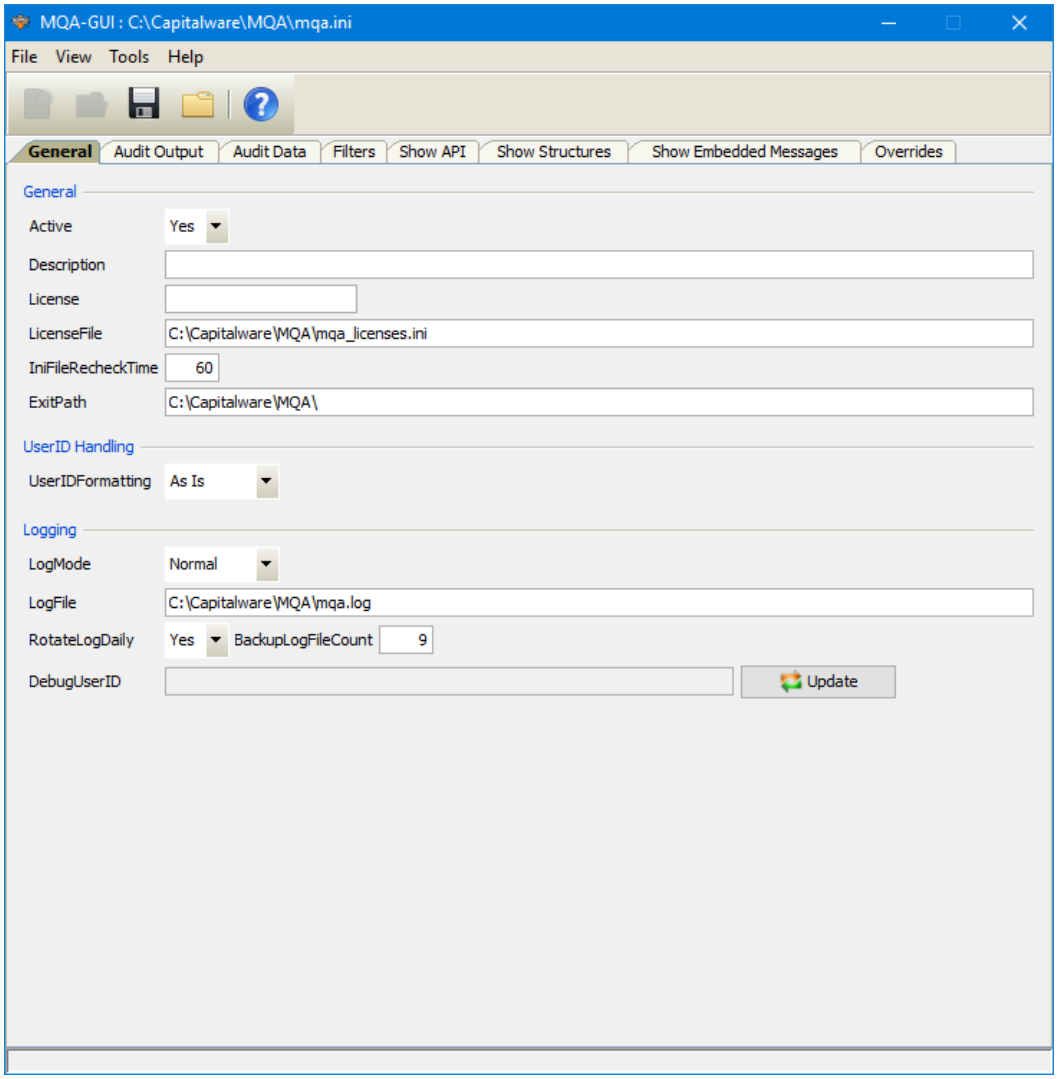
MQA-GUI User Guide



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

Last Updated: July 2020.
© Copyright Capitalware Inc. 2010, 2020.

Table of Contents

1 INTRODUCTION....1

1.1 Overview....1

2 INSTALLATION....3

2.1 Prerequisites....3
2.2 Software....3
2.3 Hardware / OS....3
2.4 Setting Up and Running the Application....3
2.4.1 For Windows Users:....3
2.4.2 For Linux and Max OS X (requires JRE/JDK):....4

3 CREATING OR UPDATING INIFILES....5

3.1 New IniFile....5
3.2 Open an IniFile....6
3.3 Close an IniFile....7
3.4 Save IniFile....8
3.5 Save As IniFile....9

4 TABS....10

4.1 General Tab....10
4.1.1 General....11
4.1.2 UserID Handling....11
4.1.3 Logging....12
4.2 Audit Output Tab....13
4.2.1 Audit Data to a Queue....14
4.2.2 Audit Data to Files....14
4.2.3 Audit Archive Files....15
4.3 Audit Data Tab....16
4.3.1 Monitor....17
4.3.2 Message Data....17
4.3.3 Mask for Binary Characters....17
4.3.4 Mask for Carriage Return Character....17
4.3.5 Mask for Line Feed Character....18
4.4 Filters Tab....19
4.4.1 Include Filters....20
4.4.2 Exclude Filters....20
4.5 Show API Tab....21
4.6 Show Structures Tab....22
4.7 Show Embedded Messages Tab....24
4.8 Overrides Tab....26
4.8.1 Managing Overrides....27

5 OTHER TOOLS....29

5.1 Hex Converter....29
5.2 Hex Editor....30

5.3 Tail File....31
5.4 Text Editor....32
5.5 Visual Difference....34
5.6 XML Editor....36
6 MISCELLANEOUS....37
6.1 Language....37
6.2 Reset View....38
6.2.1 Windows Command Line....38
6.2.2 Linux and macOS Command Line....38
7 APPENDIX A – FREQUENTLY ASKED QUESTIONS (FAQ)....39
8 APPENDIX B – MQA-GUI UPGRADE PROCEDURES....40
8.1 Windows Upgrade....40
8.2 Linux or macOS Upgrade....40
9 APPENDIX C – SUPPORT....41
10 APPENDIX D – SUMMARY OF CHANGES....42
11 APPENDIX E – LICENSE AGREEMENT....44
12 APPENDIX F – NOTICES....46

1 Introduction

1.1 Overview

MQA-GUI application allows the user to create or update MQ Auditor (MQA) IniFiles. MQA-GUI can run on any platform that supports Java v1.6 (or higher): Linux, macOS (Mac OS X), and Windows Vista/7/8/8.1/10.

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 IBM i, Linux, Unix and Windows, MQA can be configured and used with a non-default installation of MQ in a multi-install MQ environment.

2 Installation

This chapter will describe the installation and configuration (optional) of MQA-GUI. For Windows users, the installation of MQA-GUI is very simple and straightforward. For other Operating Systems, it is necessary to unzip the software package.

2.1 Prerequisites

This section lists the required hardware and software components needed to run the MQA-GUI application.

2.2 Software

The following third party software components must be installed in order to properly run MQA-GUI:

- Java Runtime Environment (JRE) 1.6 (or higher) or Java 2 SDK, Standard Edition (J2SE) v1.6 (or higher)

2.3 Hardware / OS

MQA-GUI is capable of running on the following hardware platforms that support Java v1.3 (or higher):

- Linux
- Mac OS X
- Windows Vista, 7, 8, 8.1 & 10

2.4 Setting Up and Running the Application

The following section provides instructions on installing and running the MQA-GUI application.

2.4.1 For Windows Users:

- Run the install program called: **mqagui-setup-withjre.exe**
- The installer should follow the standard Windows install procedures and provide default values for the users.
- When the install program has installed MQA-GUI, there will be a newly created folder under **Start** -> All **Programs** called **MQA-GUI**.
- To run **MQA-GUI**, select the **MQA-GUI** icon from within the **MQA-GUI** folder.

2.4.2 For Linux and Max OS X (requires JRE/JDK):

- MQA-GUI is provided as a zip file: **mqagui.zip**
- To unzip the file, open a shell prompt and execute the following commands:

  unzip mqagui.zip

- When the unzip command has finished, there should be a directory tree as follows:

  Capitalware/MQA-GUI

- Change directory by typing the following:

  cd Capitalware/MQA-GUI

- Before running MQA-GUI for the first time, you need to add the execute attribute to the mqagui.sh shell scripts. Type the following:

  chmod +x mqagui.sh

- To run **MQA-GUI** from the **Capitalware/MQA-GUI** directory, then type the following:

  ./mqagui.sh

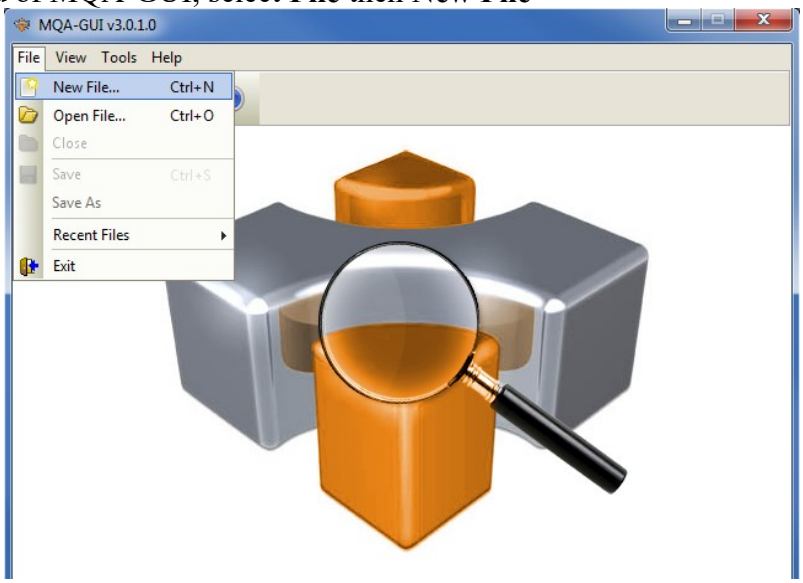
3 Creating or Updating IniFiles

This chapter will describe the how to create or update IniFiles.

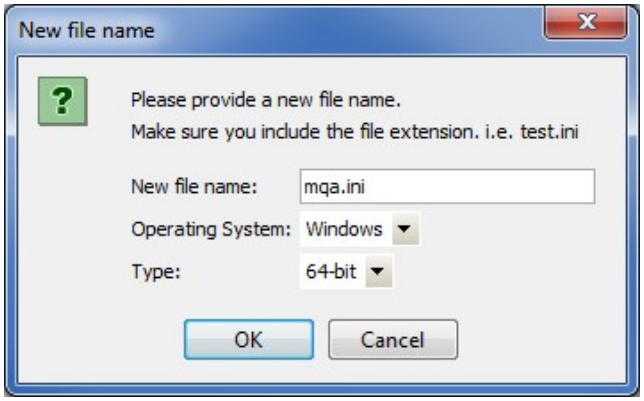
3.1 New IniFile

This section will describe how to create a new MQA IniFile. The IniFile can be for a local or remote MQA implementation.

From the main menu of MQA-GUI, select **File** then **New File**



The following is the New IniFile panel:



Fill in the following information:

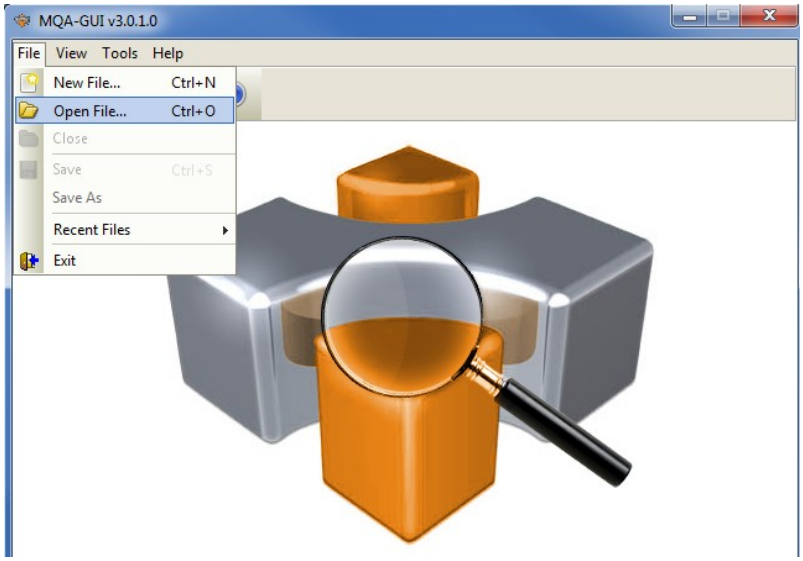
- Input a file name including the file extension of '.ini'
- Select the Operating System
- Choose between a 32-bit or 64-bit environment

Finally, click the **Ok** button to create and open the new MQA IniFile.

