put Message Options) contains the fields of the MQPUT and MQPUT1 API calls that the user wants MQA to generate in the audit information. The default value is 'Options;ResolvedQName;ResolvedQMgrName'. e.g. **ShowPMO**= Options;ResolvedQName;ResolvedQMgrName

Keyword	Description of Server-side keywords
ShowRFH	**ShowRFH** (Rules and Formatting Header) contains the fields of the MQRFH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format'. e.g. **ShowRFH**=CodedCharSetId;Encoding;Format
ShowRFH2	**ShowRFH2** (Rules and Formatting Header 2) contains the fields of the MQRFH2 embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format;Folders'. e.g. **ShowRFH2**=CodedCharSetId;Encoding;Format;Folders
ShowRMH	**ShowRMH** (Reference Message Header) contains the fields of the MQRMH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format'. e.g. **ShowRMH**=CodedCharSetId;Encoding;Format
ShowSBAD	**ShowSBAD** (SAP R/3 Bad Message Header) contains the fields of the SMQBAD embedded message structure that the user wants MQA to generate in the audit information. The default value is 'ErrorType;Reason;CodedCharSetId;Encoding;Format'. e.g. **ShowSBAD**=ErrorType;Reason;CodedCharSetId;Encoding;Format
ShowSD	**ShowSD** (Subscription Descriptor) contains the fields of the SUB API call that the user wants MQA to generate in the audit information. MQ v7.0 or higher is required. The default value is 'Options;ObjectName;SubCorrelId;PubApplIdentityData;SubName;SubLevel;SubUserData'. e.g. **ShowSD**=Options;ObjectName;SubCorrelId;PubApplIdentityData;SubName;SubLevel;SubUserData

Keyword	Description of Server-side keywords
ShowSTS	**ShowSTS** (Status Information Record) contains the fields of the STAT API call that the user wants MQA to generate in the audit information. MQ v7.0 or higher is required. The default value is 'CompCode;Reason;ObjectName;ObjectQMgrName;OpenOptions' e.g. **ShowSTS**=CompCode;Reason;ObjectName;ObjectQMgrName;OpenOptions
ShowTM	**ShowTM** (Trigger Message) contains the fields of the MQTM embedded message structure that the user wants MQA to generate in the audit information. The default value is 'QName;ProcessName'. e.g. **ShowTM**=QName;ProcessName
ShowWIH	**ShowWIH** (Work Information Header) contains the fields of the MQWIH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'CodedCharSetId;Encoding;Format'. e.g. **ShowWIH**=CodedCharSetId;Encoding;Format
ShowXQH	**ShowXQH** (Transmission Queue Header) contains the fields of the MQXQH embedded message structure that the user wants MQA to generate in the audit information. The default value is 'RemoteQName;RemoteQMgrName'. e.g. **ShowXQH**=RemoteQName;RemoteQMgrName
StartUpCleanUp	**StartUpCleanUp** specifies whether or not cleanup should be started. **StartUpCleanUp** supports 2 values [Y / N]. The default value is N. e.g. **StartUpCleanUp**=Y
Topics	**Topics** specifies how to selectively monitor topics. The default value is '*' meaning all topics are monitored. e.g. **Topics**=*

Keyword	Description of Server-side keywords
UseAsBinaryChar	**UseAsBinaryChar** specifies the replacement character. e.g. **UseAsBinaryChar**=’’ // nothing between single quotes **UseAsBinaryChar**=’.’ // single char. between single quotes **UseAsBinaryChar**=x'2E' // HEX representation of a character Note: Only used if **MaskBinaryChars** is set to 'Y'.
UseAsCR	**UseAsCR** specifies the replacement character. e.g. **UseAsCR**=’’ // nothing between single quotes **UseAsCR**=’.’ // single char. between single quotes **UseAsCR**=x'2E' // HEX representation of a character Note: Only used if **MaskCR** is set to 'Y'.
UseAsLF	**UseAsLF** specifies the replacement character. e.g. **UseAsLF**=’’ // nothing between single quotes **UseAsLF**=’.’ // single char. between single quotes **UseAsLF**=x'2E' // HEX representation of a character Note: Only used if **MaskLF** is set to 'Y'.
UseAuditQueue	**UseAuditQueue** specifies whether or not audit queue should be used rather than audit files. **UseAuditQueue** supports 2 values [Y / N]. The default value is N. e.g. **UseAuditQueue**=Y
UseExcludeApplications	**UseExcludeApplications** specifies whether or not ExcludeApplications should be activated. **UseExcludeApplications** supports 2 values [Y / N]. The default value is N. e.g. **UseExcludeApplications**=Y
UseExcludeQueues	**UseExcludeQueues** specifies whether or not ExcludeQueues should be activated. **UseExcludeQueues** supports 2 values [Y / N]. The default value is N. e.g. **UseExcludeQueues**=Y

