

MQME-GUI User Guide

The screenshot shows the MQME-GUI application window with the following configuration details:

- General**
 - Active: Yes
 - Description: [Empty]
 - License: [Empty]
 - LicenseFile: C:\Capitalware\MQME\mqme_licenses.ini
 - IniFileRecheckTime: 60
 - CCSID: 437
 - Encoding: 273
 - ExitPath: C:\Capitalware\MQME\
- UserID Handling**
 - UserIDFormatting: As Is
- Functionality**
 - Perform: Encrypt Only
 - EncryptRFH2Header: Yes
 - PartnerEnabled: No
- Encryption**
 - KeySize: 128
 - UsePP: No
 - EncPassPhrase: [Empty]
 - PassPhrase: [Empty]
- Logging**
 - LogMode: Normal
 - LogFile: C:\Capitalware\MQME\mqme.log
 - RotateLogDaily: Yes
 - BackupLogFileCount: 9
 - DebugUserID: [Empty]

An "Update" button is located at the bottom right of the configuration area.



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. 2011, 2020.

Table of Contents

1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
2 INSTALLATION.....	2
2.1 PREREQUISITES.....	2
2.2 SOFTWARE.....	2
2.3 HARDWARE / OS.....	2
2.4 SETTING UP AND RUNNING THE APPLICATION.....	2
2.4.1 For Windows Users:.....	2
2.4.2 For Linux and Max OS X (requires JRE/JDK):.....	3
3 CREATING / UPDATING INIFILES.....	4
3.1 NEW INIFILE.....	4
3.2 OPEN AN INIFILE.....	5
3.3 CLOSE AN INIFILE.....	6
3.4 SAVE INIFILE.....	7
3.5 SAVE AS INIFILE.....	8
4 TABS.....	9
4.1 GENERAL TAB.....	9
4.1.1 General.....	10
4.1.2 UserId Handling.....	10
4.1.3 Functionality.....	11
4.1.4 Encryption.....	11
4.1.5 Logging.....	12
4.2 FILTERS TAB.....	13
4.2.1 Exclude Filters.....	14
4.3 PROTECTED QUEUES TAB.....	15
4.3.1 Managing Protected Queues.....	16
4.4 PROTECTED TOPICS TAB.....	20
4.4.1 Managing Protected Topics.....	21
5 OTHER TOOLS.....	25
5.1 HEX CONVERTER.....	25
5.2 HEX EDITOR.....	26
5.3 TAIL FILE.....	27
5.4 TEXT EDITOR.....	28
5.5 VISUAL DIFFERENCE.....	30
5.6 XML EDITOR.....	32
6 MISCELLANEOUS.....	33
6.1 LANGUAGE.....	33
6.2 RESET VIEW.....	34
6.2.1 Windows Command Line.....	34
6.2.2 Linux and macOS Command Line.....	34

7 APPENDIX A – FREQUENTLY ASKED QUESTIONS (FAQ)	35
8 APPENDIX B – MQME-GUI UPGRADE PROCEDURES	36
8.1 WINDOWS UPGRADE.....	36
8.2 LINUX OR MACOS UPGRADE.....	36
9 APPENDIX C – SUPPORT	37
10 APPENDIX D – SUMMARY OF CHANGES	38
11 APPENDIX E – LICENSE AGREEMENT	40
12 APPENDIX F – NOTICES	42

1 Introduction

1.1 Overview

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

MQ Message Encryption (MQME) provides encryption for MQ message data while it resides in a queue or topic and in the MQ logs (i.e. all data at rest). In cryptography, encryption is the process of transforming information into an unreadable form (encrypted data). Decryption is the reverse process. It makes the encrypted information readable again. Only those with the key (PassPhrase) can successfully decrypt the encrypted data. MQME uses Advanced Encryption Standard (AES) to encrypt the data. AES is a data encryption scheme, adopted by the US government, that uses three different key sizes (128-bit, 192-bit, and 256-bit).

One of the features that MQME offers is the ability to control who accesses protected queues/topics. This control is obtained through the use of UserId grouping. MQME can query the local OS group or a group file. Group files are implemented in a similar manner to the way they are implemented in Unix and Linux (i.e. **/etc/group** file). Normally, the 'mqm', 'QMQM' or 'MUSR_MQADMIN' MQ UserIds or any UserId in the 'mqm' group get full access to all messages in all queues/topics. For queues protected by MQME, those privileged UserIds do **not** get access to the messages in the protected queues/topics unless they are explicitly added to the authorized list of users or groups.

Another feature of MQME is its ability to generate and validate the message via a digital signature. MQME uses the SHA-2 to create a cryptographic hash function (digital signature) for the message data. The digital signature provides verification that the message data has not been altered.

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

2 Installation

This chapter will describe the installation and configuration (optional) of MQME-GUI. For Windows users, the installation of MQME-GUI is very simple and straightforward. Other Operating Systems will need to unzip the downloaded software package.

2.1 Prerequisites

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

2.2 Software

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

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

2.3 Hardware / OS

MQME-GUI is capable of running on the following hardware platforms that support Java v1.6 (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 MQME-GUI application.

2.4.1 For Windows Users:

- Run the install program called: **mqmegui-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 MQME-GUI, there will be a newly created folder under **Start -> All Programs** called **MQME-GUI**.
- To run **MQME-GUI**, select the **MQME-GUI** icon from within the **MQME-GUI** folder.

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

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

```
unzip mqmegui.zip
```

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

```
Capitalware/MQME-GUI
```

- Change directory by typing the following:

```
cd Capitalware/MQME-GUI
```

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

```
chmod +x mqmegui.sh
```

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

```
./mqmegui.sh
```

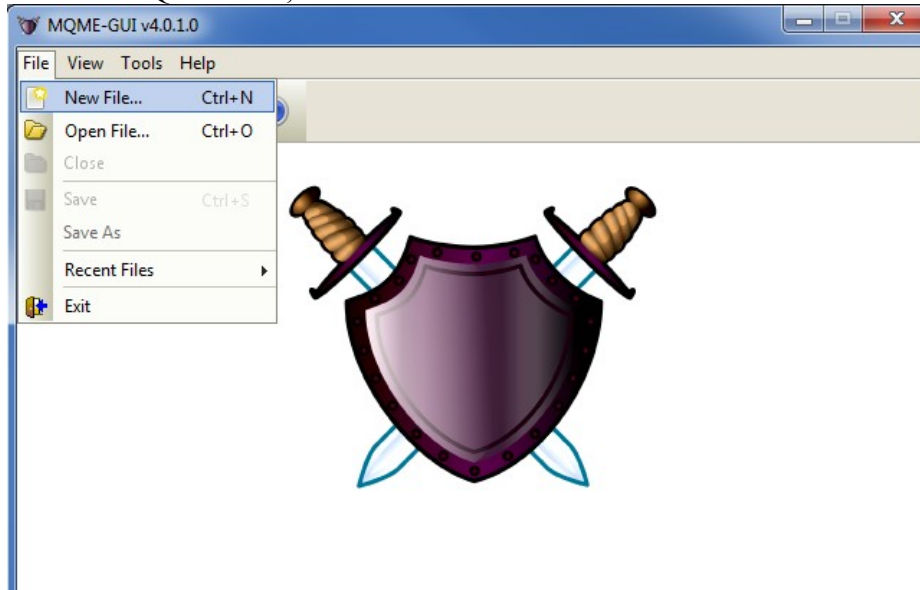
3 Creating / Updating IniFiles

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

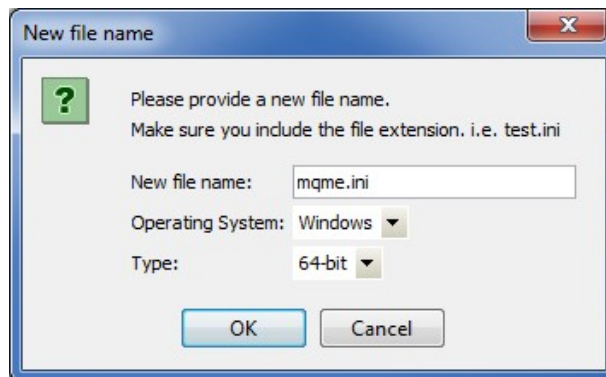
3.1 New IniFile

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

From the main menu of MQME-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 MQME IniFile.