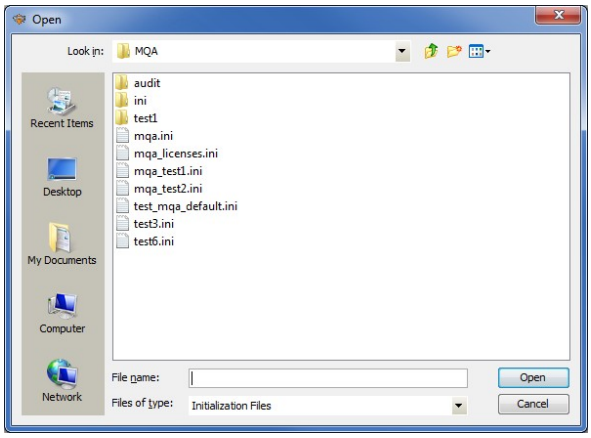
3.2 Open an IniFile

This section will describe how to open a MQA IniFile to be viewed or edited. The IniFile can be for a local or remote MQA implementation.

From the main menu of MQA-GUI, select **File** then **Open File**



The following is the **Open** dialog box:

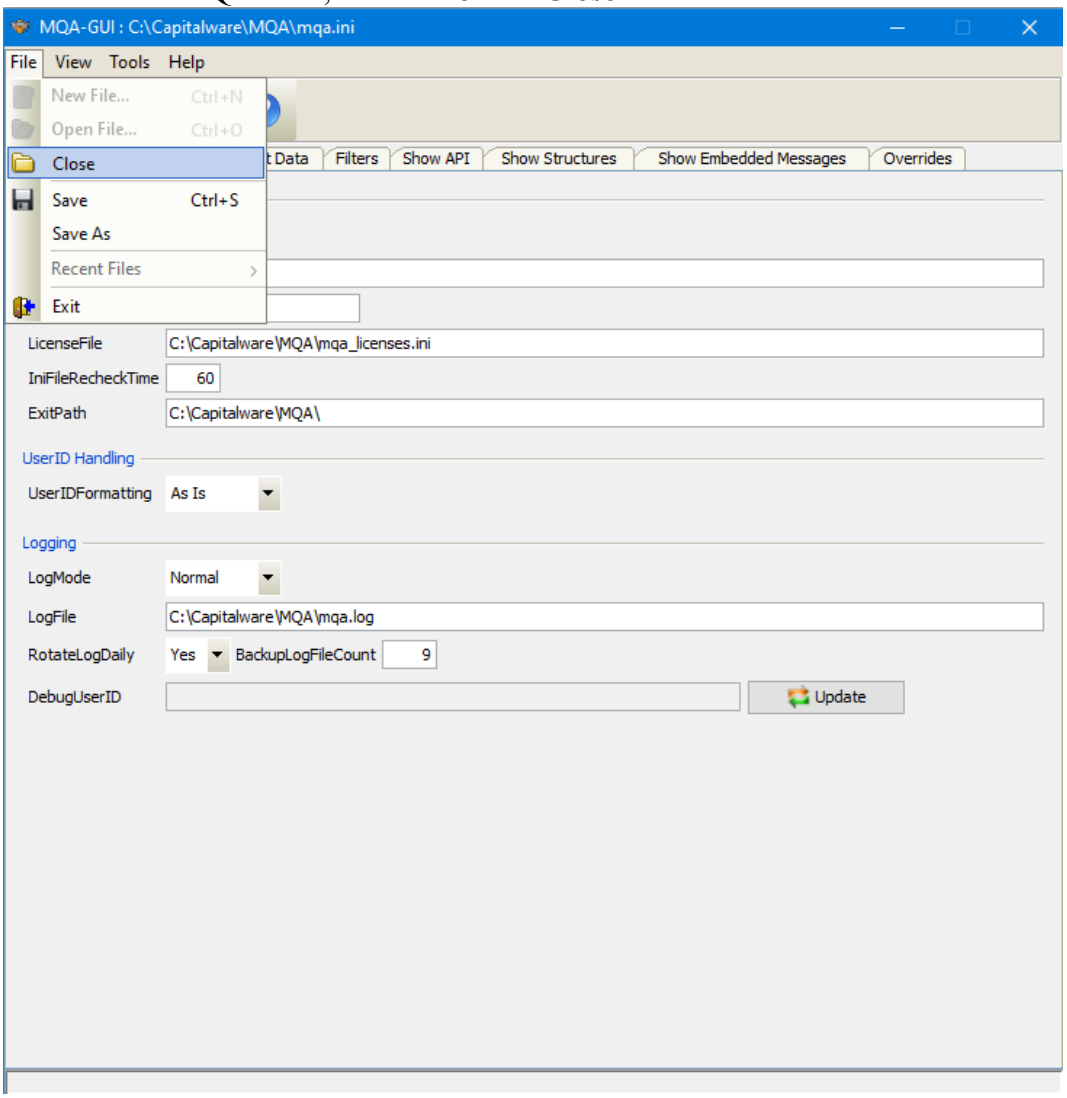


Select the IniFile that you wish to view or edit and then click the **Open** button.

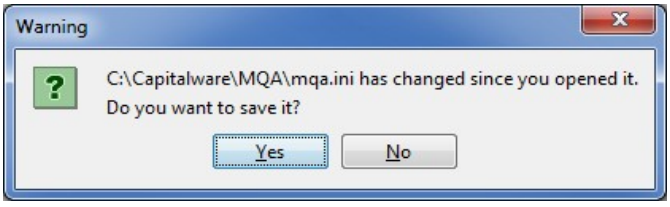
3.3 Close an IniFile

This section will describe how to close a MQA IniFile.

From the main menu of MQA-GUI, select **File** then **Close**



If the user has altered the IniFile then the user will be presented with the following dialog box:

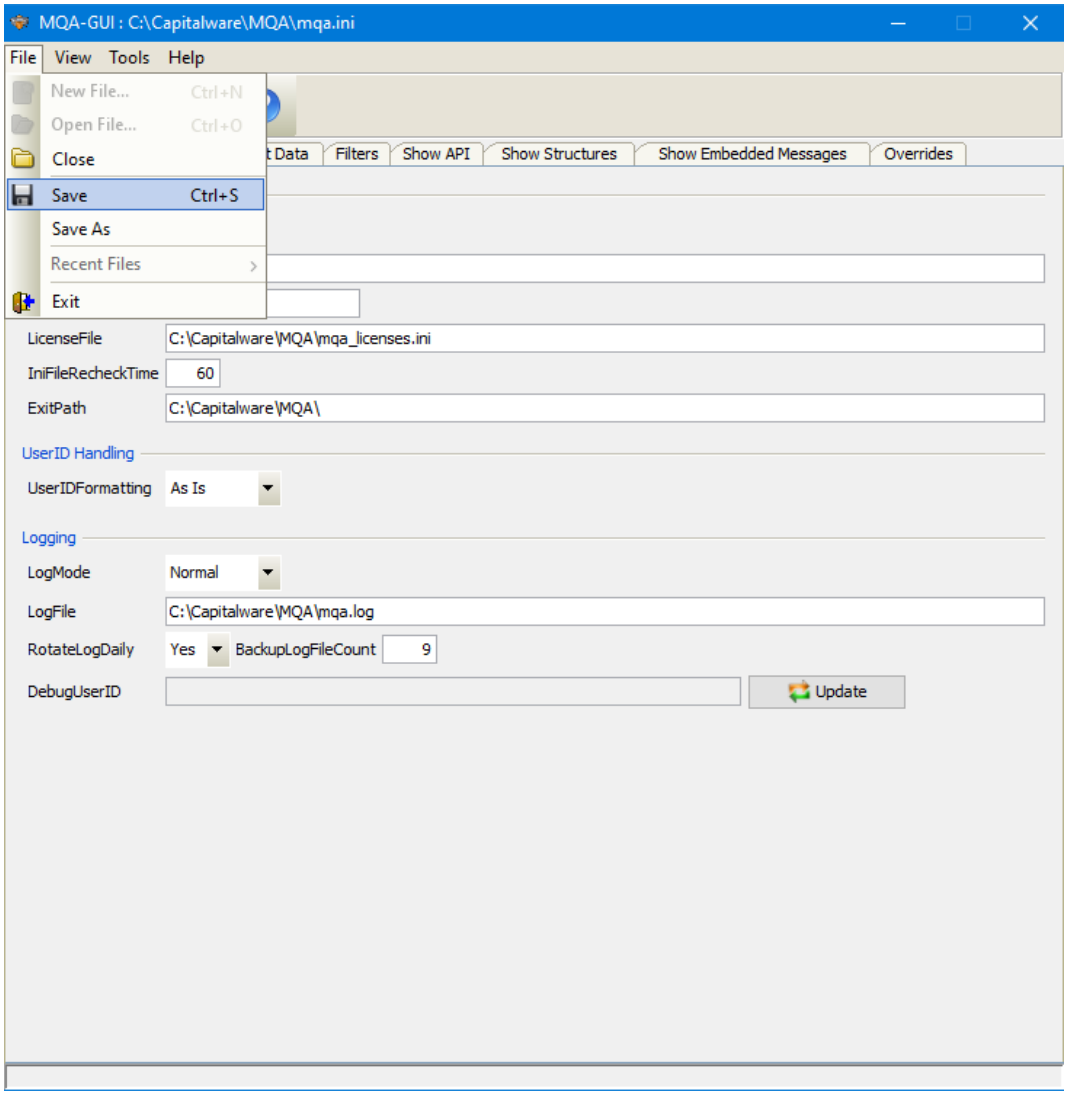


Select **Yes** to save the changes; otherwise select **No**. The MQA-GUI program will make a backup of the original file in the same directory and call it: ***{inifilename}.backup***

3.4 Save IniFile

This section will describe how to save an open MQA IniFile.

From the main menu of MQA-GUI, select **File** then **Save**

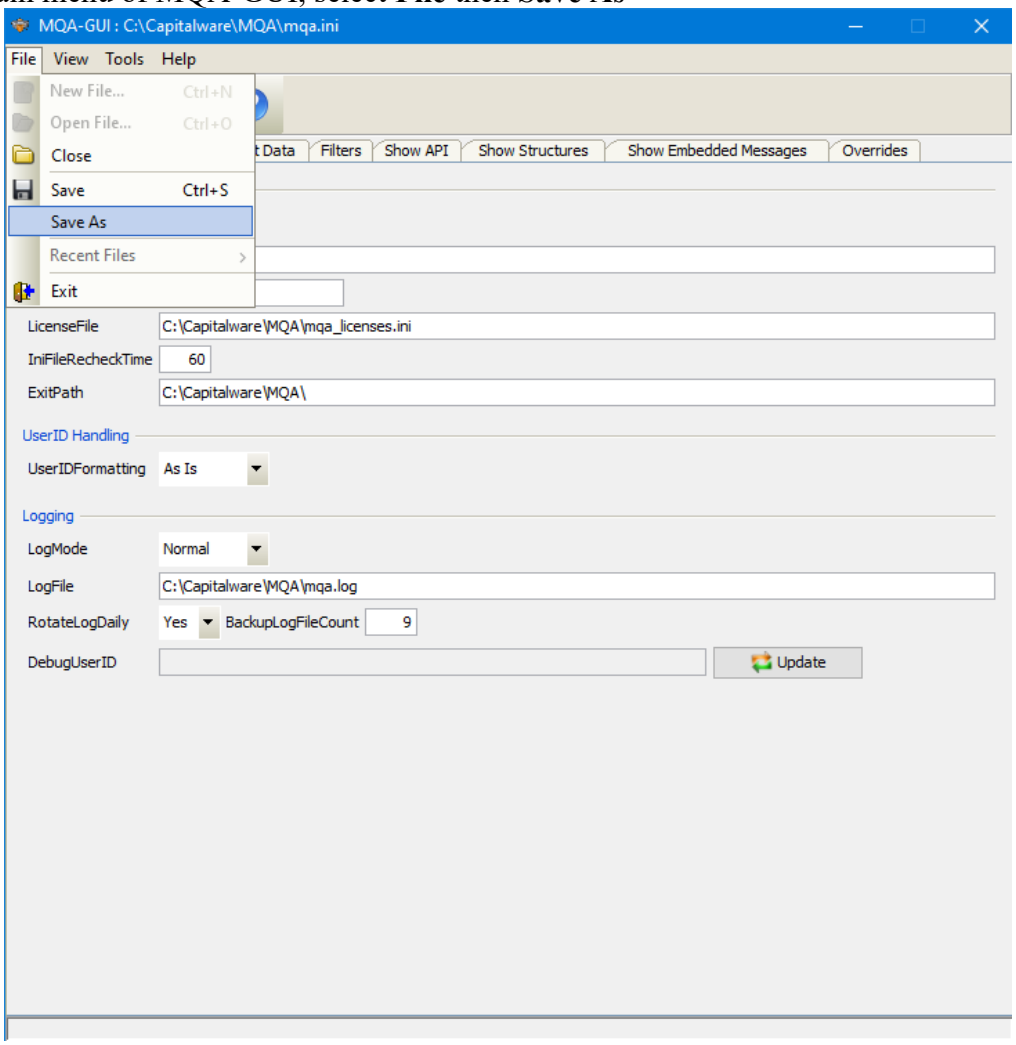


The parameter information will be saved to the IniFile.

