

# ***MQ Channel Encryption Installation and Operation Manual***



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

Last Updated: January 2021.  
© Copyright Capitalware Inc. 2010, 2021.

# Table of Contents

<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 OVERVIEW.....	1
1.2 EXECUTIVE SUMMARY.....	2
1.3 MESSAGE DIAGRAM (LOGICAL VIEW).....	2
1.4 CONTEXT DIAGRAM (LOGICAL VIEW).....	3
1.5 PREREQUISITES.....	4
1.5.1 Operating System.....	4
1.5.2 IBM MQ.....	5
1.5.3 Windows 32-bit.....	5
1.5.4 Windows 64-bit.....	5
<b>2 INSTALLING MQ CHANNEL ENCRYPTION.....</b>	<b>6</b>
2.1 CHANNEL EXIT.....	6
2.1.1 Windows Installation.....	7
2.1.2 Linux 32-bit Installation.....	7
2.1.3 Unix and Linux 64-bit Installation.....	8
2.1.4 IBM i Installation.....	9
2.1.5 IBM APAR IY91269.....	10
2.1.6 IBM APAR IZ87819.....	10
2.1.7 MQCE-GUI Installation.....	11
<b>3 CONFIGURING QMGR TO QMGR CHANNELS.....</b>	<b>12</b>
3.1 MESSAGE EXIT DATA (MSGDATA).....	13
3.1.1 Absolute Path.....	13
3.1.2 Relative Path.....	13
3.1.3 Environment Variables.....	14
3.2 SENDER CHANNEL.....	15
3.2.1 Windows.....	15
3.2.2 Linux 32-bit.....	15
3.2.3 Unix and Linux 64-bit.....	15
3.2.4 IBM i.....	16
3.3 RECEIVER CHANNEL.....	17
3.3.1 Windows.....	17
3.3.2 Linux 32-bit.....	17
3.3.3 Unix and Linux 64-bit.....	17
3.3.4 IBM i.....	18
3.4 SERVER CHANNEL.....	19
3.4.1 Windows.....	19
3.4.2 Linux 32-bit.....	19
3.4.3 Unix and Linux 64-bit.....	19
3.4.4 IBM i.....	20
3.5 REQUESTER CHANNEL.....	21
3.5.1 Windows.....	21
3.5.2 Linux 32-bit.....	21
3.5.3 Unix and Linux 64-bit.....	21
3.5.4 IBM i.....	22

3.6 CLUSTER SENDER CHANNEL.....	23
3.6.1 Windows.....	23
3.6.2 Linux 32-bit.....	23
3.6.3 Unix and Linux 64-bit.....	23
3.6.4 IBM i.....	24
3.7 CLUSTER RECEIVER CHANNEL.....	25
3.7.1 Windows.....	25
3.7.2 Linux 32-bit.....	25
3.7.3 Unix and Linux 64-bit.....	25
3.7.4 IBM i.....	26
<b>4 CONFIGURING SVRCONN AND CLNTCONN CHANNELS.....</b>	<b>27</b>
4.1 USER DATA (SENDDATA AND RCVDATA).....	28
4.1.1 Absolute Path.....	28
4.1.2 Relative Path.....	28
4.1.3 Environment Variables.....	29
4.2 SERVER CONNECTION CHANNEL.....	30
4.2.1 Windows.....	30
4.2.2 Linux 32-bit.....	30
4.2.3 Unix and Linux 64-bit.....	31
4.2.4 IBM i.....	31
4.3 CLIENT CONNECTION CHANNEL.....	32
4.3.1 Windows.....	32
4.3.2 Linux 32-bit.....	32
4.3.3 Unix and Linux 64-bit.....	33
4.3.4 IBM i.....	33
4.4 CONFIGURING SEND/RECEIVE EXIT IN MQ EXPLORER.....	34
4.4.1 Local One Time Setup.....	34
4.4.2 Creating a Client Channel Definition Table Entry.....	34
4.5 JAVA BASED APPLICATIONS.....	36
4.5.1 Java Code Samples.....	36
4.5.2 Java Run-Time Settings.....	36
4.6 CONFIGURING MQCEJ FOR USE IN WEBSPHERE APPLICATION SERVER.....	37
4.6.1 Updating WAS's JVM Classpath.....	37
4.6.2 Configuring WAS Admin Console.....	37
4.7 CONFIGURING SECURITY EXIT FOR JBOSS V7 OR HIGHER.....	38
4.7.1 Updating standalone.xml (or standalone-full.xml).....	38
4.7.2 Define the activation-config properties.....	39
4.8 CONFIGURING MQCEJ FOR USE IN J2EE APPLICATION SERVER.....	41
4.8.1 Batch or Quiet mode for J2EE based applications.....	41
<b>5 INIFILE KEYWORDS.....</b>	<b>44</b>
5.1 LOGGING.....	44
5.2 KEYSIZE.....	45
5.3 PERFORM.....	45
5.4 ALLSEGMENTS.....	45
5.5 ENCPASSPHRASE, PASSPHRASE AND USEPP.....	46
5.6 LICENSEFILE.....	47
5.7 LICENSE KEY.....	47