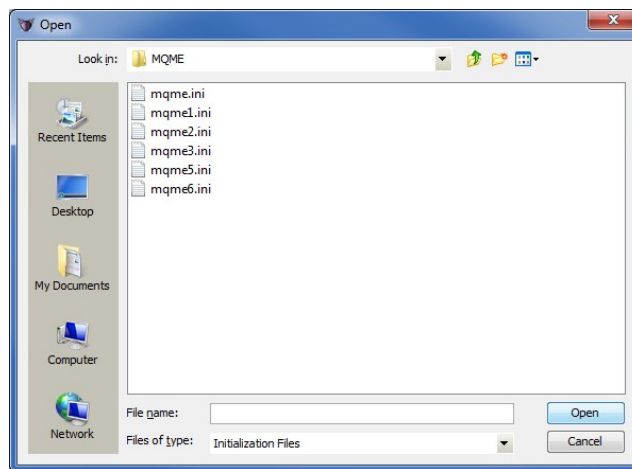
3.2 Open an IniFile

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

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



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

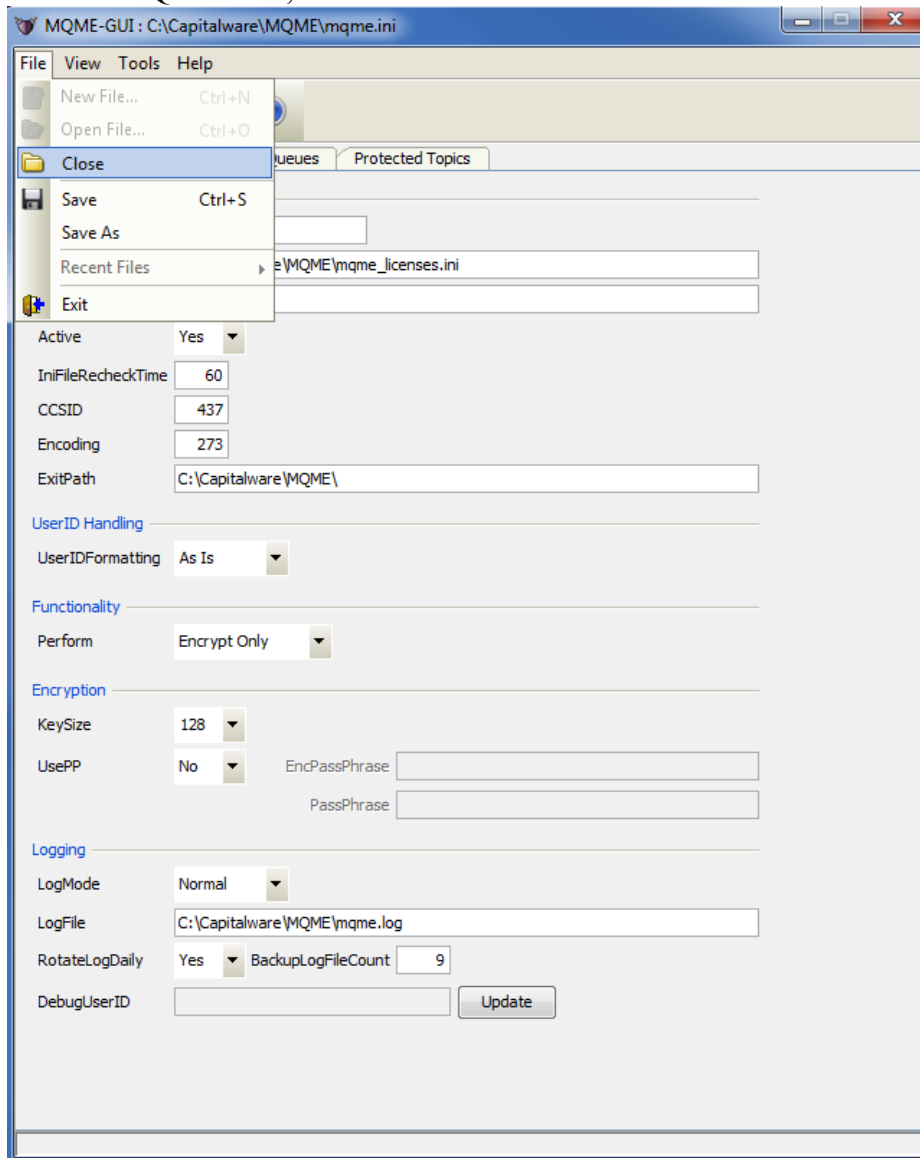


Select the IniFile that you wish to view / edit and then press the **Open** button.

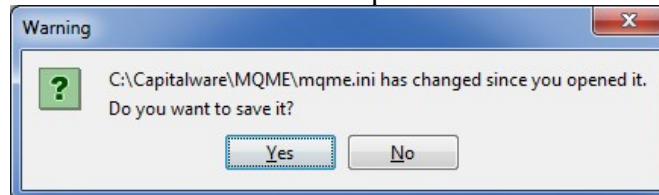
3.3 Close an IniFile

This section will describe how to close a MQME IniFile.

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



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



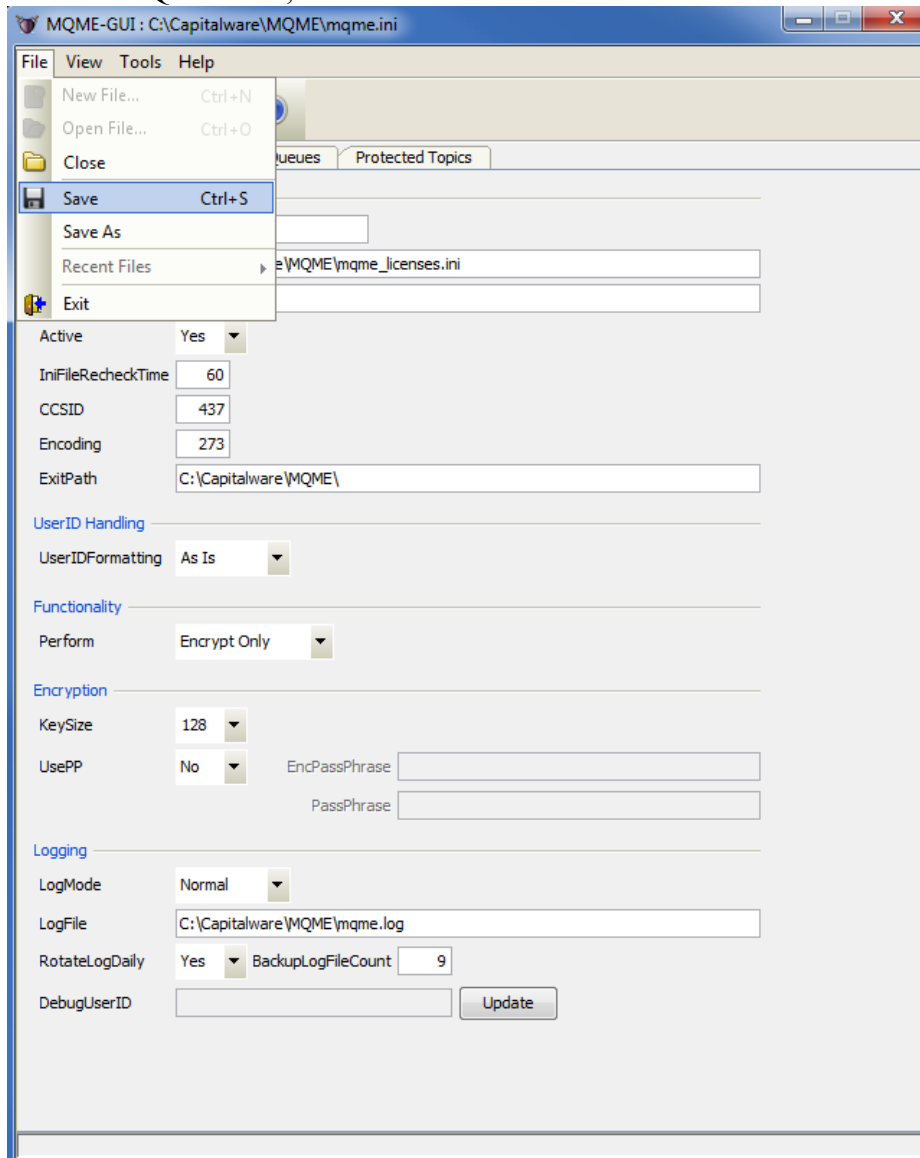
Select **Yes** to save the changes otherwise select **No**.

The MQME-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 MQME IniFile.

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



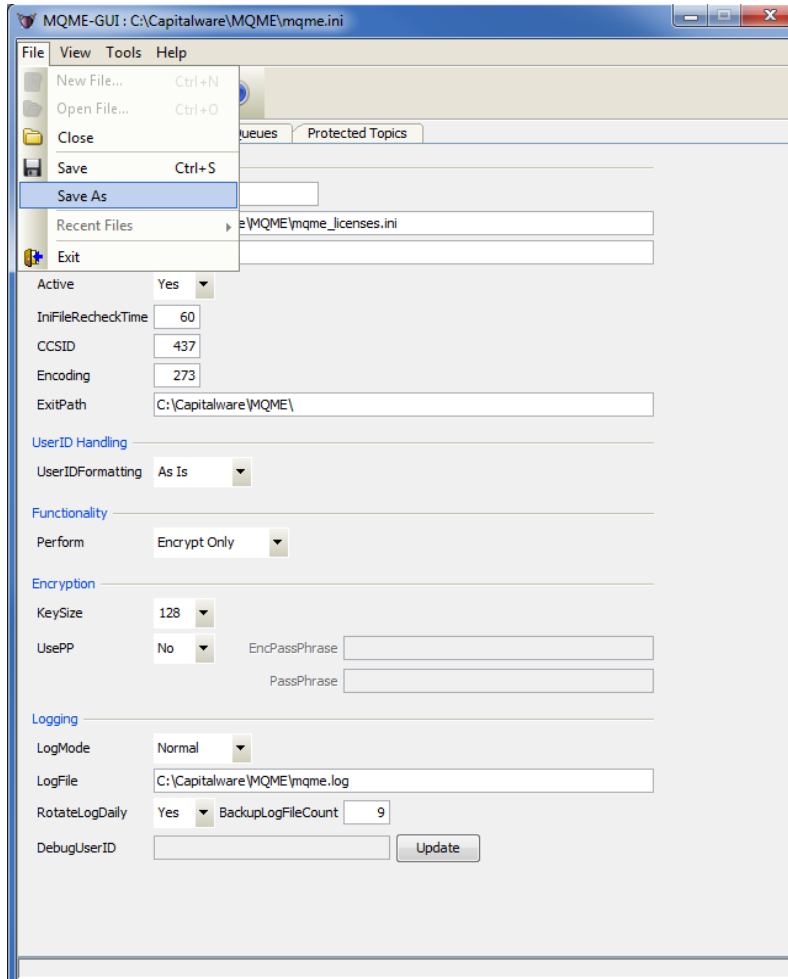
The parameter information will be saved to the IniFile.