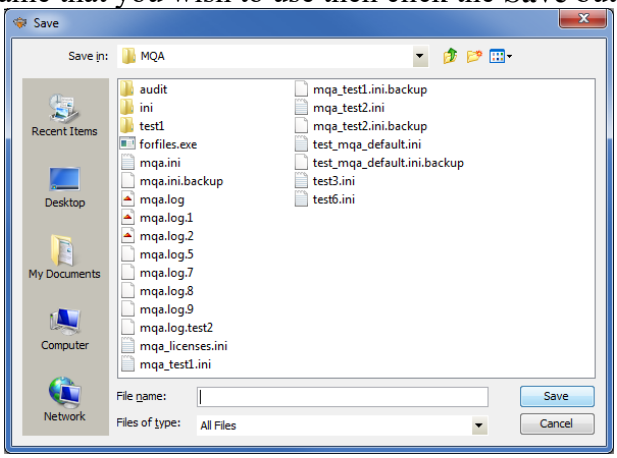
The MQA-GUI program will make a backup of the original file in the same directory and call it: ***{inifilename}.backup***

3.5 Save As IniFile

This section will describe how to save an open MQA IniFile with a new file name.
From the main menu of MQA-GUI, select **File** then **Save As**



Select or input the IniFile name that you wish to use then click the **Save** button.

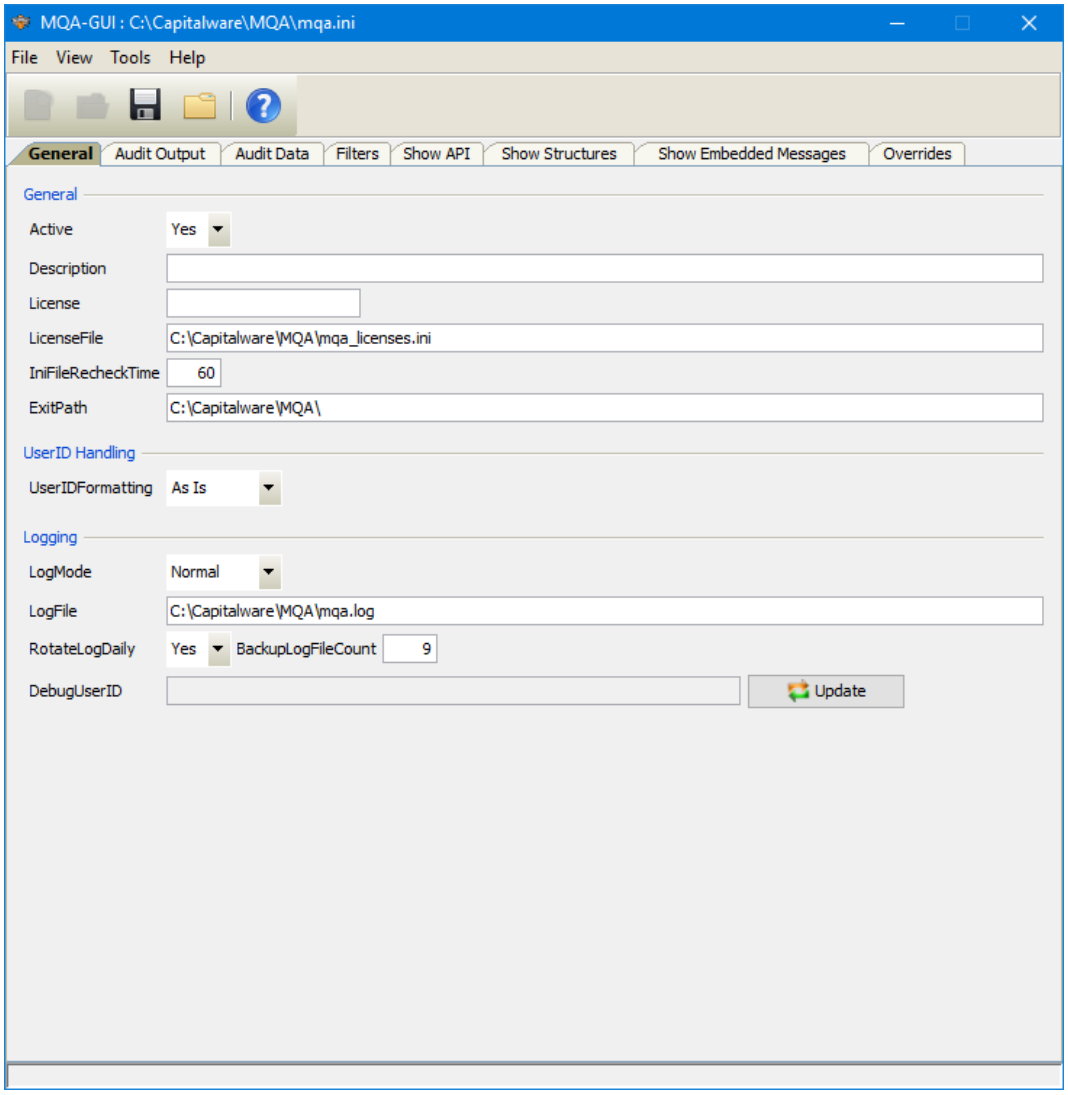


4 Tabs

This chapter will describe the various parameters on each tab.

4.1 General Tab

This section will describe various parameters on the General tab of a MQA IniFile.
The following is a General Tab with default values:



The IniFile keywords and their values on the General Tab are discussed below.

4.1.1 General

This section describes the general keywords and their values.

➢ **Active** controls enabling or disabling MQA.

➢ **Description** parameter is optional and is not used by MQA. It can be used to provide a brief description / purpose of the IniFile.

➢ **License key** is provided by Capitalware Inc. The license is for a particular queue manager. Your license will look something like: 1000-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).

➢ **LicenseFile** specifies the location of License file that contains all of the customer's license keys. The following are the default values for LicenseFile:

- For Windows: **C:\Capitalware\MQA\mqa_licenses.ini**
- For IBM MQ 32-bit on Unix & Linux: **/var/mqm/exits/mqa_licenses.ini**
- For IBM MQ 64-bit on Unix and Linux: **/var/mqm/exits64/mqa_licenses.ini**
- For IBM i: **/QIBM/UserData/mqm/mqa/mqa_licenses.ini**

➢ **IniFileRecheckTime** specifies the amount, in seconds, before the IniFile is checked whether it has changed or not. The default value is 60.

➢ **ExitPath** explicitly sets the API exit path.

4.1.2 UserID Handling

This section describes the keywords related to UserID handling.

➢ **UserIDFormatting** controls the handling of the UserID. It supports 3 values [A / U / L]. (As Is, Uppercase and Lowercase). The default value is A.

4.1.3 Logging

This section describes the keywords related to logging and their values.

- **LogMode** specifies what type of logging the user wishes to use. LogMode supports 4 values: Q (Quiet), N (Normal), V (Verbose) and D (Debug). The default value is N.

- **LogFile** specifies the location of the log file. The default values are as follows:
  - For Windows: **C:\Capitalware\MQA\mqa.log**
  - For IBM MQ 32-bit on Unix and Linux: **/var/mqm/exits/mqa.log**
  - For IBM MQ 64-bit on Unix and Linux: **/var/mqm/exits64/mqa.log**
  - For IBM I: **/QIBM/UserData/mqm/mqa/mqa.log**

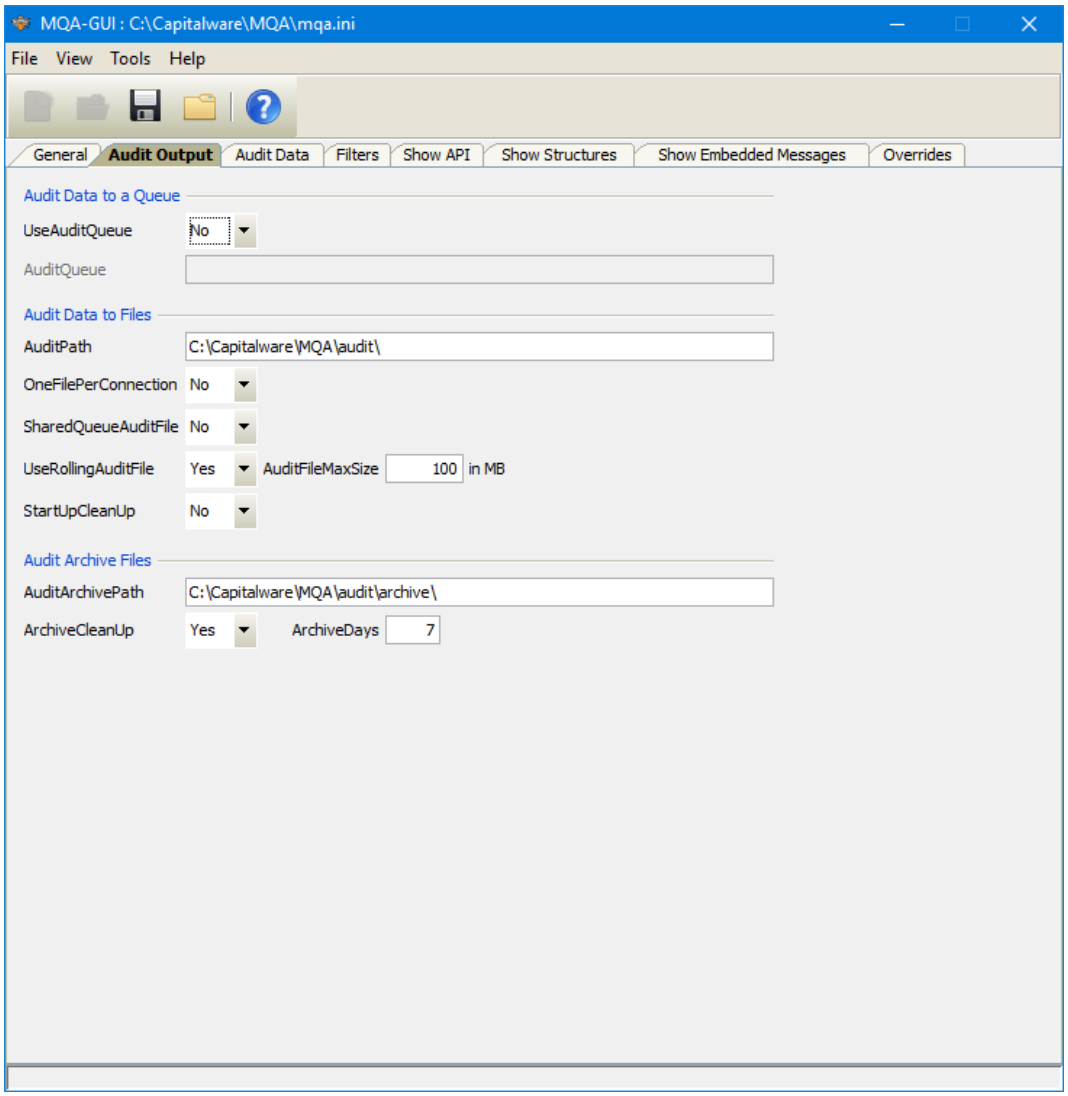
- **RotateLogDaily** specifies whether or not the daily log file rotation should take place. The default value is Y (Yes).

- **BackupLogFileCount** specifies the number of backup log files to be kept / retained by MQA. The default value is 9.

- **DebugUserID** allows MQA to limit debugging output to specific UserIDs. The default value is an empty field.

4.2 Audit Output Tab

This section will describe various parameters on the General tab of a MQA IniFile.
The following is a General Tab with default values:



The IniFile keywords and their values on the General Tab are discussed below.

4.2.1 Audit Data to a Queue

This section describes the keywords related to directing the audit data to an Audit Queue.

- **UseAuditQueue** controls the use of AuditQueue. Set to Y to activate feature.

- **AuditQueue** is either a local or remote queue of where the audit data is to be outputted to.

4.2.2 Audit Data to Files

This section describes the keywords related to directing the audit data to Audit Files (QMgr and Queue).

- **AuditPath** specifies path to the Audit files Setting this parameter will override the default value for AuditPath. The default values are 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/`

- **OneFilePerConnection** allows the MQAdmin wishes to have all audit information outputted only to the Queue Manager Audit file. The default value is N (No). 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).

- **SharedQueueAuditFile** allows the MQAdmin wishes to have all audit information for a particular queue outputted to a single audit queue file. The default value is N (No). Setting 'SharedQueueAuditFile' to 'Y' (Yes) will cause MQA to output all audit queue information to a single Queue Audit file.

- **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 (Yes). Setting 'UseRollingAuditFile' to 'Y' will activate this feature.

- **AuditFileMaxSize** specifies how large an Audit file can become before it is moved to the archive directory. The value represents the maximum size in MB (MegaBytes) for the Audit file 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'.