Keyword	Description of Server-side keywords
UseExcludeTopics	**UseExcludeTopics** specifies whether or not ExcludeTopics should be activated. **UseExcludeTopics** supports 2 values [Y / N]. The default value is N. e.g. **UseExcludeTopics**=Y
UseExcludeUserIDs	**UseExcludeUserIDs** specifies whether or not ExcludeQueues should be activated. **UseExcludeUserIDs** supports 2 values [Y / N]. The default value is N. e.g. **UseExcludeUserIDs**=Y
UserIDFormatting	**UserIDFormatting** specifies how MQ Auditor will handle the incoming UserID. UserIDFormatting supports 3 values [A / U / L]. ('As Is, Uppercase and Lowercase). The default value is A. UserIDFormatting=U
UserIDs	**UserIDs** specifies which particular UserIDs are to be monitored. The default value is '*'. meaning all UserIDs are monitored. e.g. **UserIDs**=fred;wilma;barney;betty
UseRollingAuditFile	**UseRollingAuditFile** specifies whether or not MQA should manage t the Audit (QMgr and Queue) file size. **UseRollingAuditFile** supports 2 values [Y / N]. The default value is N. e.g. **UseRollingAuditFile**=Y

8 Appendix B – MQA Upgrade Procedures

To upgrade an existing installation of MQA from an older version to a newer version, do please do the following in the appropriate section below.

8.1.1 Windows Upgrade

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

8.1.2 Linux 32-bit Upgrade

- Login under the mqm account
- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQA API Exit
- Backup all MQA IniFiles in the MQA install directory
- Copy the appropriate tar file to the ***/var/mqm/*** directory
- Un-tar the contents of the tar file.
  i.e. For Linux, do the following command:
  ```
  tar -xvf  mqa_linux_x86.tar
  ```
- Run the script as follows:
  ```
  ./setmqa.sh
  ```
- Restore the MQA IniFiles if they were altered / deleted.
- Delete the MQA tar file
- Start the queue manager
- Restart your MQ applications and any monitoring tools

8.1.3 Unix and Linux 64-bit Upgrade

- Stop all MQ applications connecting to the queue manager including monitoring tools
- Stop the queue manager using the MQA API Exit
- Backup all MQA IniFiles in the MQA install directory
- Copy the appropriate tar file to the ***/var/mqm/*** directory
- Un-tar the contents of the tar file.
  i.e. For AIX 5.3, do the following command:
  ```
  tar -xvf  mqa_aix53_64.tar
  ```
- Run the script as follows:
  ```
  ./setmqa.sh
  ```
- Restore the MQA IniFiles if they were altered / deleted.
- Delete the MQA tar file
- Start the queue manager
- Restart your MQ applications and any monitoring tools

8.1.4 IBM i Upgrade

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

```
ftp -s:mqa_iseries.ftp  iseries_hostname
```

```
your-IBM i-userid
your-IBM i-password

binary
cd QGPL
put mqa.savf
quote SITE NAMEFMT 1
cd /QIBM/UserData/mqm/
put mqa_iseries.tar
quit
```

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

```
RSTLIB SAVLIB(MQA) DEV(*SAVF) SAVF(QGPL/MQA)
CLRSAVF FILE(QGPL/MQA)
CHGOBJOWN  OBJ(MQA) OBJTYPE(*LIB)  NEWOWN(QMQM)
qsh
cd /QIBM/UserData/mqm/
tar -xvf mqa_iseries.tar
chown -R QMQM mqa
chmod -R 777 mqa
rm mqa_iseries.tar
```

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

9 Appendix C – Capitalware Product Display Version

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

`cwdspver`

9.1 Examples

9.1.1 Windows

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

```
cwdspver.exe
```

9.1.2 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
```

9.1.3 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
```

9.1.4 IBM i

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

```
CALL MQA/CWDSPVER
```

10 Appendix D – Support

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

By email at:
support@capitalware.com

By regular mail at:

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

11 Appendix E – Summary of Changes

- MQ Auditor v3.2.0
  - Added 6 new keywords to handle replacing Carriage Return, Line Feed or binary data for the message data of the audit record
  - Added missing override fields for overriding Applications and UserId sections: Topics, UseExcludeQueues, ExcludeQueues, UseExcludeTopics & ExcludeTopics
  - 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 an issue with default exit path
  - Fixed an issue with retrieving override values.
  - Fixed an issue with moving audit files to archive directory on startup i.e. StartUpCleanUp=Y
  - Fixed issue when an invalid or expired license key is used

- MQ Auditor v3.1.0
  - Added code to make sure AuditPath and AuditArchivePath have a trailing slash.
  - Updated code so that when ArchiveCleanup is Yes, only '*.csv' files are deleted.
  - Added code to allocate ResObjectString on MQOPEN if application did not provide it.
  - Erase ExitUserArea field on on exiting.
  - Added code to append trailing slash for ExitPath if it is missing.
  - Added code to append trailing slash for AuditPath if it is missing.
  - Added code to append trailing slash for AuditArchivePath if it is missing.
  - Fixed issue with UserIDFormatting
  - Tuned the logging code

- MQ Auditor v3.0.0
  - Added Topics keyword to control what topics will be audited.
  - Added UseExcludeTopics and ExcludeTopics keywords to explicitly exclude topics from being audited.
  - Enhanced the processing for Subscriptions.
  - Fixed an issue with large usr folder for an MQRFH2 (JMS) messsage
  - Fixed an issue with formatting of some MQCHARV fields for MQSD structure
  - Fixed an issue with MQPut1 using the correct audit file.
  - Fixed an issue with PMO 'NewMsgHandle' string being in the wrong location.
  - Fixed an issue with determining the application name.
  - Fixed an issue with PCF messages being converted to human-readable large than 10KB.
  - Fixed an issue with PCF messages not being handled for MQCALLBACK.
  - Fixed an issue in the logging framework where a constant was being modified.
  - Removed MQAPILevel keyword as it is no longer needed.