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

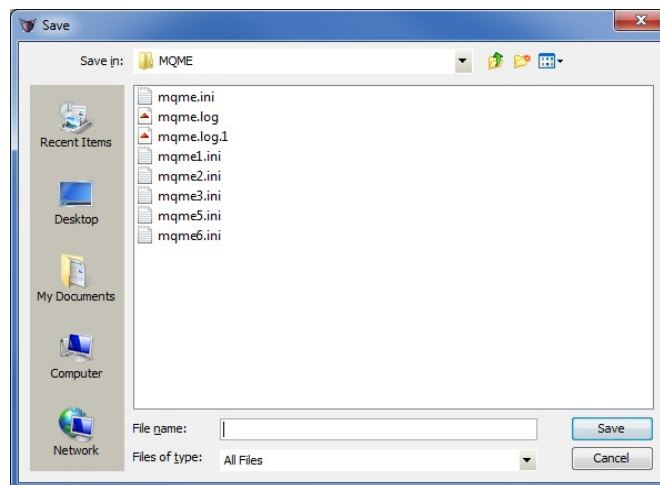
3.5 Save As IniFile

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

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



Select or Input the IniFile name that you wish use then press the **Save** button.

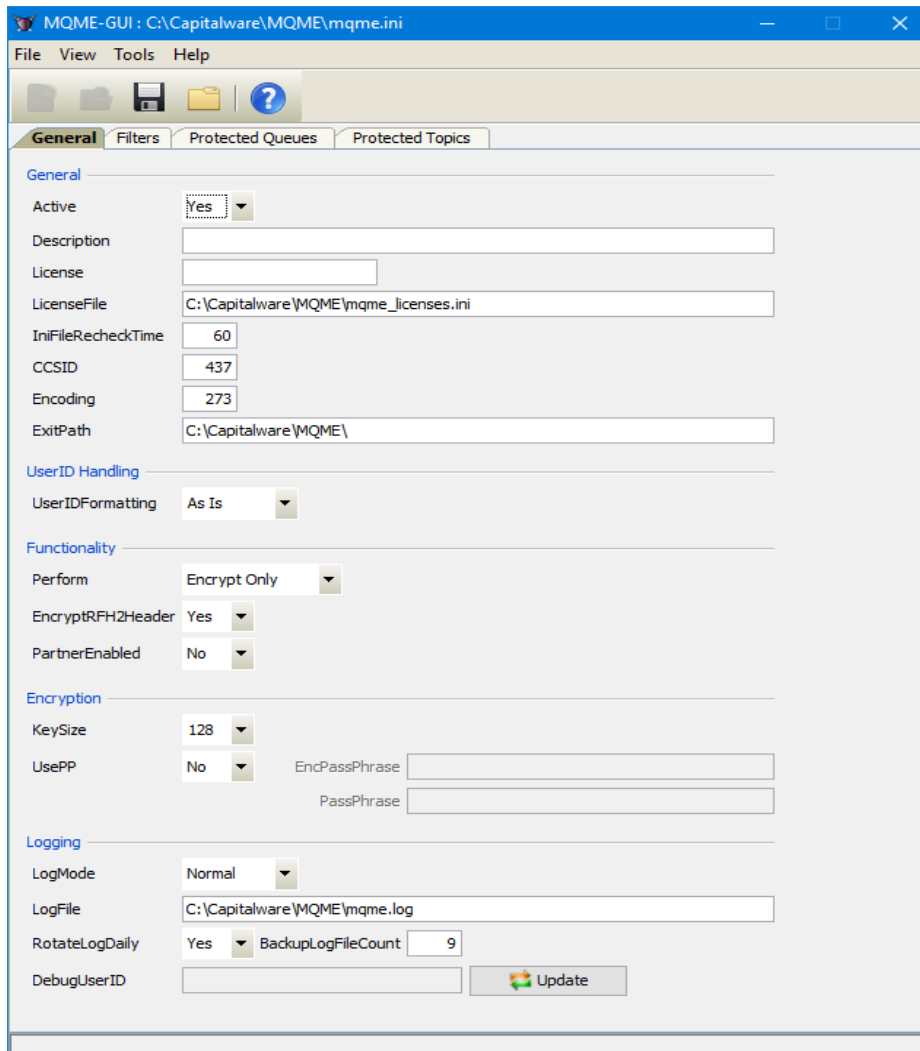


4 Tabs

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 MQME 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 MQME.
- **Description** parameter is optional and is not used by MQME. 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\MQME\mqme_licenses.ini**
 - For IBM MQ 32-bit on Unix & Linux: **/var/mqm/exits/mqme_licenses.ini**
 - For IBM MQ 64-bit on Unix and Linux: **/var/mqm/exits64/mqme_licenses.ini**
 - For IBM i: **/QIBM/UserData/mqm/mqme/mqme_licenses.ini**
- **CCSID** enables monitoring of internal MQ applications issuing MQ API calls.
- **Encoding** specifies what type of monitoring the user wishes to have.
- **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 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.

4.1.3 Functionality

This section describes the keywords related to functionality.

- **Perform** indicates what functionality that MQME will perform. Perform supports 3 values [S / E / B]. The default value is E.
 - **S** means that MQME will only sign the message
 - **E** means that MQME will only encrypt the message
 - **B** means that MQME will sign and encrypt the message

When signing the message, MQME creates the digital signature using cryptographic hash function of SHA-2.

- **EncryptRFH2Header** specifies that the MQRFH2 header fields and folders are to be encrypted when the message payload is encrypted. EncryptRFH2Header supports 2 values [Y / N]. The default value is Y.
- **PartnerEnabled** indicates that MQ Channel Encryption (MQCE) is used on the channels and that the message payload should not be decrypted. PartnerEnabled supports 2 values [Y / N]. The default value is N.

4.1.4 Encryption

This section describes the keywords for the global encryption / decryption settings.

- **KeySize** specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256.
- **UsePP** allows the use of a user specified PassPhrase
- **EncPassPhrase** specifies a global encrypted PassPhrase for this IniFile that will be used for the protected queues/topics.
- **PassPhrase** specifies the actual 16, 24 or 32-byte PassPhrase that will be used for the message encryption and / or decryption.

4.1.5 Logging

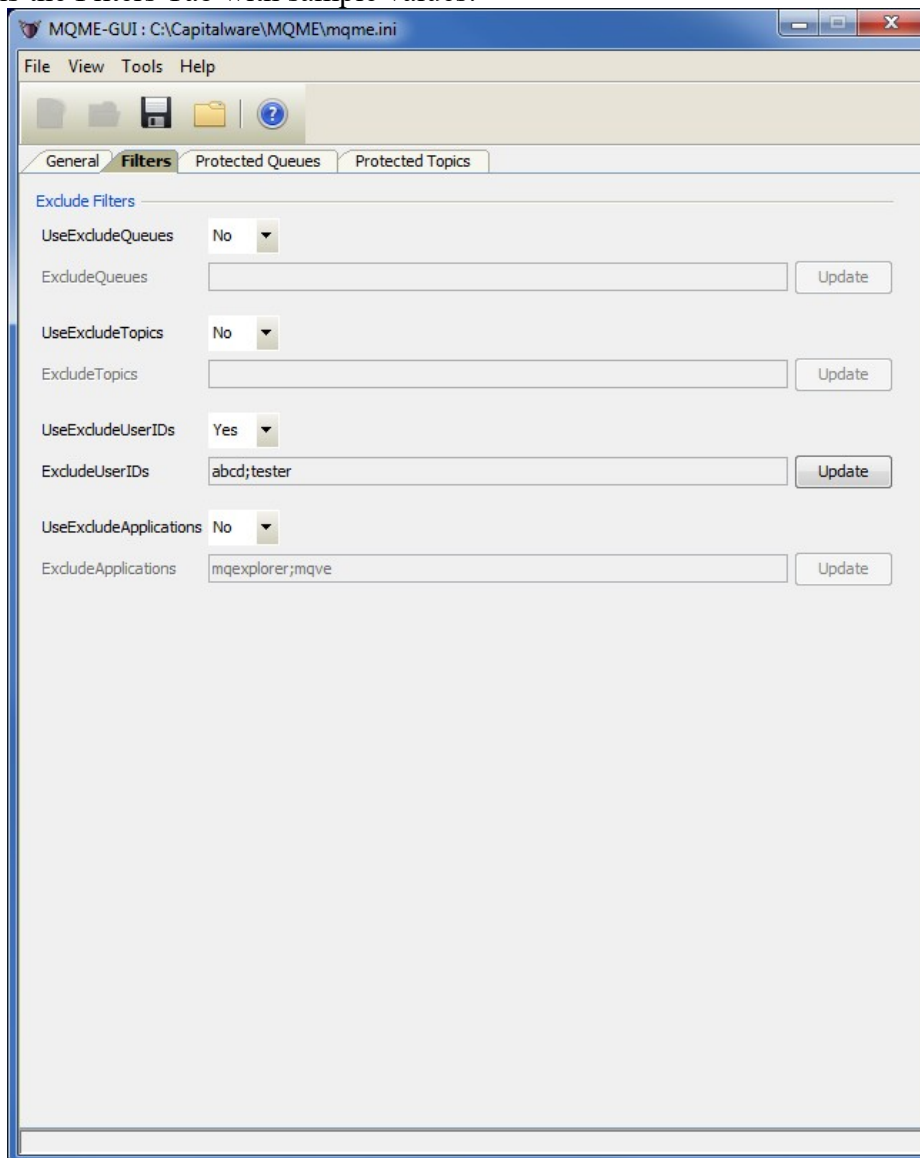
This section describes the keywords related to logging and their values.