➢ **StartUpCleanUp** runs the cleanup task of the audit directory on queue manager startup. The default value is N (No). Setting 'StartUpCleanUp' to 'Y' (Yes) will activate this feature.

4.2.3 Audit Archive Files

This section describes the keywords related to Audit Archive files.

➢ **AuditArchivePath** specifies the path to the Audit files in the archive directory 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/
```

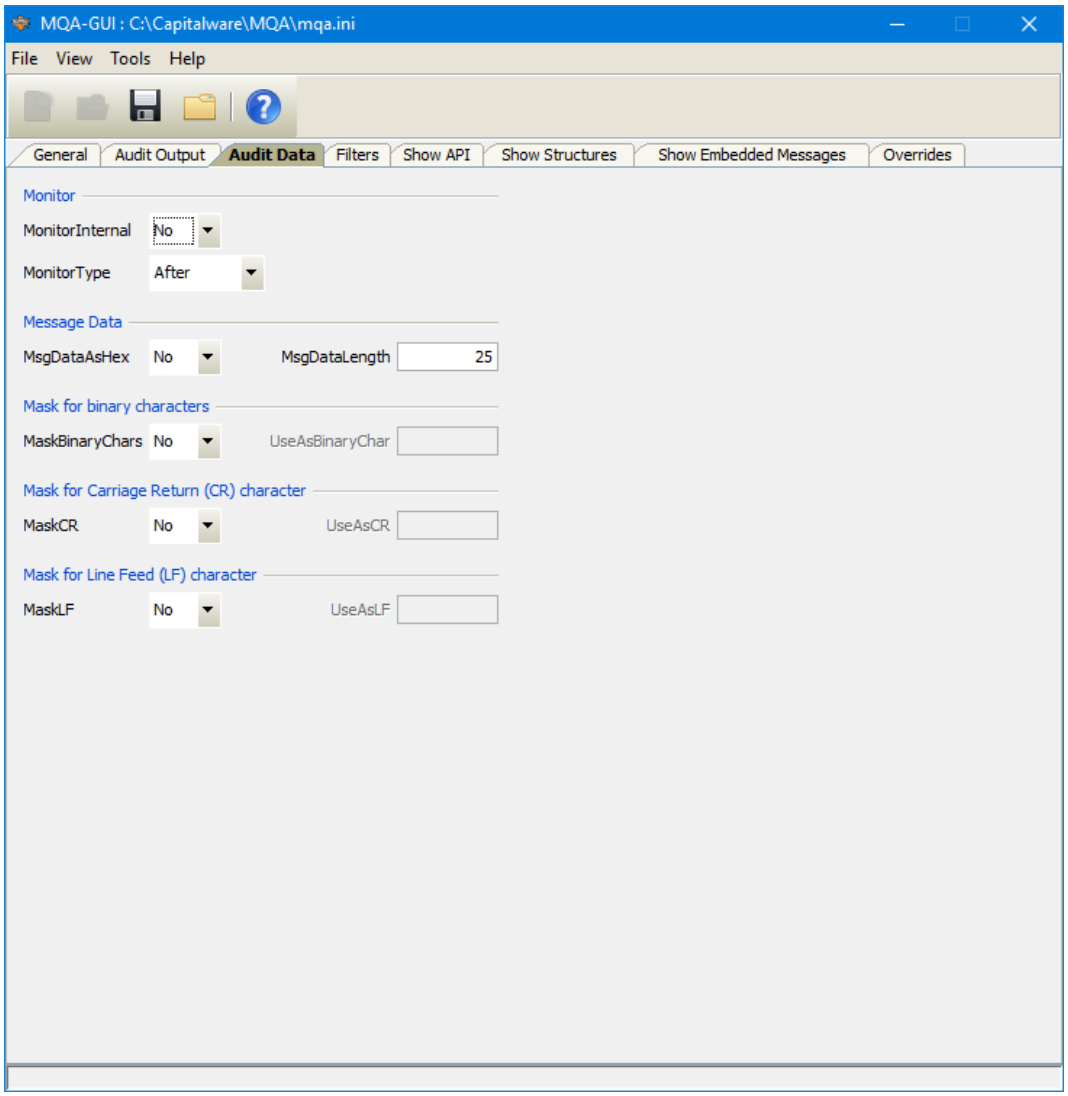
For IBM MQ on IBM i:
```
AuditArchivePath=/QIBM/UserData/mqm/mqa/audit/archive/
```

➢ **ArchiveCleanUp** turns on the cleanup of the archive directory. The default value is Y (Yes).

➢ **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'.

4.3 Audit Data Tab

This section will describe various parameters on the General tab of a MQA IniFile.
The following is a General Tab with default values:



The IniFile keywords and their values on the General Tab are discussed below.

4.3.1 Monitor

This section describes the monitor related keywords and their values.

- **MonitorInterval** enables monitoring of internal MQ applications issuing MQ API calls.

- **MonitorType** specifies what type of monitoring the user wishes to have. i.e. Both, After & Conversion

4.3.2 Message Data

- **MsgDataAsHex** specifies how MQA will handle the message data when the audit information is generated. Setting 'MsgDataAsHex' to 'Y' (Yes) will cause the message data to be converted to hexadecimal format in the audit record.

- **MsgDataLength** specifies how much of the message data will be generated with the audit information

4.3.3 Mask for Binary Characters

This section describes the 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)

4.3.4 Mask for Carriage Return Character

This section describes the keywords related to 'Mask for Carriage Return' (CR) 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)

4.3.5 Mask for Line Feed Character

This section describes the keywords related to 'Line Feed' (LF) character.

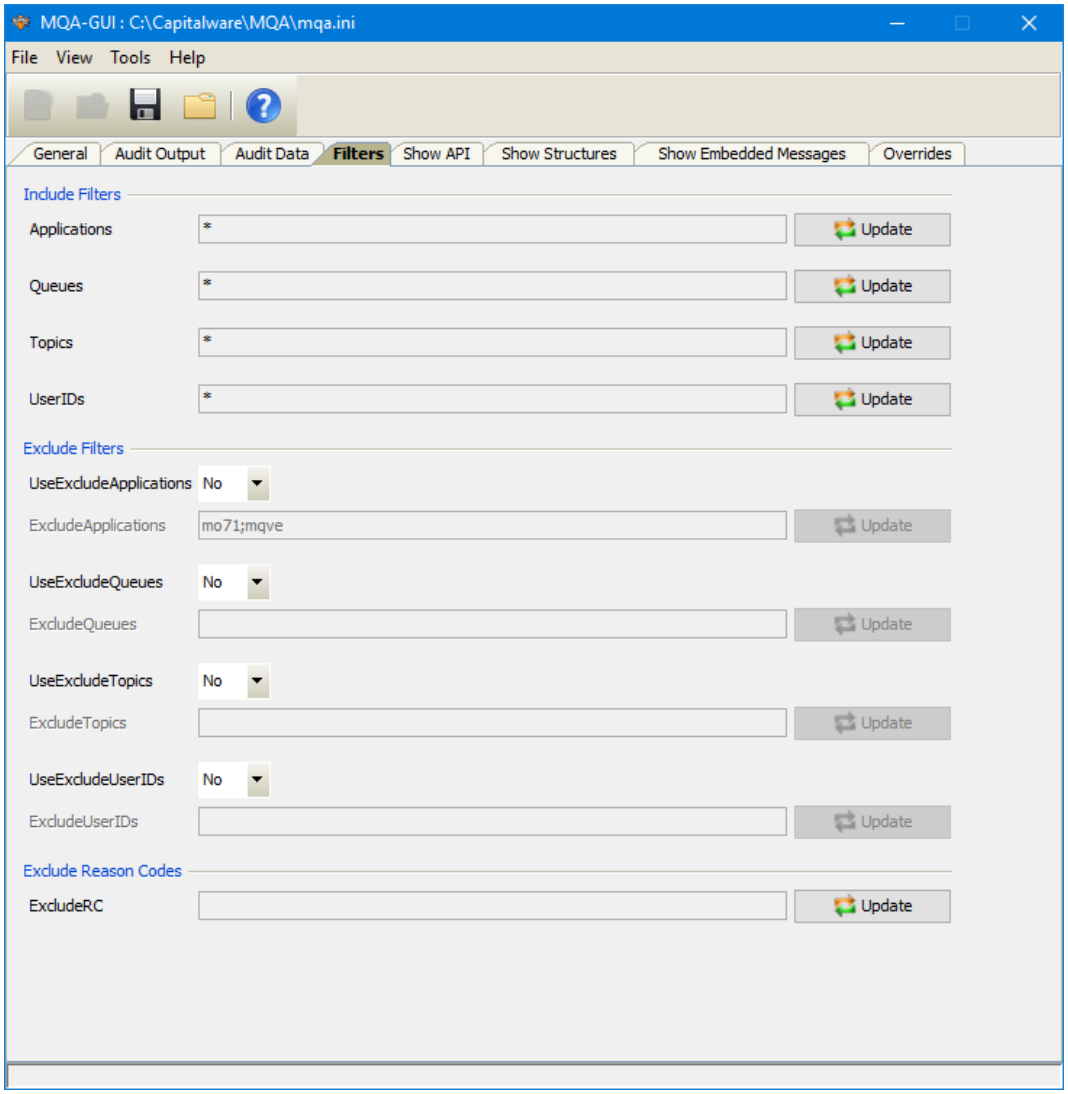
1. **MaskLF** controls whether UseAsLF is used or not. Set to Y to activate feature.

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)

4.4 Filters Tab

This section will describe the parameters on the Filters tab of a MQA IniFile.

The following is the Filters Tab with sample values:



The following are the IniFile parameters on the Filters Tab:

4.4.1 Include Filters

This section describes the Include Filters keywords and their values.

- **Applications** specifies how to selectively monitor applications. The default value is '*' (monitor all applications).

- **Queues** specifies how to selectively monitor queues. The default value is '*' (monitor all queues).

- **Topics** specifies how to selectively monitor topics. The default value is '*' (monitor all topics).

- **UserIDs** specifies how to selectively monitor UserIDs connecting to the queue manager. The default value is '*' (monitor all UserIDs).

4.4.2 Exclude Filters

This section describes the Exclude Filter keywords and their values.

- **UseExcludesApplications** controls the use of ExcludeApplications. Set to Y to activate feature.

- **ExcludesApplications** specifies which applications should not be tracked.

- **UseExcludesQueues** controls the use of ExcludeQueues. Set to Y to activate feature.

- **ExcludesQueues** specifies which queues should not be tracked.

- **UseExcludesTopics** controls the use of ExcludeTopics. Set to Y to activate feature.

- **ExcludesTopics** specifies which topics should not be tracked.

- **UseExcludesUserIDs** controls the use of ExcludeUserIDs. Set to Y to activate feature.

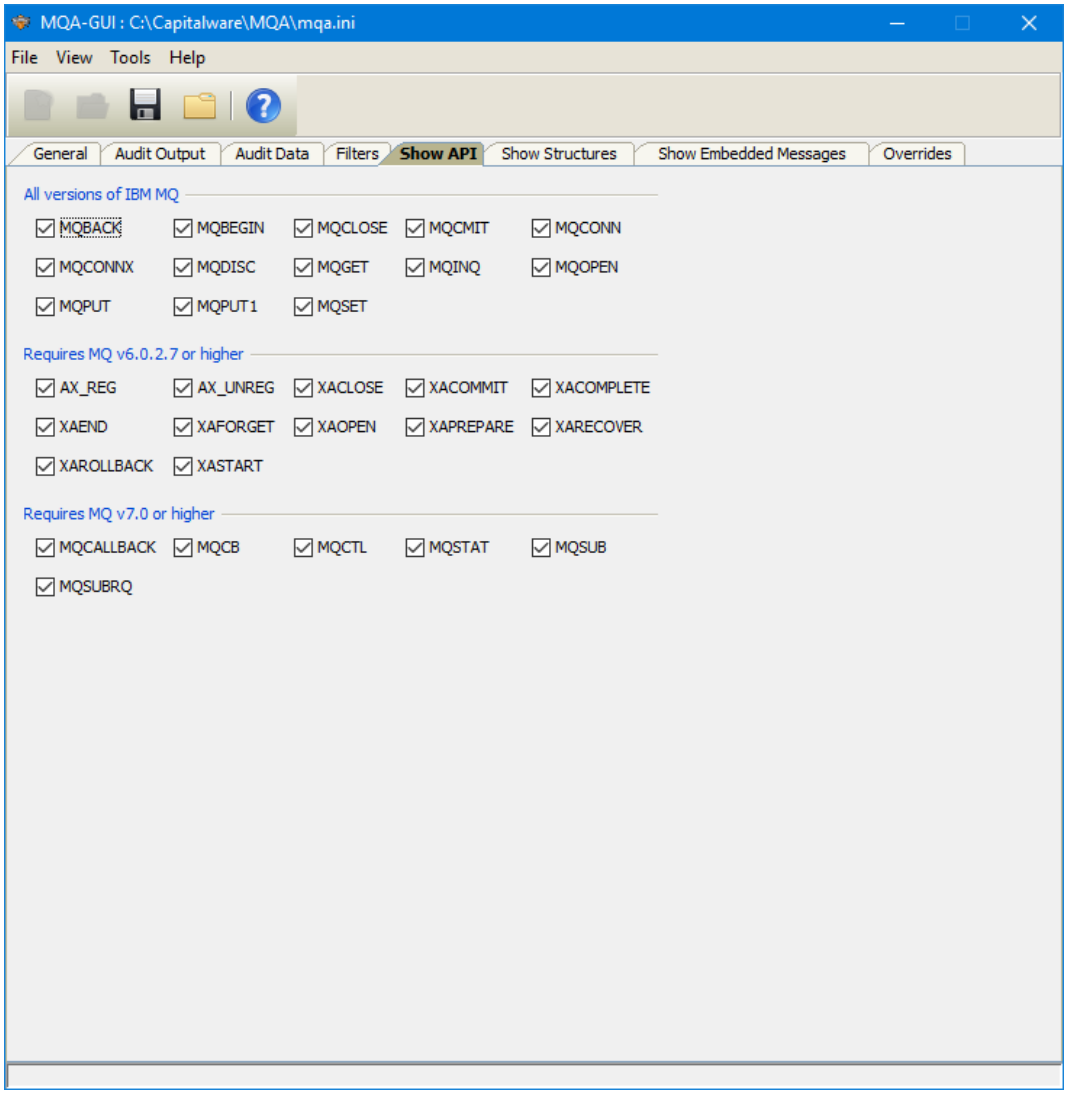