- MQ Auditor v2.4.0
  - Added the ability to exclude applications - new keywords: UseExcludeApplications & ExcludeApplications
  - 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
  - Fixed issue for MQPUT1 freeing memory twice

- MQ Auditor v2.3.0
  - Added MQOD v4 fields, MQPMO v3 fields & MQGMO v4 fields
  - 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 a bug for message data not being outputted for MQPUT/1 when MD is null
  - Fixed an issue with MQGET Before (call type) outputting incorrect amount of the data buffer
  - Fixed an issue with using "size_t" variable type when it should have been "int"

- MQ Auditor v2.2.0
  - Changed the Audit CSV file template to use dashes '-' rather than underscores '_'
  - Added code to move Audit file to Archive directory when SharedQueueAuditFile is set to 'Y'. If ArchiveCleanUp is set to 'Y' and SharedQueueAuditFile is set to 'Y', when the Audit file is moved to the archive directory, it is renamed from `QMgrName_QueueName.csv` to `QMgrName_QueueName_DYYYY_MM_DD.csv`
  - Fixed an issue with testing for passed length value to the exit (check for negative value)

- MQ Auditor v2.1.0
  - Added keyword IniFileRecheckTime to only check the IniFile modification time after 'x' seconds
  - Added keyword DebugUserID to selectively enable debugging by UserID
  - Added a check for queue manager stopping/quiescing for MQGET call
  - Improved the IniFile processing speed.
  - Changed data buffer length handling when UseAuditQueue is set to 'Y'
  - Changed logfile rotation so that it is only done under the mqm UserID
  - Changed Audit record timestamp to use microseconds (6 digits) rather than milliseconds (3 digits).
  - Fixed an issue with Enterprise License key not being loaded from a License file.
  - Tested with MQ v8.0
  - Tested with Windows 8/8.1

- MQ Auditor v2.0.0
  - Added a new Component: Audit Queue Off Load
  - Added ChannelName to audit record for MQConn/MQConnX API calls - only available for MQ v7.1 or higher
  - Added Conversion for MonitorType - this causes audit records to be written for before and after conversion for MQGET API call.
  - Added support non-default install for MQ v7.1 & higher multi-install feature on Linux, Unix and Windows
  - Increased the accepted IniFile parameter length from 1024 to 2048 characters
  - Changed the handling of rolling audit file feature
  - Changed outputting MQMD ApplIdentityData & ApplOriginData as character rather than HEX.
  - Changed how audit files are copied to the archive directory
  - Fixed an issue with all audit data going to the queue manager audit file
  - Fixed an issue with MQClose and files being left open
  - Fixed a non-threading issue on AIX
  - Fixed an issue AuditQueue truncating the MsgData value
  - Fixed a bug in MQCallback with MQGMO v4 structure
  - Fixed a bug in MQCallback where MQOPEN was not called first
  - Fixed a bug in MQCallback where Active=N and null pointer
  - Fixed GMO field name order.
  - Fixed a crash when the IniFile is invalid.
  - Updated the logger
  - Tested with MQ v7.5

- MQ Auditor v1.2.0
  - Fixed an issue with vsnprintf/vfprintf on Linux 64-bit servers.
  - Enhanced the PCF message handling process
  - Improved the speed of the writing the Audit record to a file or queue

- MQ Auditor v1.1.0
  - Added support for UseExcludeUserIDs & ExcludeUserIDs keywords
  - Added support for the following embedded message types: MQCIH, MQDH, MQDLH, MQIIH, MQRFH, MQRFH2, MQRMH, MQTM, MQWIH, MQXQH, MQHSAP, SMQBAD
  - Added code to support the "NONE" value for the Show*** keywords
  - Added program cwdspver to display the product version number
  - Fixed a bug related to UserID used for MQCONN/MQCONNX may be different for MQOPEN
  - Fixed the bug related to the flag that determines if a queue is being monitored
  - Fixed a bug related to Perl and Linux
  - Fixed a bug related to the handling of the AccountingToken field of the MQMD
  - Fixed a bug related to memcpy for RFH2 structure overwriting memory
  - Fixed a bug related to an unset pointer

- Fixed a number of bugs related to correctly formatting the structures for output to Audit files

- MQ Auditor v1.0.0
  - Initial release.

12 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 mqa.ini, mqa_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.

13 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.