- **LogMode** specifies what type of logging the user wishes to have. LogMode supports 4 values: Quiet, Normal, Verbose and Debug. The default value is Normal.
- **LogFile** specifies the location of the log file. The default is as follows:
 - For Windows: C:\Capitalware\MQME\mqme.log
 - For IBM MQ 32-bit on Unix and Linux: /var/mqm/exits/mqme.log
 - For IBM MQ 64-bit on Unix and Linux: /var/mqm/exits64/mqme.log
 - For IBM I: /QIBM/UserData/mqm/mqme/mqme.log
- **RotateLogDaily** specifies whether or not MQME when auto-rotate the log files. The default value is Yes.
- **BackupLogFileCount** specifies the number of backup log files to be kept / retained. The default value is 9.

4.2 Filters Tab

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

The following is the Filters Tab with sample values:



The following are the IniFile parameters on the Filters Tab:

4.2.1 Exclude Filters

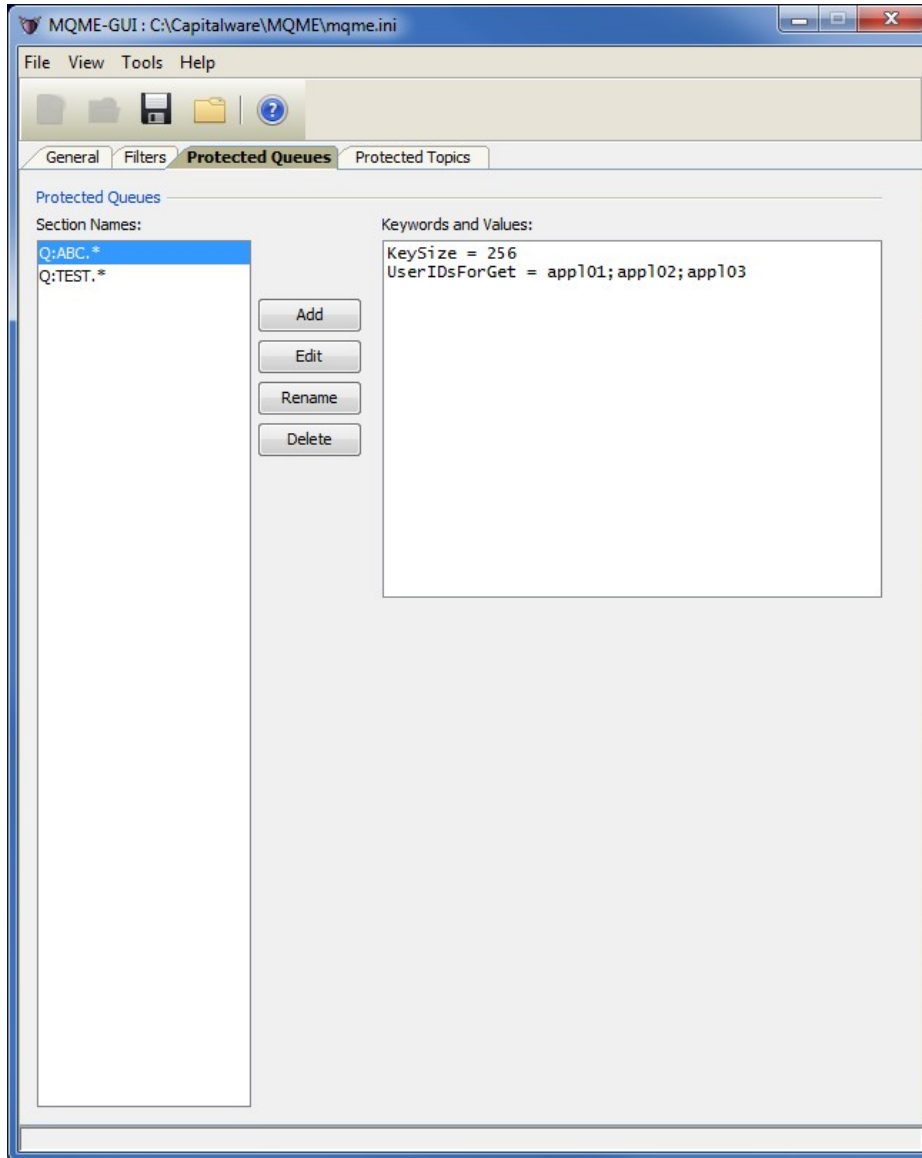
This section describes the Exclude Filter keywords and their values.

- **UseExcludesApplications** allows the user to specifically exclude applications from the encryption / decryption processing.
- **ExcludesApplications** explicitly specifies which applications should not be protected by MQME.
- **UseExcludesQueues** allows the user to specifically exclude queues from the encryption / decryption processing.
- **ExcludesQueues** explicitly specifies which queues should not be protected by MQME.
- **UseExcludesTopics** allows the user to specifically exclude topics from the encryption / decryption processing.
- **ExcludesTopics** explicitly specifies which topics should not be protected by MQME.
- **UseExcludesUserIDs** allows the user to specifically exclude UserIds from the encryption / decryption processing.
- **ExcludesUserIDs** explicitly specifies which UserIds should not be protected by MQME.

4.3 Protected Queues Tab

This section will describe various parameters on the Protected Queues tab of a MQME IniFile.

The following is the Protected Queues Tab with sample values:

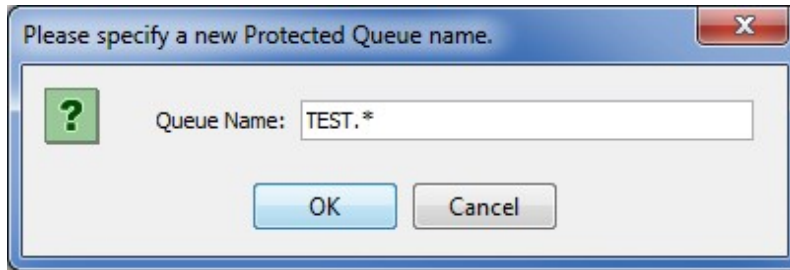


This section describes IniFile keywords that the user will use so that MQME will perform encryption / decryption processing against.

4.3.1 Managing Protected Queues

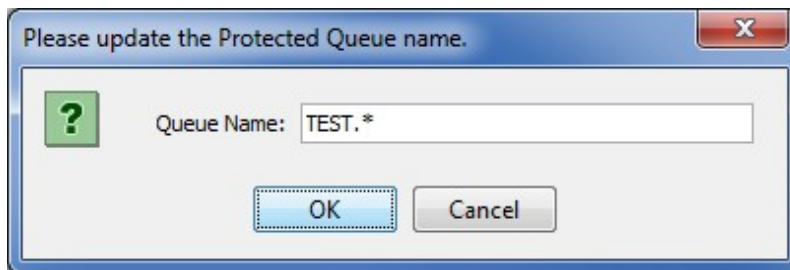
This section describes the necessary steps to manage the protected queue:

Click the **Add** button to add an entry to the list.



Input the Protect Queue name including wildcards and then click the **Ok** button.

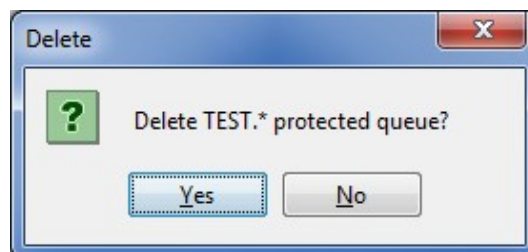
Click the **Rename** button to delete an entry in the list.



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

Click the **Delete** button to delete an entry in the list.

Click the **Yes** button to permanently remove the entry from the list.



Click the **Edit** button to update an entry in the list.

The screenshot shows the 'Edit Protected Queue: TEST.*' dialog box. It is organized into several sections:

- Authorized UserIds for Reading:** Includes 'UserIDsForGet' (text: roger), 'UseGroupsForGet' (Yes), 'GroupsForGet' (grp1;grp2), and 'UseGroupFileForGet' (Yes) with 'GroupFileForGet' (C:\Capitalware\MQME\groupsForGet.ini). Each has an 'Update' button.
- Authorized UserIds for Writing:** Includes 'UserIDsForPut' (Text: Test1;Test2), 'UseGroupsForPut' (Yes), 'GroupsForPut' (GrpA;GrpB), and 'UseGroupFileForPut' (Yes) with 'GroupFileForPut' (C:\Capitalware\MQME\groupsForPut.ini). Each has an 'Update' button.
- Functionality:** Includes 'Perform' (Encrypt Only), 'EncryptRFH2Headers' (Yes), and 'PartnerEnabled' (No).
- Encryption:** Includes 'KeySize' (256), 'UsePP' (No), 'EncPassPhrase', and 'PassPhrase' text boxes.
- Authorized Application Names for Reading:** Includes 'UseApplicationsForGet' (Yes) and 'ApplicationsForGet' (ag1) with an 'Update' button.
- Authorized Application Names for Writing:** Includes 'UseApplicationsForPut' (Yes) and 'ApplicationsForPut' (ap1) with an 'Update' button.

At the bottom right, there are 'Save' and 'Cancel' buttons.

4.3.1.1 Authorized UserIds for Reading

This section describes how to authorize UserIds or a UserId against a group, so that MQME will decrypt the message for the user.