- **ExcludesUserIDs** specifies which queues should not be tracked.

- **ExcludeRC** specifies the MQ Reason Codes to not generate Audit information.

4.5 Show API Tab

This section will describe the parameters on the Show API tab of a MQA IniFile.

The following is the Show API Tab with sample values:



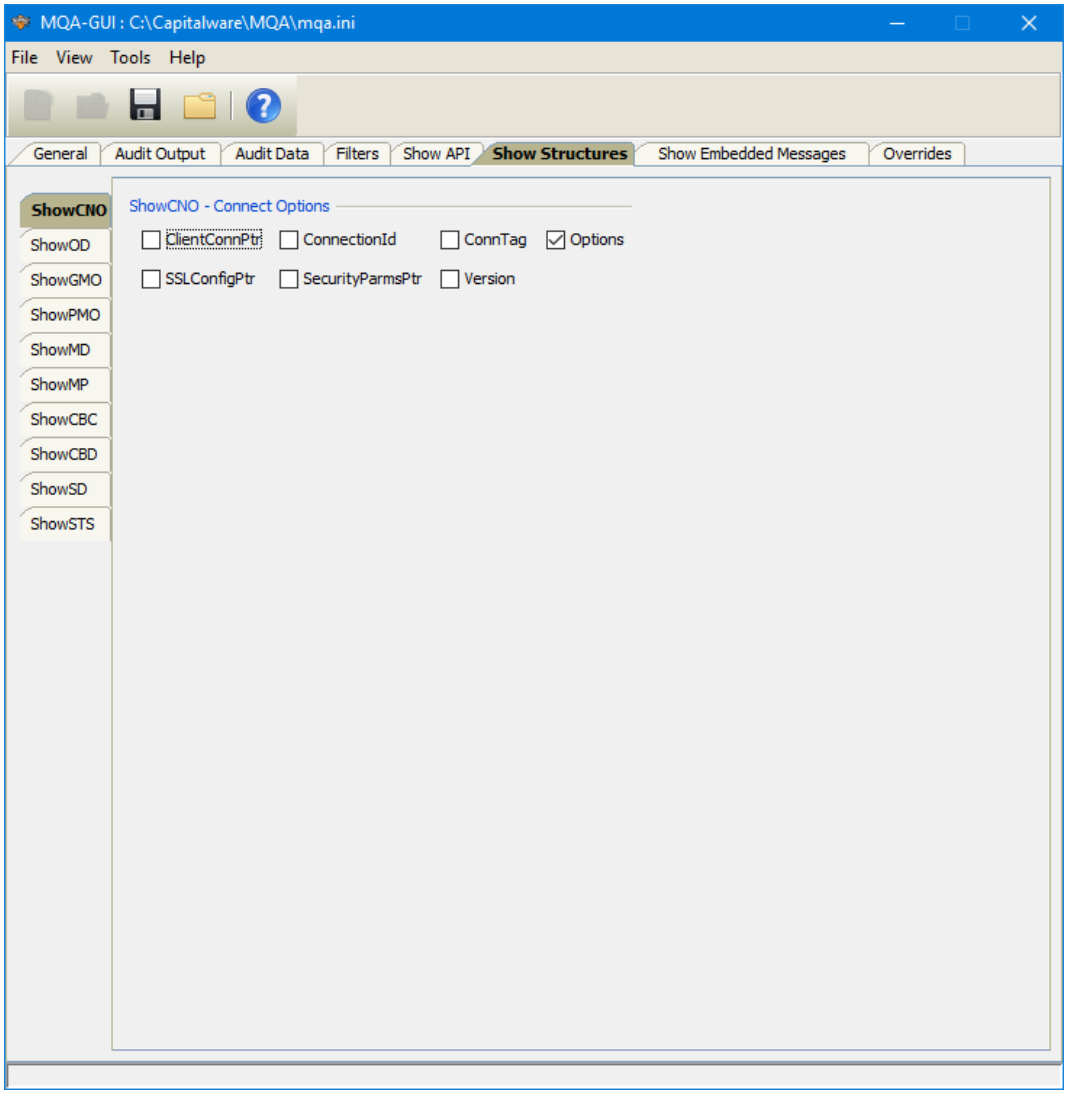
The one parameter is ShowAPI.

- **ShowAPI** specifies the MQ API calls that the user wants MQA to generate in the audit information.

4.6 Show Structures Tab

This section will describe various parameters on the Show Structures tab of a MQA IniFile.

The following is the Show Structures Tab with sample values:



The parameters on the Show Structures Tab are as follows:

- **ShowCNO** specifies the MQCNO (Connect Options) fields of the MQCONN / MQCONNX API calls that the user wants MQA to generate in the audit information.

- **ShowOD** specifies the MQOD (Object Descriptor) fields of the MQOPEN and MQPUT1 API calls that the user wants MQA to generate in the audit information.

- **ShowGMO** specifies the MQGMO (Get Message Options) fields of the MQGET API call that the user wants MQA to generate in the audit information.

- **ShowPMO** specifies the MQPMO (Put Message Options) fields of the MQPUT and MQPUT1 API calls that the user wants MQA to generate in the audit information.

- **ShowMD** specifies the MQMD (Message Descriptor) fields of the MQGET, MPUT and MQPUT1 API calls that the user wants MQA to generate in the audit information.

- **ShowMP** specifies the message properties that the user wants MQA to generate in the audit information.

- **ShowCBC** specifies the MQCBC (Call Back Context) fields of the CallBack API call that the user wants MQA to generate in the audit information.

- **ShowCBD** specifies the MQCBD (Call Back Data Descriptor) fields of the CallBack API call that the user wants MQA to generate in the audit information.

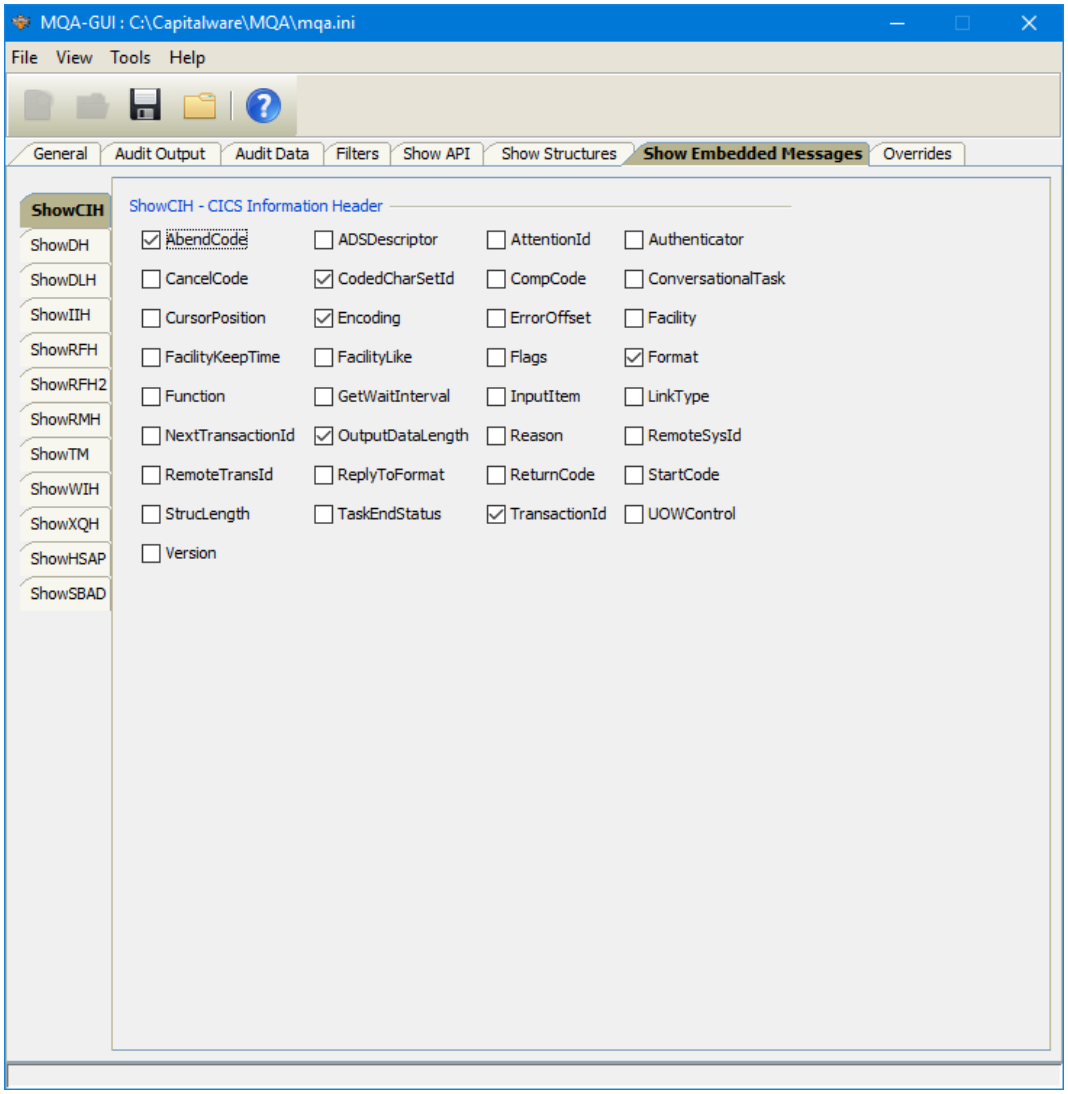
- **ShowSD** specifies the MQSD (Subscription Descriptor) fields of the SUB API call that the user wants MQA to generate in the audit information.

- **ShowSTS** specifies the MQSTS (Status Information Record) fields of the STAT API call that the user wants MQA to generate in the audit information.

4.7 Show Embedded Messages Tab

This section will describe various parameters on the Show Embedded Messages tab of a MQA IniFile.

The following is the Show Embedded Messages Tab with sample values:



The IniFile parameters on the Show Embedded Messages Tab are as follows:

- **ShowCIH** specifies the MQCIH (CICS Information Header) fields that the user wants MQA to generate in the audit information.

- **ShowDH** specifies the MQDH (Distribution Header) fields that the user wants MQA to generate in the audit information.

- **ShowDLH** specifies the MQDLH (Dead Letter Header) fields that the user wants MQA to generate in the audit information.

- **ShowIIH** specifies the MQIIH (IMS Information Header) fields that the user wants MQA to generate in the audit information.

- **ShowRFH** specifies the MQRFH (Rules and Formatting Header) fields that the user wants MQA to generate in the audit information.

- **ShowRFH2** specifies the MQRFH2 (Rules and Formatting Header 2) fields that the user wants MQA to generate in the audit information.

- **ShowRMH** specifies the MQRMH (Reference Message Header) fields that the user wants MQA to generate in the audit information.

- **ShowTM** specifies the MQTM (Trigger Message) fields that the user wants MQA to generate in the audit information.

- **ShowWIH** specifies the MQWIH (Work Information Header) fields that the user wants MQA to generate in the audit information.

- **ShowXQH** specifies the MQXQH (Transmission Queue Header) fields that the user wants MQA to generate in the audit information.

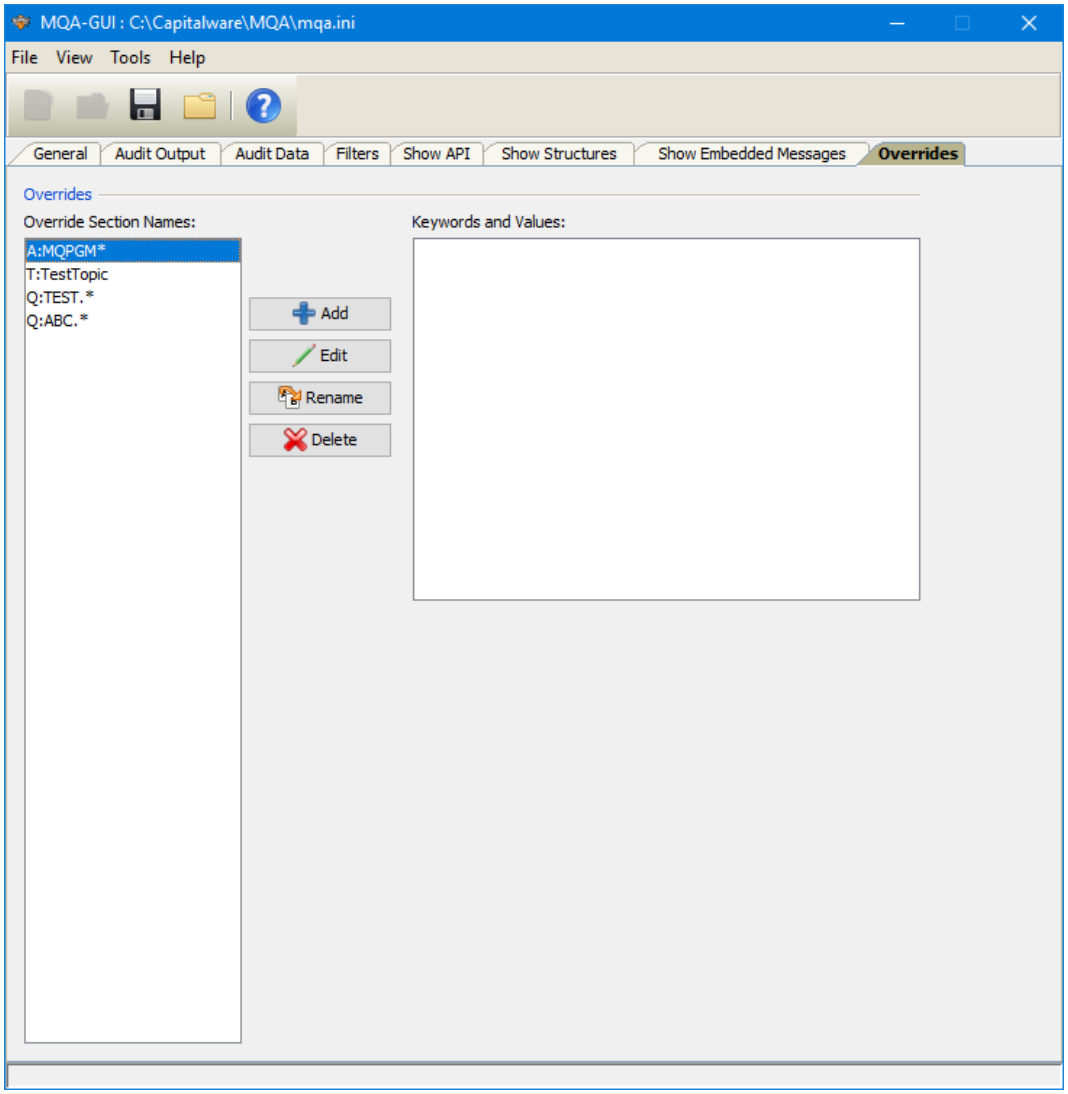
- **ShowHSAP** specifies the MQHSAP (SAP R/3 Header) fields that the user wants MQA to generate in the audit information.

- **ShowSBAD** specifies the SMQBAD (SAP R/3 Bad Message Header) fields that the user wants MQA to generate in the audit information.

4.8 Overrides Tab

This section will describe various parameters on the Overrides tab of a MQA IniFile.

The following is the Overrides Tab with sample values:

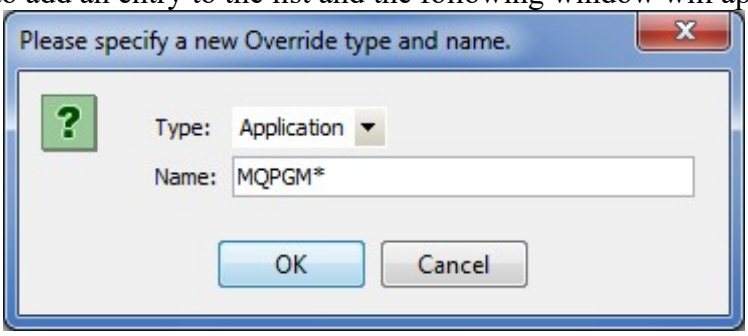


This section describes IniFile keywords that override the global IniFile values. To explicitly override the global IniFile values by Application Name, UserID, Queue Name or Topic, the user can create an Inifile stanza to hold those specific values.

4.8.1 Managing Overrides

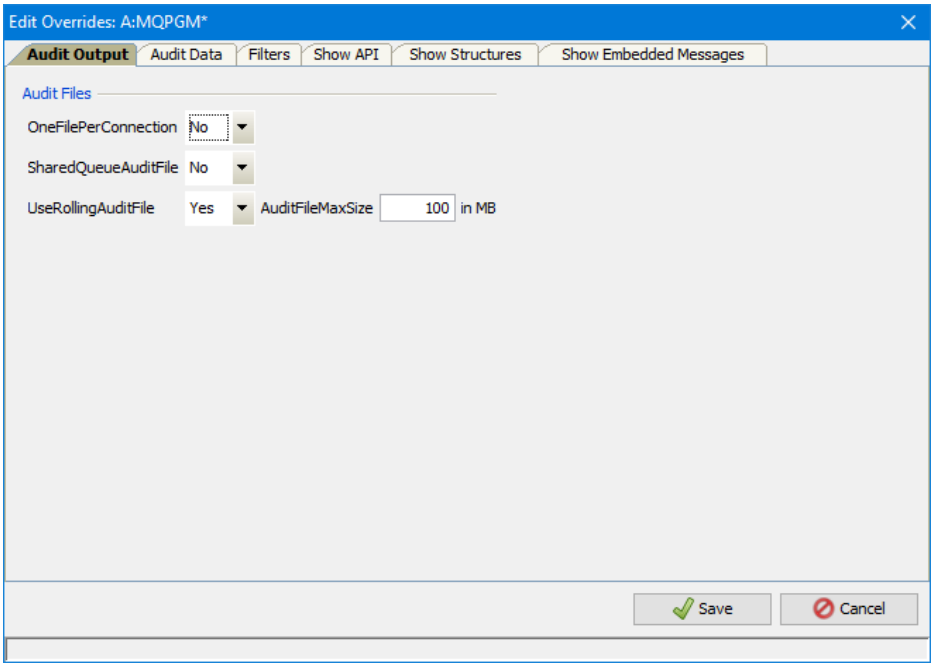
This section describes the necessary steps to manage Application Name filtering entries.

Click the **Add** button to add an entry to the list and the following window will appear.



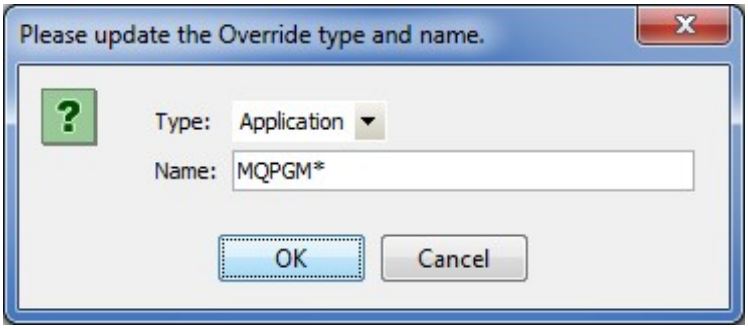
Select the Type (Applications, Queues or UserIDs) and input the Application, Queue or UserID name including wildcards and then click the **Ok** button.

Click the **Edit** button to update an entry in the list and the following window will appear.



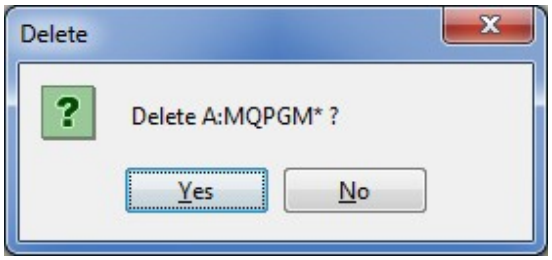
Then update any field to be overridden and then click the **Save** button at the bottom

Click the **Rename** button to delete an entry in the list and the following window will appear.



Then update the override type and/or name then click the **Ok** button.

Click the **Delete** button to delete an entry in the list and the following window will appear.



Then click the **Yes** button to permanently remove the entry from the list.

5 Other Tools

This chapter will describes how the other tools included with MQA-GUI.

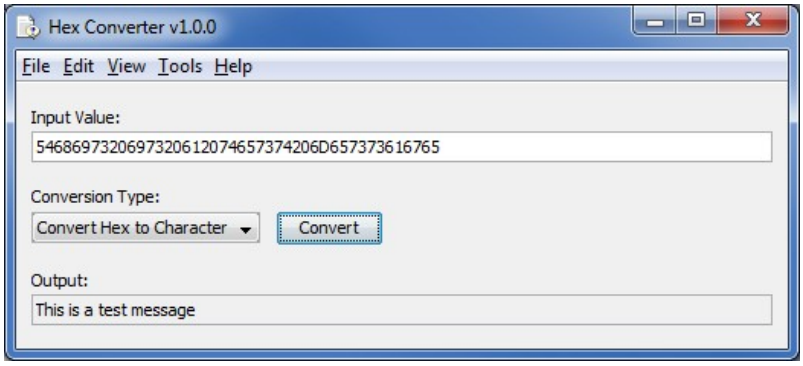
5.1 Hex Converter

This section will describe how to invoke and use the Hex Converter tool. Hex Converter tool can convert a hexadecimal number to its corresponding string representation and vice versa. i.e. 0x'3132333041424344' to '1230ABCD'.

For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **Hex Converter**.

This is the main panel of the Hex Converter.



- The user first selects the Conversion type: "Hex to Character" or "Character to Hex"
- Input the value hex or character value into the "Input Value" field
- Click the Convert button to convert the data

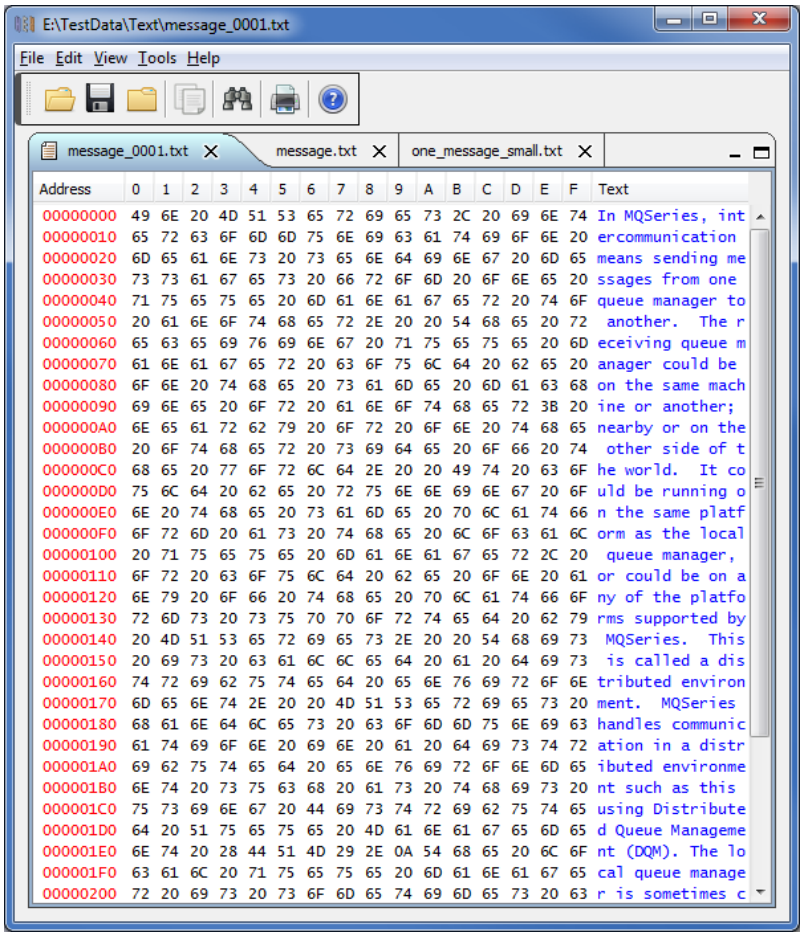
5.2 Hex Editor

This section will describe how to invoke and use the Hex Editor tool. The Hex Editor is a simple multipurpose hex editor. The Hex Editor supports multi-file editing.

For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **Hex Editor**.

This is the main panel of the Hex Editor and it has 3 open files.



Supported features:

- Open / editing of multiple files
- Recent File List
- Copy, Find and Find Next
- Save, Save As and Print

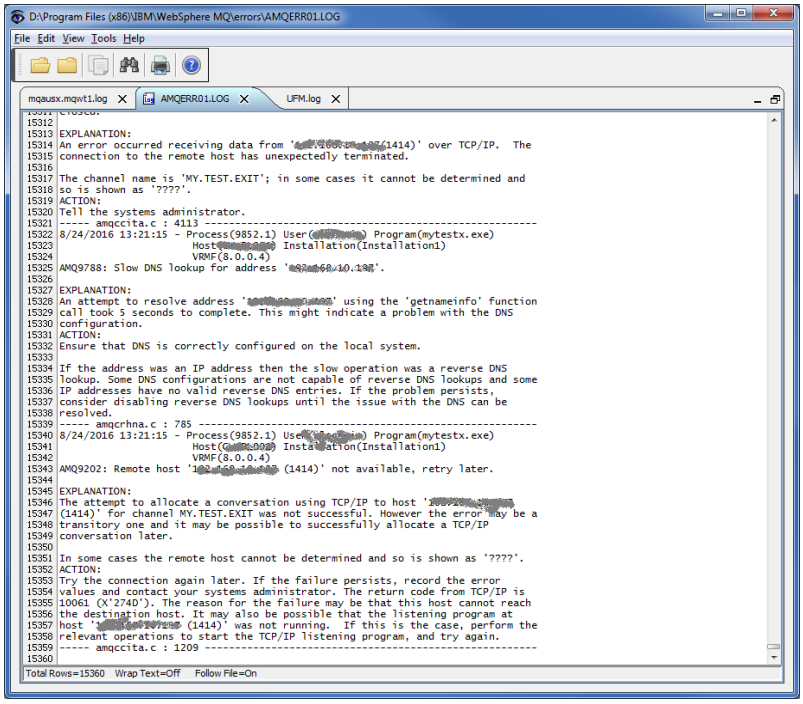
5.3 Tail File

This section will describe how to invoke and use the Tail File tool. The Tail File is a simple multipurpose file monitoring viewer. The Tail File supports monitoring of multiple files at the same time.

For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **Tail File**.

This is the main panel of the Tail File and it has 3 open files.



Supported features:

- Opening of multiple files
- Wrap Text and Recent File List
- Copy, Find, Find Next and Print

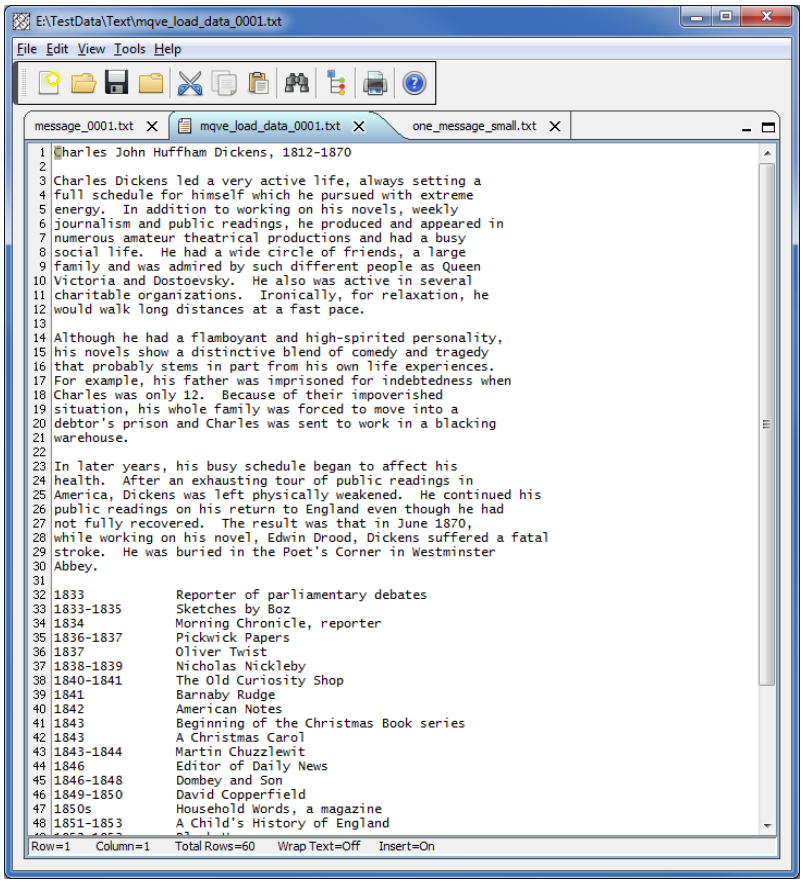