- **UserIDsForGet** specifies the UserIds (wildcards are valid) which are authorized so that MQME will decrypt the message for the user. Separate each UserId with a semi-colon ';'.
- **UseGroupsForGet** keyword controls the use of the group file. Set to 'Y' to allow authorization via the local OS groups or a group file.
- **GroupsForGet** keyword specifies the authorized groups.
- **UseGroupFileForGet** keyword controls the use of the group file. Set to 'Y' to allow authorization via the group file.
- **GroupFileForGet** specifies the location of the group file.

4.3.1.2 Authorized UserIds for Writing

This section describes how to authorize UserIds or a UserId against a group, to write (put) to a protected queue.

- **UserIDsForPut** specifies the UserIds (wildcards are valid) that are authorized to write (put) messages to the queue. Separate each UserId with a semi-colon ';'.
- **UseGroupsForPut** keyword controls the use of the group file. Set to 'Y' to allow authorization via the local OS groups or a group file.
- **GroupsForPut** keyword specifies the authorized groups.
- **UseGroupFileForPut** keyword controls the use of the group file. Set to 'Y' to allow authorization via the group file.
- **GroupFileForPut** specifies the location of the group file.

4.3.1.3 Functionality

This section describes the keywords related to functionality.

- **Perform** indicates what functionality that MQME will perform. Perform supports 3 values [S / E / B]. The default value is E.
 - **S** means that MQME will only sign the message
 - **E** means that MQME will only encrypt the message
 - **B** means that MQME will sign and encrypt the message

When signing the message, MQME creates the digital signature using cryptographic hash function of SHA-2.

- **EncryptRFH2Header** specifies that the MQRFH2 header fields and folders are to be encrypted when the message payload is encrypted. EncryptRFH2Header supports 2 values [Y / N]. The default value is Y.
- **PartnerEnabled** indicates that MQ Channel Encryption (MQCE) is used on the channels and that the message payload should not be decrypted. PartnerEnabled supports 2 values [Y / N]. The default value is N.

4.3.1.4 Encryption

This section describes the keywords for the encryption / decryption settings.

- **KeySize** specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256.
- **UsePP** allows the use of a user specified PassPhrase
- **EncPassPhrase** specifies an encrypted PassPhrase that will be used for the protected queue.
- **PassPhrase** specifies the actual 16, 24 or 32-byte PassPhrase that will be used for the message encryption and / or decryption.

4.3.1.5 Authorized Application Name for Reading

This section describes how to add a secondary level of authorization. MQME will first check the UserId and then application name and if both are authorized then MQME will decrypt the message data when the message data is retrieved from a queue.

- **UseApplicationsForGet** allows the use of authorization by application name
- **ApplicationsForGet** specifies a list of application names to be checked

4.3.1.6 Authorized Application Name for Writing

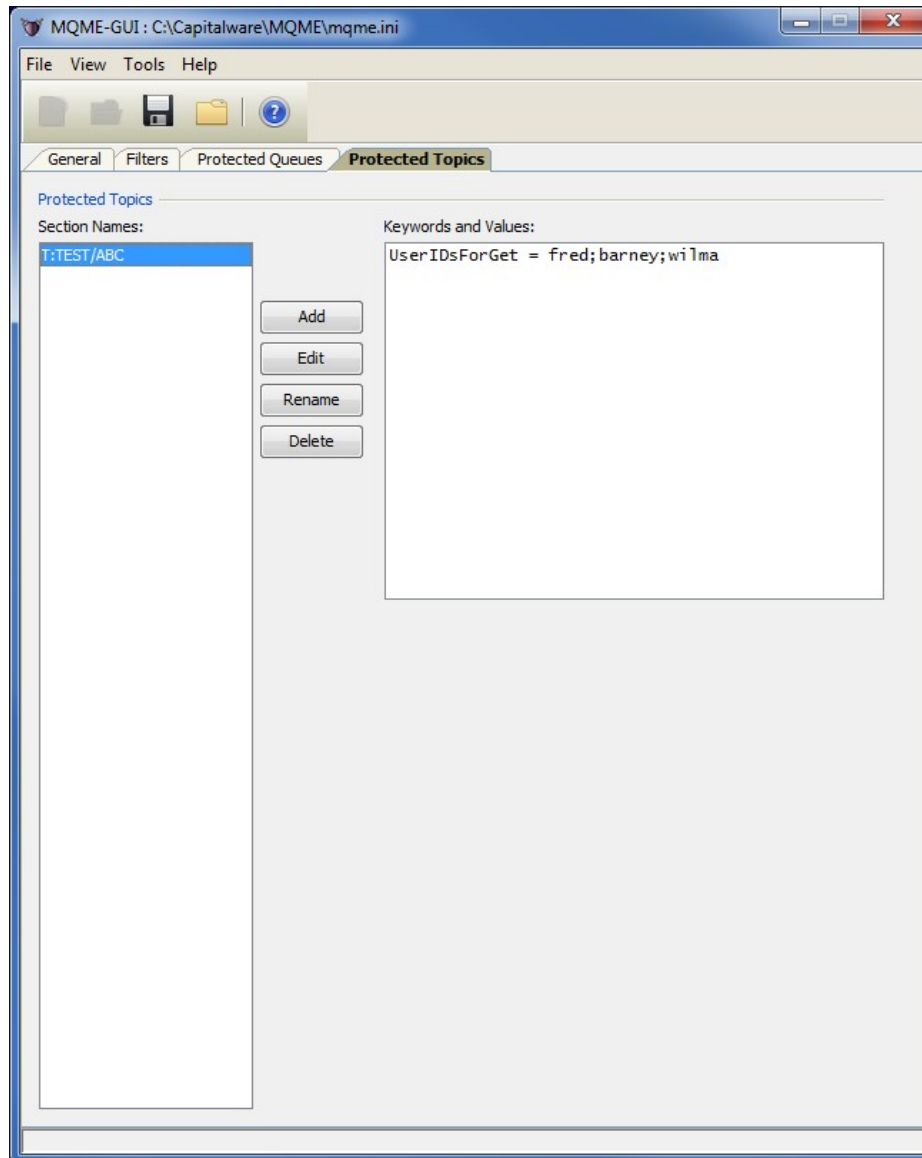
This section describes how to add a secondary level of authorization. MQME will first check the UserId and then application name and if both are authorized then MQME will encrypt the message data when the message data is written to a queue.

- **UseApplicationsForPut** allows the use of authorization by application name
- **ApplicationsForPut** specifies a list of application names to be checked

4.4 Protected Topics Tab

This section will describe various parameters on the Protected Topics tab of a MQME IniFile.

The following is the Protected Topics Tab with sample values:

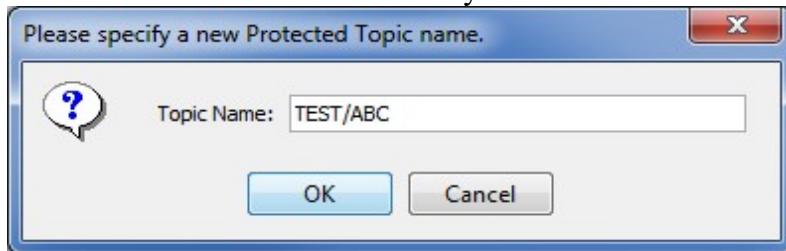


This section describes IniFile keywords that the user will use so that MQME will perform encryption / decryption processing against.

4.4.1 Managing Protected Topics

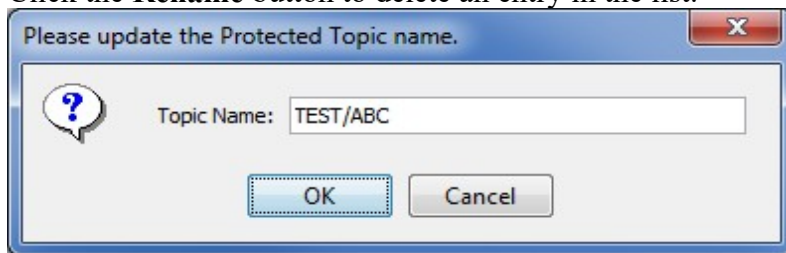
This section describes the necessary steps to manage the protected topic:

Click the **Add** button to add an entry to the list.



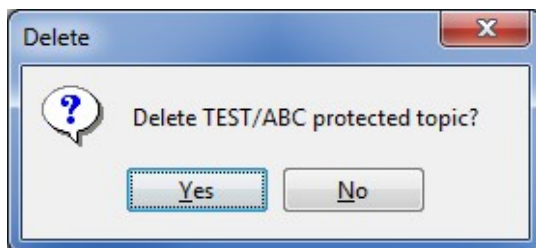
Input the Protect Topic name including wildcards and then click the **Ok** button.

Click the **Rename** button to delete an entry in the list.



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

Click the **Delete** button to delete an entry in the list.



Click the **Yes** button to permanently remove the entry from the list.

Click the **Edit** button to update an entry in the list.

The screenshot shows a dialog box titled "Edit Protected Topic: TEST/ABC". It contains several sections for configuring permissions and functionality:

- Authorized UserIds for Reading:** Includes a text field for "UserIDsForGet" (containing "fred;barney;wilma"), a "UseGroupsForGet" dropdown (set to "No"), a "GroupsForGet" text field, and a "UseGroupFileForGet" dropdown (set to "No"). There is a "GroupFileForGet" text field and an "Update" button.
- Authorized UserIds for Writing:** Includes a text field for "UserIDsForPut" (containing "*"), a "UseGroupsForPut" dropdown (set to "No"), a "GroupsForPut" text field, and a "UseGroupFileForPut" dropdown (set to "No"). There is a "GroupFileForPut" text field and an "Update" button.
- Functionality:** Includes a "Perform" dropdown (set to "Encrypt Only"), an "EncryptRFH2Headers" dropdown (set to "Yes"), and a "PartnerEnabled" dropdown (set to "No").
- Encryption:** Includes a "KeySize" dropdown (set to "128"), a "UsePP" dropdown (set to "No"), an "EncPassPhrase" text field, and a "PassPhrase" text field.
- Authorized Application Names for Reading:** Includes a "UseApplicationsForGet" dropdown (set to "No") and an "ApplicationsForGet" text field, with an "Update" button.
- Authorized Application Names for Writing:** Includes a "UseApplicationsForPut" dropdown (set to "No") and an "ApplicationsForPut" text field, with an "Update" button.