5.4 Text Editor

This section will describe how to invoke and use the Text Editor tool. The Text Editor is a simple multipurpose plain-text editor. The Text Editor supports multi-file editing.

For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **Text Editor**.

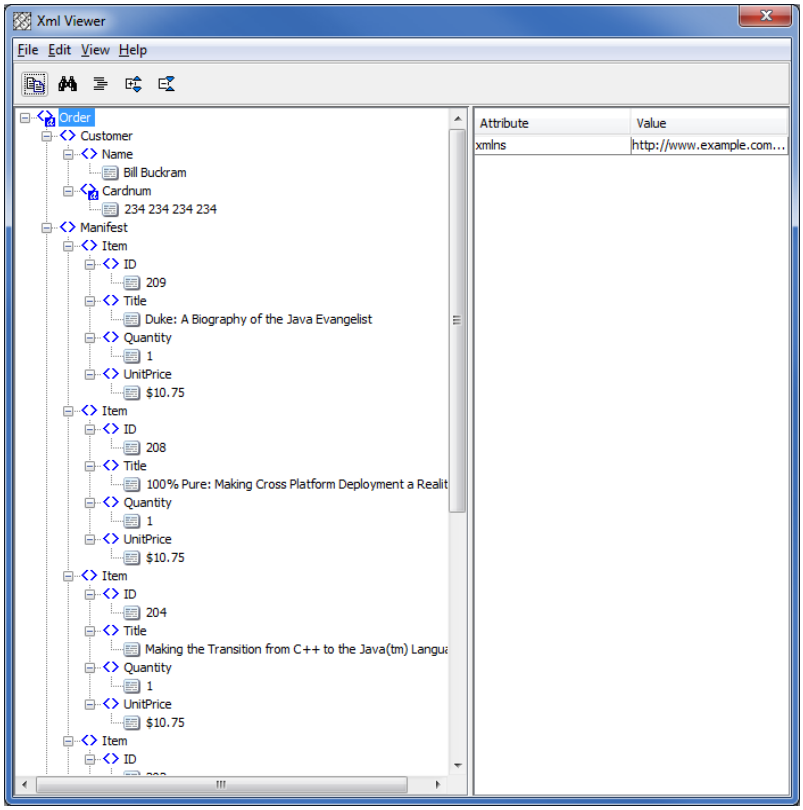
This is the main panel of the Text Editor and it has 3 open files.



Supported features:

- Open / editing of multiple files
- Builtin XML Viewer, Wrap Text
- Recent File List
- Cut, Copy, Paste, Find and Find Next
- Save, Save As and Print

The following shows the Text Editor's builtin XML Viewer.



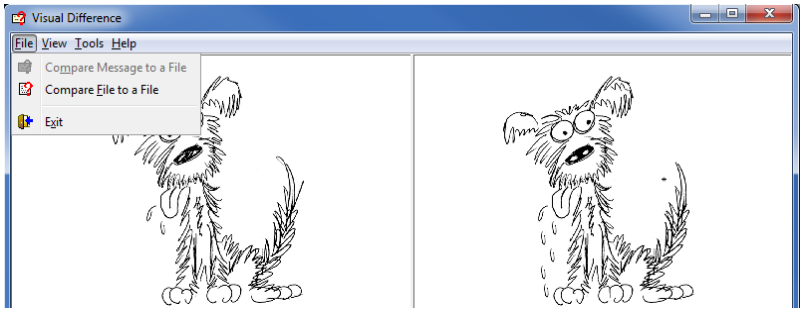
5.5 Visual Difference

This section will describe how to invoke and use the Visual Difference tool. The Visual Difference is a simple file to file comparison tool.

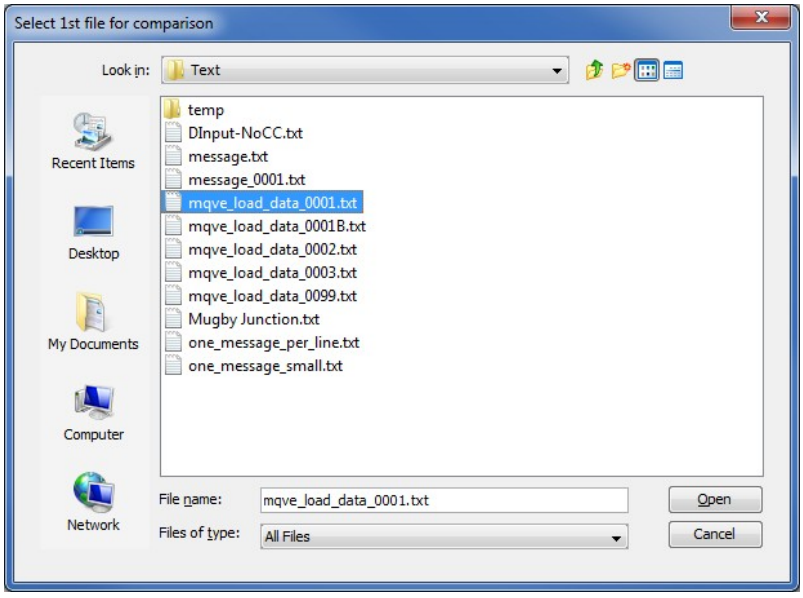
For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **Visual Difference**.

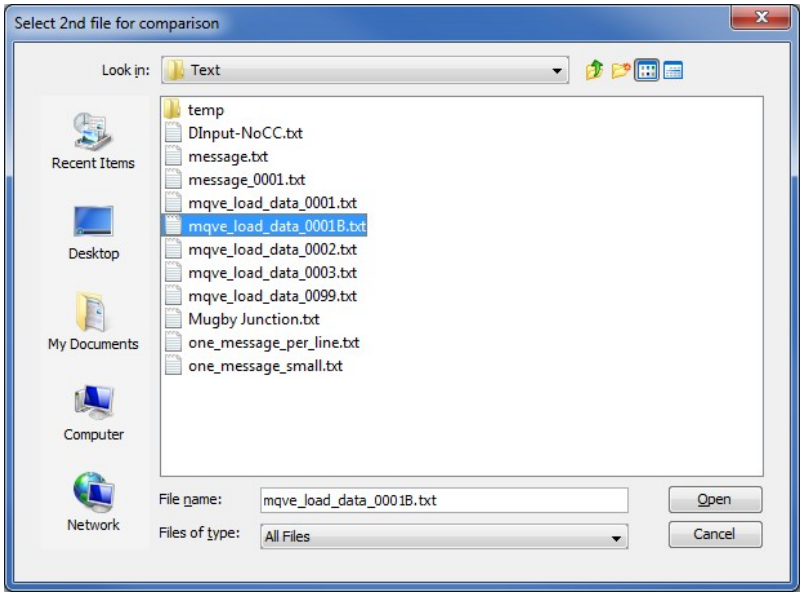
This is the main panel of the Visual Difference. To initiate the file comparison process, click File and then click Compare File to File



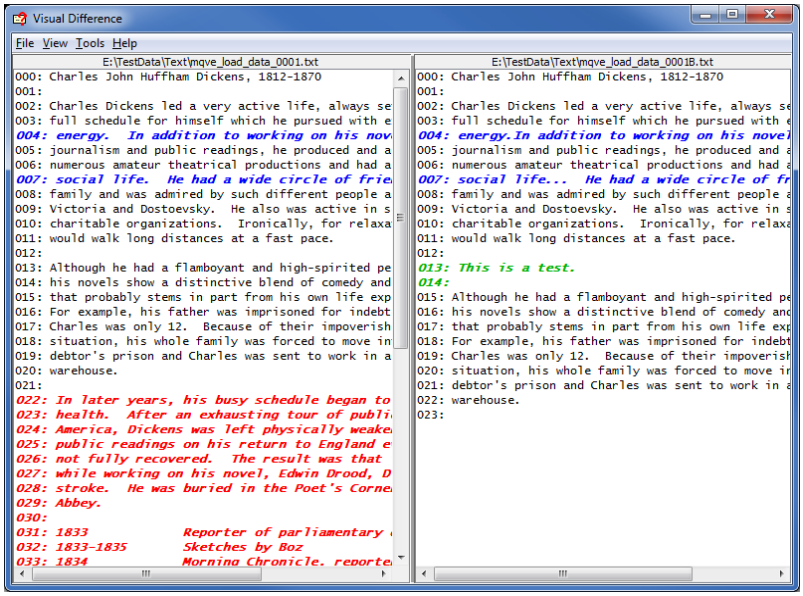
First the user needs to select the first file for comparison.



Next, the user needs to select the second file for the comparison.



The following shows the difference between 2 files. Differences are highlighted: Green for inserts, Red for deletions and Blue for changes.



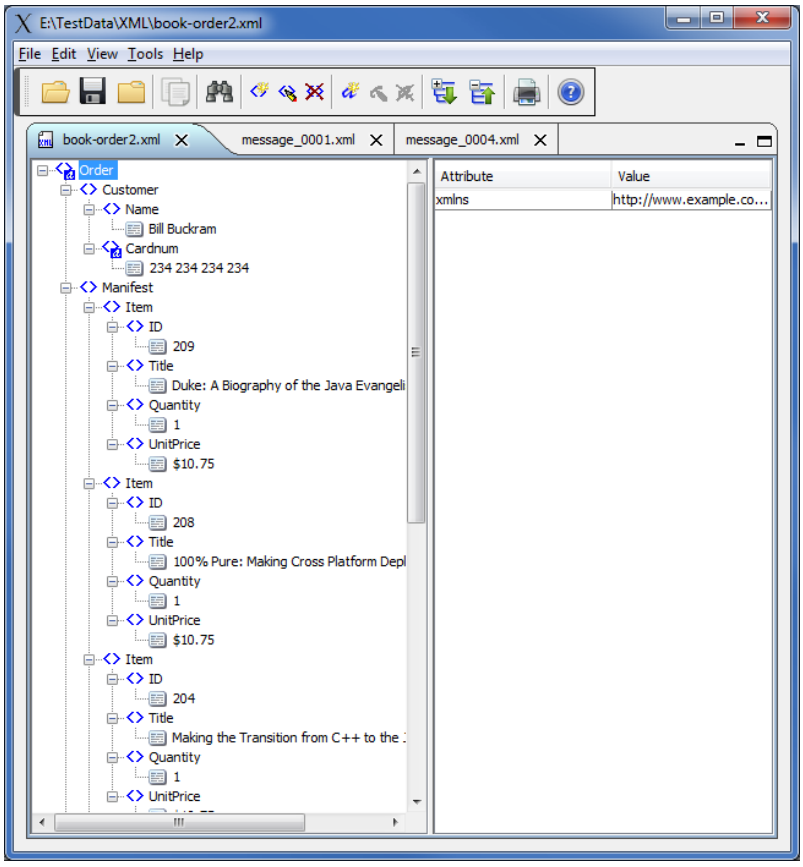
5.6 XML Editor

This section will describe how to invoke and use the XML Editor tool. The XML Editor is a simple multipurpose plain-text editor. The XML Editor supports multi-file editing.

For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQA-GUI.

From the main panel of MQA-GUI, select **Tools**, then **XML Editor**.

This is the main panel of the XML Editor and it has 3 open files.



Supported features:

- Open / editing of multiple files
- Recent File List
- Copy, Find and Find Next
- Save, Save As and Print

6 Miscellaneous

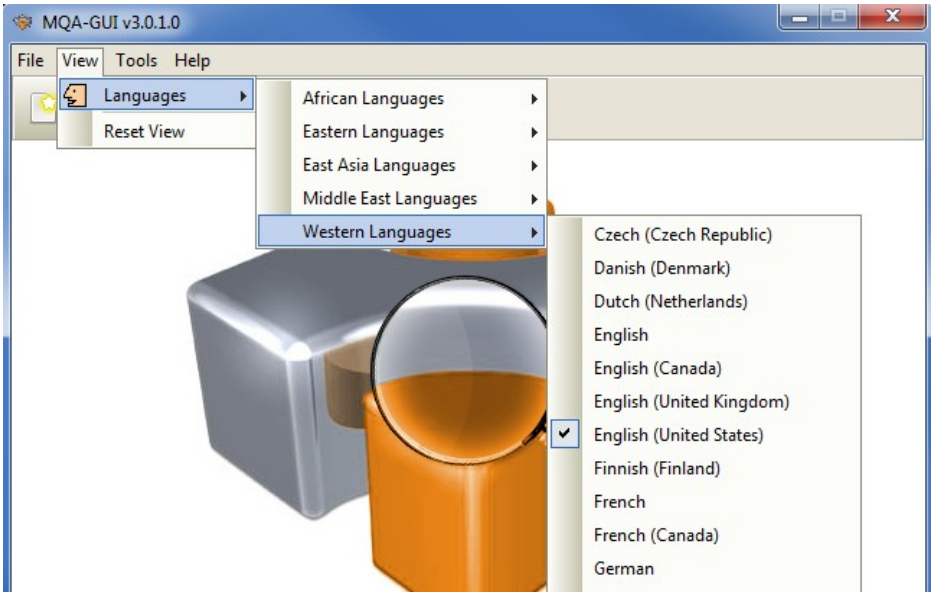
This chapter will describes how the miscellaneous features of MQA-GUI.

6.1 Language

This section will describe how to switch MQA-GUI's Language Text to another Language Text. MQA-GUI has been designed to support Java's multi-language environment. Currently, MQA-GUI has language support (menu items, buttons and titles) for the following 55 languages:

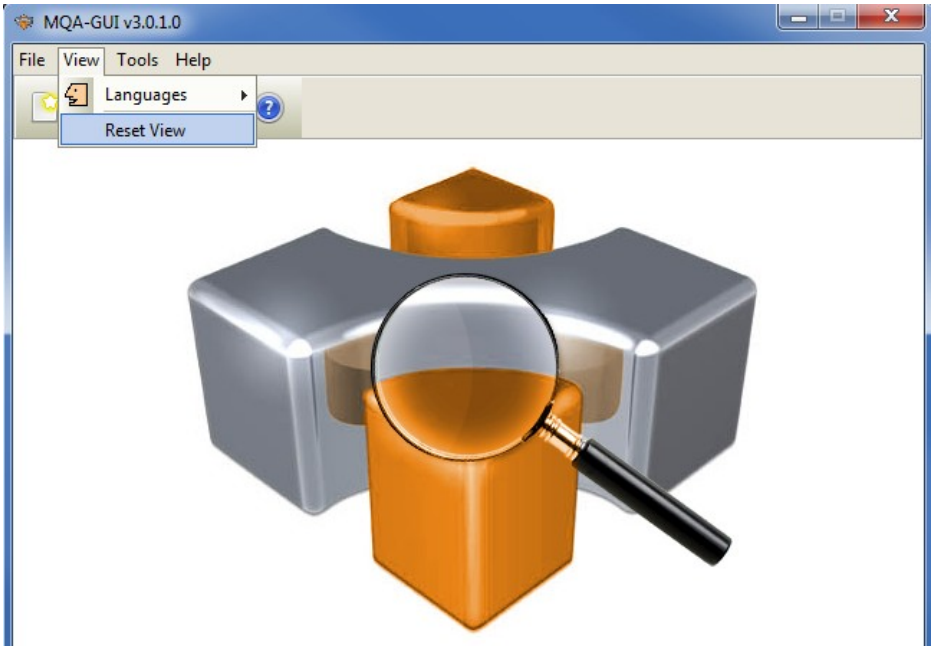
➢ Amharic (am)	➢ Hungarian (hu)	➢ Shona (sn)
➢ Arabic (ar)	➢ Igbo (ig)	➢ Sindhi (sd)
➢ Azerbaijani (az)	➢ Indonesian (in)	➢ Spanish (es)
➢ Bengali (bn)	➢ Italian (it)	➢ Sundanese (su)
➢ Cebuano (ceb)	➢ Japanese (ja)	➢ Swahili (sw)
➢ Chinese (China) (zh)	➢ Javanese jv)	➢ Swedish (sv)
➢ Chinese (Taiwan) (zh)	➢ Kannada (kn)	➢ Tamil (ta)
➢ Czech (cs)	➢ Korean (ko)	➢ Telugu (te)
➢ Danish (da)	➢ Malay (ms)	➢ Thai (th)
➢ Dutch (nl)	➢ Malayalam (ml)	➢ Turkish (tr)
➢ English (en)	➢ Marathi (mr)	➢ Ukrainian (uk)
➢ Finnish (fi)	➢ Norwegian (no)	➢ Urdu (ur)
➢ French (fr)	➢ Panjabi (pa)	➢ Uzbek (uz)
➢ German (de)	➢ Pashto (ps)	➢ Vietnamese (vi)
➢ Greek (el)	➢ Persian (fa)	➢ Xhosa (xh)
➢ Gujarati (gu)	➢ Polish (pl)	➢ Yoruba (yo)
➢ Hausa (ha)	➢ Portuguese (pt)	➢ Zulu (zu)
➢ Hebrew (iw)	➢ Romanian (ro)	
➢ Hindi (hi)	➢ Russian (ru)	

From the main panel of MQA-GUI, select **View**, then **Language**.



6.2 Reset View

This section will describe how to reset the view of MQA-GUI if the display is not correct.



The user can also reset the view from the command line by passing in the ***RG*** parameter.

6.2.1 Windows Command Line

On Windows, issue the following command to reset the view of MQA-GUI:

mqagui.exe RG

6.2.2 Linux and macOS Command Line

On Linux and macOS, issue the following command to reset the view of MQA-GUI:

./mqagui.sh RG

7 Appendix A – Frequently Asked Questions (FAQ)

Q. Can IniFiles be create on Windows for MQA on Unix?
A. Yes. Once the IniFile has been saved, the user can copy or ftp (in ASCII mode) the file to the remote server.

8 Appendix B – MQA-GUI Upgrade Procedures

To upgrade an existing installation of MQA-GUI, please do the following in the appropriate section below.

8.1 Windows Upgrade

- Stop all instances of MQA-GUI
- Backup any MQA-GUI data files in the MQA-GUI install directory
- Go to the Windows **Control Panel** and select **Program and Features**
- Select MQA-GUI and then click the **Uninstall** button
- Run the install program called: **mqagui-setup-withjre.exe**
- The installer follows the standard Windows install procedures and provides default values for the user.
- When the install program has completed execution, there will be a newly created folder under **Start** -> **All Programs** called **MQA-GUI**.
- Restore the MQA-GUI data files if necessary
- To run MQA-GUI, select the MQA-GUI icon from within the MQA-GUI folder.

8.2 Linux or macOS Upgrade

- Stop all instances of MQA-GUI
- Backup any MQA-GUI data files in the MQA-GUI install directory
- Remove the installation i.e. rm -Rf Capitalware/MQA-GUI/
- Unzip the file, open a shell prompt and execute the following commands:

  unzip mqagui.zip

- When the unzip command has finished, there should be a directory tree as follows:

  Capitalware/MQA-GUI

- Change directory by typing the following:

  cd Capitalware/MQA-GUI

- Before running MQA-GUI for the first time, you need to add the execute attribute to the mqagui.sh shell scripts. Type the following:

  chmod +x mqagui.sh

- To run MQA-GUI from the Capitalware/MQA-GUI directory, then type the following:

  ./mqagui.sh

9 Appendix C – Support

The support for MQA-GUI can be found at the following location:

By email at:
support@capitalware.com

By regular mail at:

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

10 Appendix D – Summary of Changes

- MQA-GUI v3.2.0
  - Added ‘Audit Output’ and ‘Audit Data’ tabs to better organize the presentation.
  - 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 a bug when resetting the view.

- MQA-GUI v3.1.0
  - Added the ‘Recent Files’ menu option under the File menu option.
  - Added images to buttons for more user-friendly display.
  - Added support for 55 languages for the menu options, buttons and titles (not IniFile keywords).
  - Added the ‘Reset View’ menu option under the View menu option.
  - Added 6 tools under the Tools menu option:
    - Compare File to a File
    - Hex Converter
    - Hex Editor
    - Tail File
    - Text Editor
    - XML Editor

- MQA-GUI v3.0.0
  - Added the ability to filter topics - new keyword: Topics
  - Added the ability to exclude topics - new keywords: UseExcludeTopics & ExcludeTopics
  - Removed the keyword MQAPILevel as it is no longer required.

- MQA-GUI 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

- MQA-GUI v2.3.0
  - Added MQOD v4 fields, MQPMO v3 fields & MQGMO v4 fields
  - Added missing ShowMP keyword under the "Show V7 Structures" tab

- MQA-GUI 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

- MQA-GUI v2.0.0
  - Added Conversion for MonitorType - this causes audit records to be written for before and after conversion for MQGET.

- MQA-GUI 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.

- MQA-GUI v1.0.0
  - Initial release.

11 Appendix E – 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.

12 Appendix F – 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.