At the bottom of the dialog, there are "Save" and "Cancel" buttons.

4.4.1.1 Authorized UserIds for Reading

This section describes how to authorize UserIds or a UserId against a group, so that MQME will decrypt the message for the user.

- **UserIDsForGet** specifies the UserIds (wildcards are valid) which are authorized so that MQME will decrypt the message for the user. Separate each UserId with a semi-colon ';'.
- **UseGroupsForGet** keyword controls the use of the group file. Set to 'Y' to allow authorization via the local OS groups or a group file.
- **GroupsForGet** keyword specifies the authorized groups.
- **UseGroupFileForGet** keyword controls the use of the group file. Set to 'Y' to allow authorization via the group file.
- **GroupFileForGet** specifies the location of the group file.

4.4.1.2 Authorized UserIds for Writing

This section describes how to authorize UserIds or a UserId against a group, to write (put) to a protected topic.

- **UserIDsForPut** specifies the UserIds (wildcards are valid) that are authorized to write (put) messages to the topic. Separate each UserId with a semi-colon ';'.
- **UseGroupsForPut** keyword controls the use of the group file. Set to 'Y' to allow authorization via the local OS groups or a group file.
- **GroupsForPut** keyword specifies the authorized groups.
- **UseGroupFileForPut** keyword controls the use of the group file. Set to 'Y' to allow authorization via the group file.
- **GroupFileForPut** specifies the location of the group file.

4.4.1.3 Functionality

This section describes the keywords related to functionality.

- **Perform** indicates what functionality that MQME will perform. Perform supports 3 values [S / E / B]. The default value is E.
 - **S** means that MQME will only sign the message
 - **E** means that MQME will only encrypt the message
 - **B** means that MQME will sign and encrypt the message

When signing the message, MQME creates the digital signature using cryptographic hash function of SHA-2.

- **EncryptRFH2Header** specifies that the MQRFH2 header fields and folders are to be encrypted when the message payload is encrypted. EncryptRFH2Header supports 2 values [Y / N]. The default value is Y.
- **PartnerEnabled** indicates that MQ Channel Encryption (MQCE) is used on the channels and that the message payload should not be decrypted. PartnerEnabled supports 2 values [Y / N]. The default value is N.

4.4.1.4 Encryption

This section describes the keywords for the encryption / decryption settings.

- **KeySize** specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256.
- **UsePP** allows the use of a user specified PassPhrase
- **EncPassPhrase** specifies an encrypted PassPhrase that will be used for the protected topic.
- **PassPhrase** specifies the actual 16, 24 or 32-byte PassPhrase that will be used for the message encryption and / or decryption.

4.4.1.5 Authorized Application Name for Reading

This section describes how to add a secondary level of authorization. MQME will first check the UserId and then application name and if both are authorized then MQME will decrypt the message data when the message data is retrieved from a topic.

- **UseApplicationsForGet** allows the use of authorization by application name
- **ApplicationsForGet** specifies a list of application names to be checked

4.4.1.6 Authorized Application Name for Writing

This section describes how to add a secondary level of authorization. MQME will first check the UserId and then application name and if both are authorized then MQME will encrypt the message data when the message data is written to a topic.

- **UseApplicationsForPut** allows the use of authorization by application name
- **ApplicationsForPut** specifies a list of application names to be checked

5 Other Tools

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

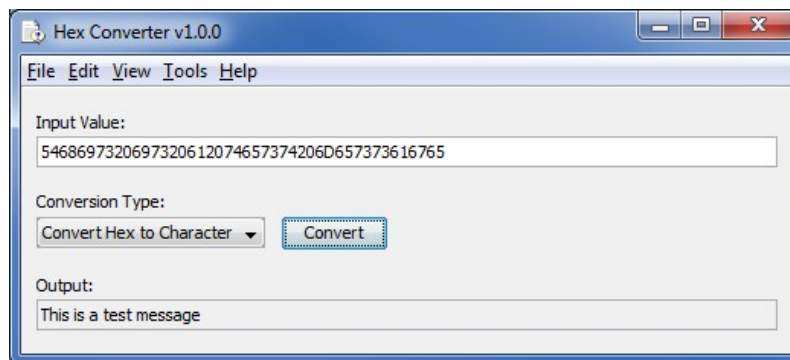
5.1 Hex Converter

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 MQME-GUI.

From the main panel of MQME-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

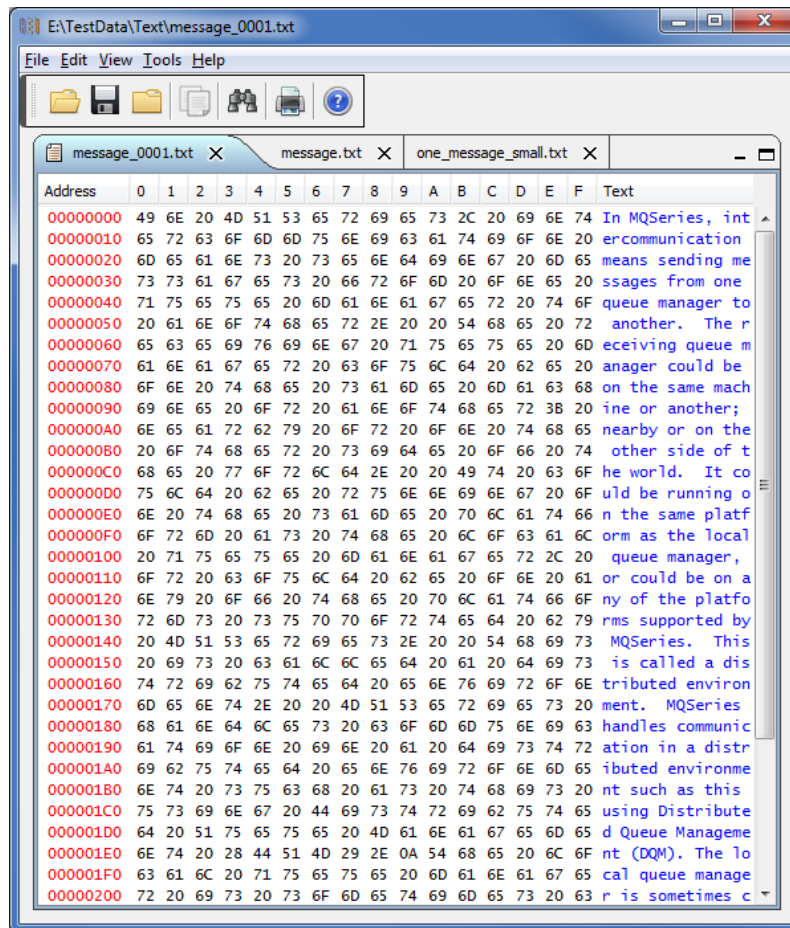
5.2 Hex Editor

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 MQME-GUI.

From the main panel of MQME-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

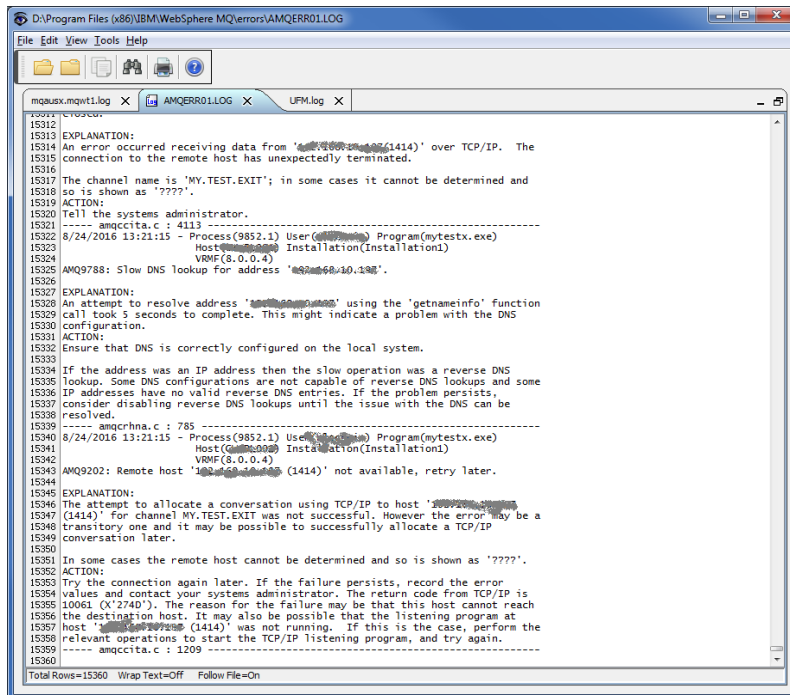
5.3 Tail File

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 MQME-GUI.

From the main panel of MQME-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

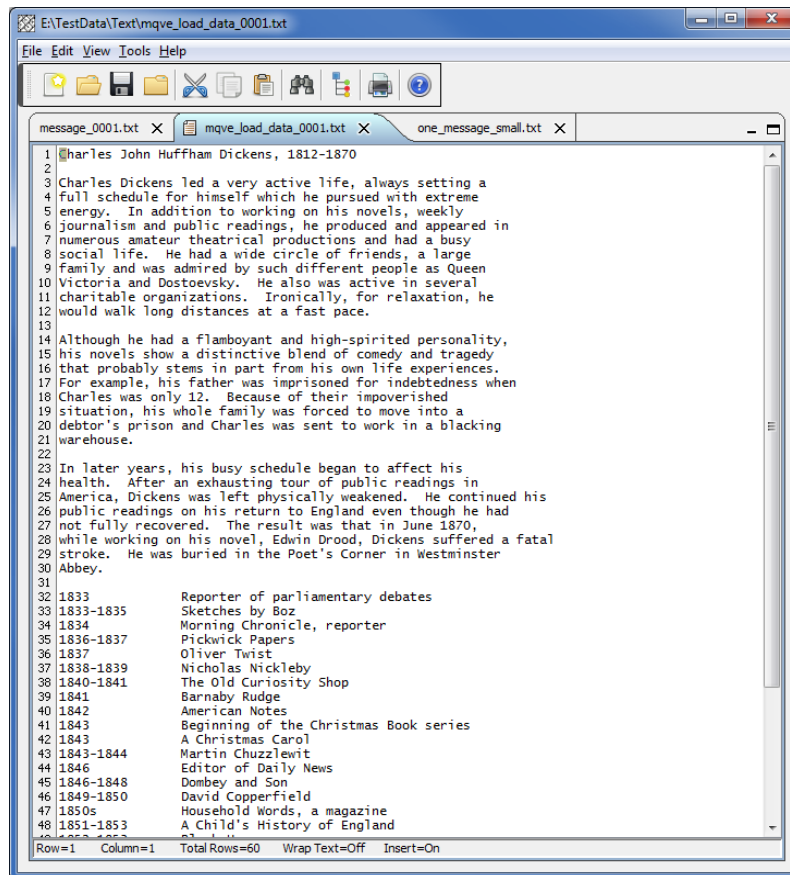
5.4 Text Editor

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 MQME-GUI.

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

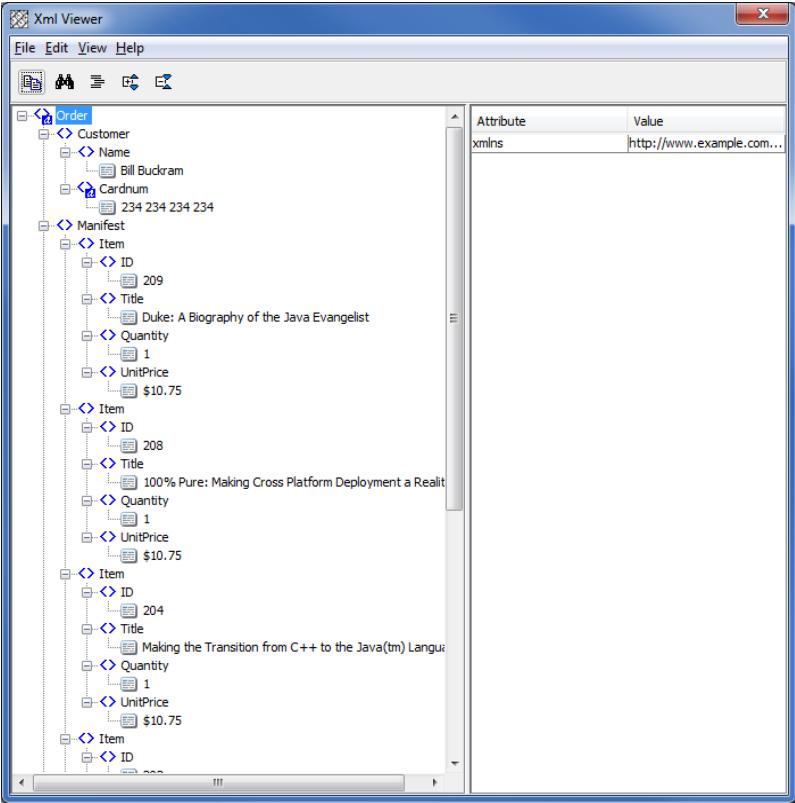
This is the main panel of the Text Editor and it has 3 open files.



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.



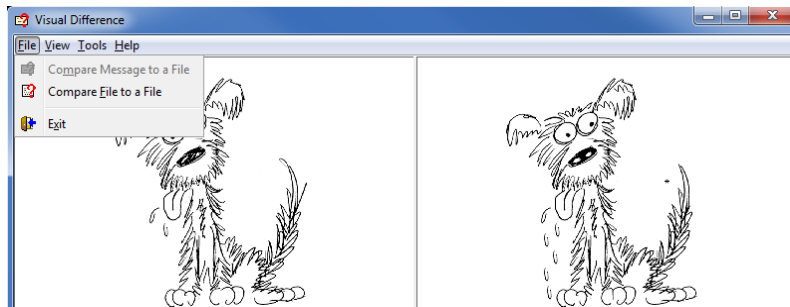
5.5 Visual Difference

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

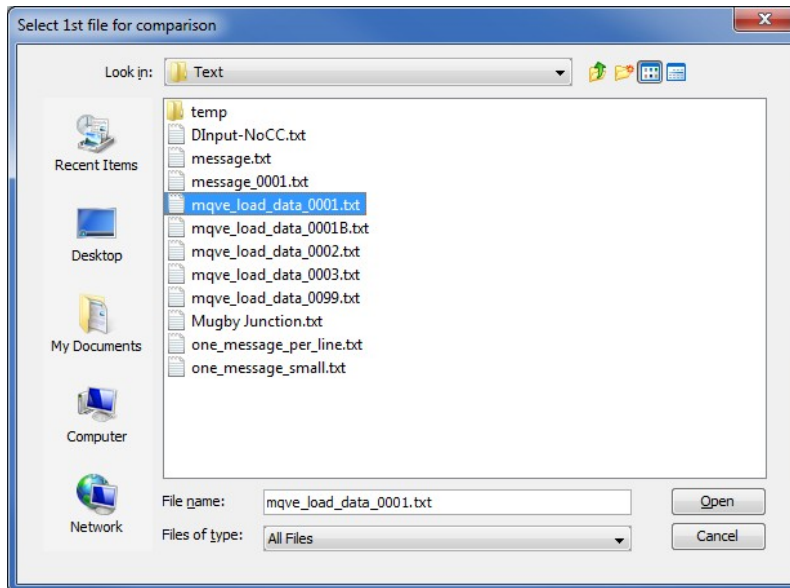
For the Other Tools, you can have unlimited number of them running (up to what your box can support). Also, while any of the Other Tools are running, the user can still use MQME-GUI.

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

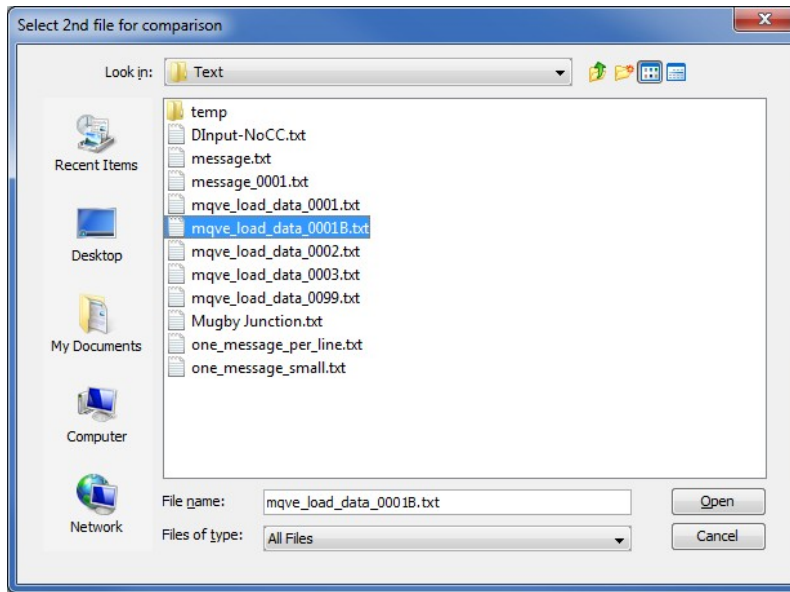
This is the main panel of the Visual Difference. To initiate the file comparison process, click File and then click Compare File to File



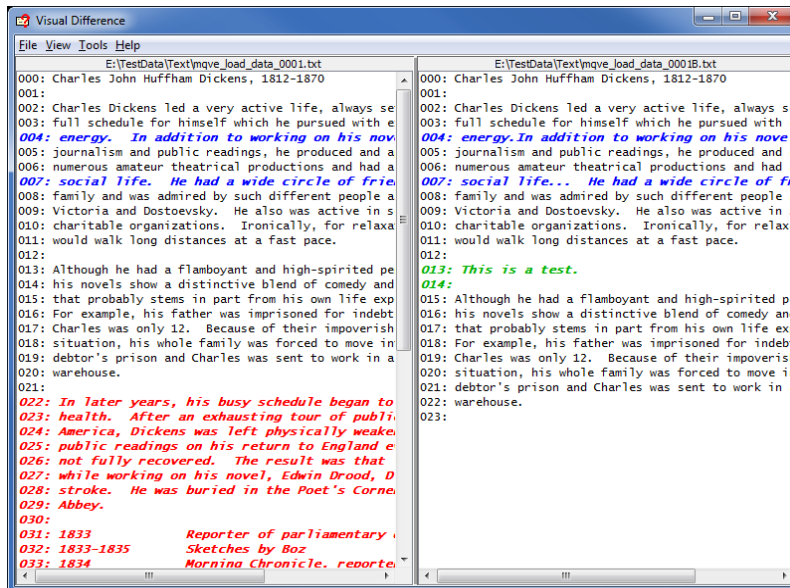
First the user needs to select the first file for the comparison.



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.



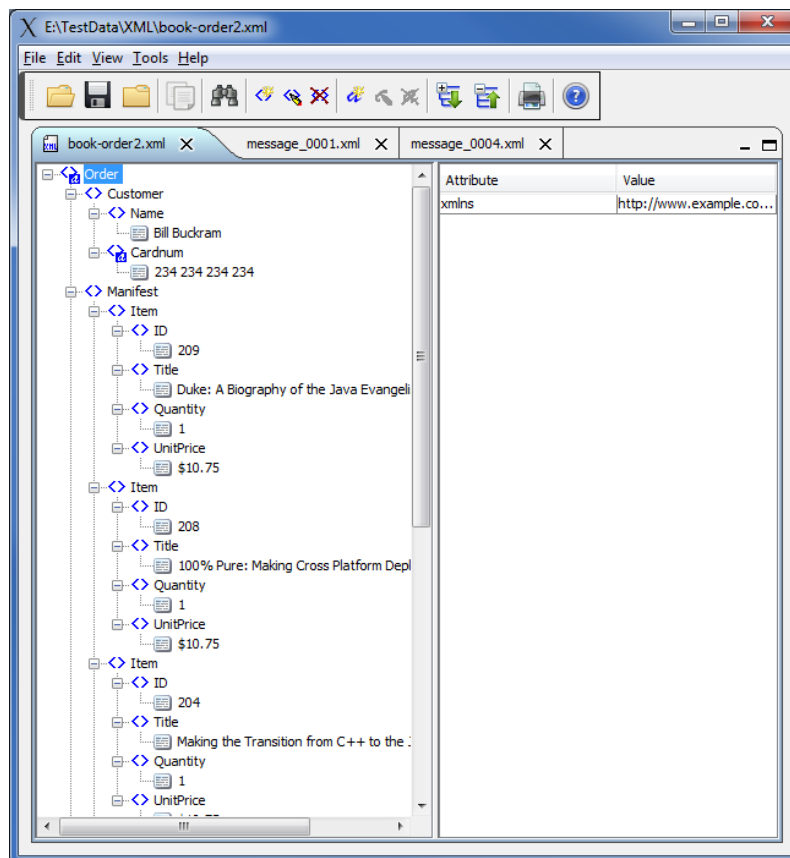
5.6 XML Editor

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 MQME-GUI.

From the main panel of MQME-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

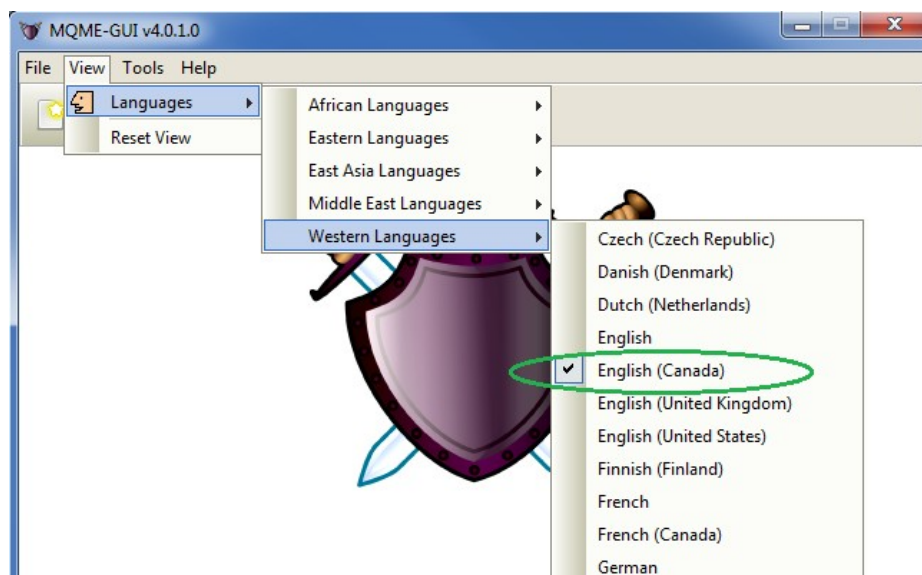
This chapter will describes how the miscellaneous features of MQME-GUI.

6.1 Language

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

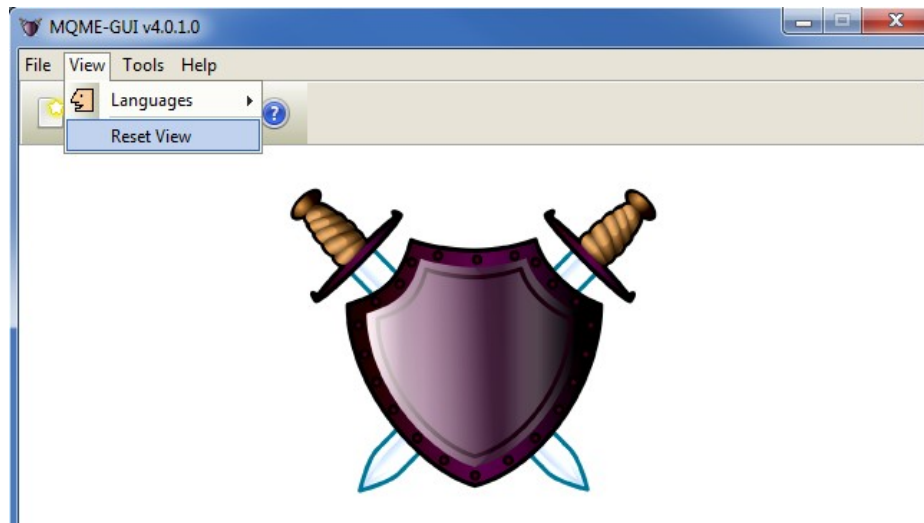
➤ Amharic (am)	➤ Hungarian (hu)	➤ Shona (sn)
➤ Arabic (ar)	➤ Igbo (ig)	➤ Sindhi (sd)
➤ Azerbaijani (az)	➤ Indonesian (in)	➤ Spanish (es)
➤ Bengali (bn)	➤ Italian (it)	➤ Sundanese (su)
➤ Cebuano (ceb)	➤ Japanese (ja)	➤ Swahili (sw)
➤ Chinese (China) (zh)	➤ Javanese (jv)	➤ Swedish (sv)
➤ Chinese (Taiwan) (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 MQME-GUI, select **View**, then **Language**.



6.2 Reset View

This section will describe how to reset the view of MQME-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 MQME-GUI:

```
mqmegui.exe RG
```

6.2.2 Linux and macOS Command Line

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

```
./mqmegui.sh RG
```

7 Appendix A – Frequently Asked Questions (FAQ)

Q. Can IniFiles be create on Windows for MQME 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 – MQME-GUI Upgrade Procedures

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

8.1 Windows Upgrade

- Stop all instances of MQME-GUI
- Backup any MQME-GUI data files in the MQME-GUI install directory
- Go to the Windows **Control Panel** and select **Program and Features**
- Select MQME-GUI and then click the **Uninstall** button
- Run the install program called: **mqmegui-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 **MQME-GUI**.
- Restore the MQME-GUI data files if necessary
- To run MQME-GUI, select the MQME-GUI icon from within the MQME-GUI folder.

8.2 Linux or macOS Upgrade

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

```
unzip mqmegui.zip
```

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

```
Capitalware/MQME-GUI
```

- Change directory by typing the following:

```
cd Capitalware/MQME-GUI
```

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

```
chmod +x mqmegui.sh
```

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

```
./mqmegui.sh
```


9 Appendix C – Support

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

By email at:

support@capitalware.com

By regular mail at:

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

10 Appendix D – Summary of Changes

- MQME-GUI v4.2.0
 - Added keyword EncryptRFH2Header to handle encrypting the MQRFH2 header or not
- MQME-GUI v4.1.0
 - 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
- MQME-GUI v4.0.0
 - Added Topic section so that MQME will protect topics.
 - Added EncryptPassPhrase keyword to support the use of encrypted PassPasshrase.
 - Added UseExcludeTopics and ExcludeTopics keywords to explicitly exclude topics from being protected.
 - Removed MQAPILevel keyword as it is no longer needed.
- MQME-GUI v3.3.0
 - Added the ability to exclude applications - new keywords: UseExcludeApplications & ExcludeApplications
 - Added the ability to exclude UserIds - new keywords: UseExcludeUserIDs & ExcludeUserIDs
- MQME-GUI v3.1.0
 - Added keyword IniFileRecheckTime to only check the IniFile modification time after 'x' seconds
 - Added keyword DebugUserID to selectively enable debugging by UserID
- MQME-GUI v3.0.1
 - Added new keywords for UserId formatting: UserIDFormatting
- MQME-GUI v3.0.0
 - Removed keywords: UserIDs, UseGroups, Groups, GroupFile, UseApplications and Applications

- Added new keywords for read authority: UserIDsForGet, UseGroupsForGet, GroupsForGet, UseGroupFileForGet, GroupFileForGet, UseApplicationsForGet and ApplicationsForGet
 - Added new keywords for write authority: UserIDsForPut, UseGroupsForPut, GroupsForPut, UseGroupFileForPut, GroupFileForPut, UseApplicationsForPut and ApplicationsForPut
 - Updated the GUI
- MQME-GUI v2.0.0
- Added new keyword UserIDsForPut which restricts who can put messages to a queue
 - Updated the GUI
- MQME-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 mqme.ini, mqme_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.