<b>6 MISCELLANEOUS.....</b>	<b>48</b>
6.1 WINDOWS.....	48
6.2 UNIX AND LINUX.....	49
6.3 IBM i.....	49
<b>7 APPENDIX A – MQCE.INI FILE.....</b>	<b>50</b>
<b>8 APPENDIX B – MQCE UPGRADE PROCEDURES.....</b>	<b>53</b>
8.1.1 <i>Windows Upgrade</i> .....	53
8.1.2 <i>Linux 32-bit Upgrade</i> .....	53
8.1.3 <i>Unix and Linux 64-bit Upgrade</i> .....	54
8.1.4 <i>IBM i Upgrade</i> .....	54
<b>9 APPENDIX C - ENCRYPT PASSPHRASE.....</b>	<b>55</b>
9.1 EXAMPLES.....	55
9.1.1 <i>Windows</i> .....	55
9.1.2 <i>Linux 32-bit</i> .....	56
9.1.3 <i>Unix or Linux 64-bit</i> .....	56
9.1.4 <i>IBM i</i> .....	56
<b>10 APPENDIX D - CLIENT CHANNEL DEFINITION TABLE EDITOR.....</b>	<b>57</b>
<b>11 APPENDIX E – CAPITALWARE PRODUCT DISPLAY VERSION.....</b>	<b>59</b>
11.1 EXAMPLES.....	59
11.1.1 <i>Windows</i> .....	59
11.1.2 <i>Linux 32-bit</i> .....	59
11.1.3 <i>Unix and Linux 64-bit</i> .....	59
11.1.4 <i>IBM i</i> .....	59
<b>12 APPENDIX F - EXPLICITLY SETTING VALUES IN MSGDATA, RCVDATA &amp; SENDDATA.....</b>	<b>60</b>
<b>13 APPENDIX G - ENCRYPTION AND DIGITAL SIGNATURE.....</b>	<b>61</b>
<b>14 APPENDIX H - SUPPORT.....</b>	<b>62</b>
<b>15 APPENDIX I - SUMMARY OF CHANGES.....</b>	<b>63</b>
<b>16 APPENDIX J - LICENSE AGREEMENT.....</b>	<b>65</b>
<b>17 APPENDIX K - NOTICES.....</b>	<b>67</b>



# 1 Introduction

## 1.1 Overview

*MQ Channel Encryption* (MQCE) provides encryption for MQ message data. 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.

MQCE provides encryption for message data, which flows between IBM MQ resources. MQCE 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 environments. It operates with Sender, Receiver, Server, Requester, Cluster-Sender, Cluster-Receiver, Server Connection and Client Connection channels of the MQ queue managers.

MQCE is a simple drop-in solution that provides cryptographic protection for MQ queue managers. The protection can be queue manager to queue manager or client application to queue manager.

- Queue manager to queue manager protection means all messages flowing over a channel between 2 queue managers will be encrypted.
- Client application to queue manager protection means application-level message data flowing between a MQ client application and queue manager will be encrypted.

The MQCE can be configured as a queue manager channel message exit or as a channel send/receive exit pair.

MQCE 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). AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process.

MQCE uses the SHA-2 to create a cryptographic hash function (digital signature) for the message data.

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

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

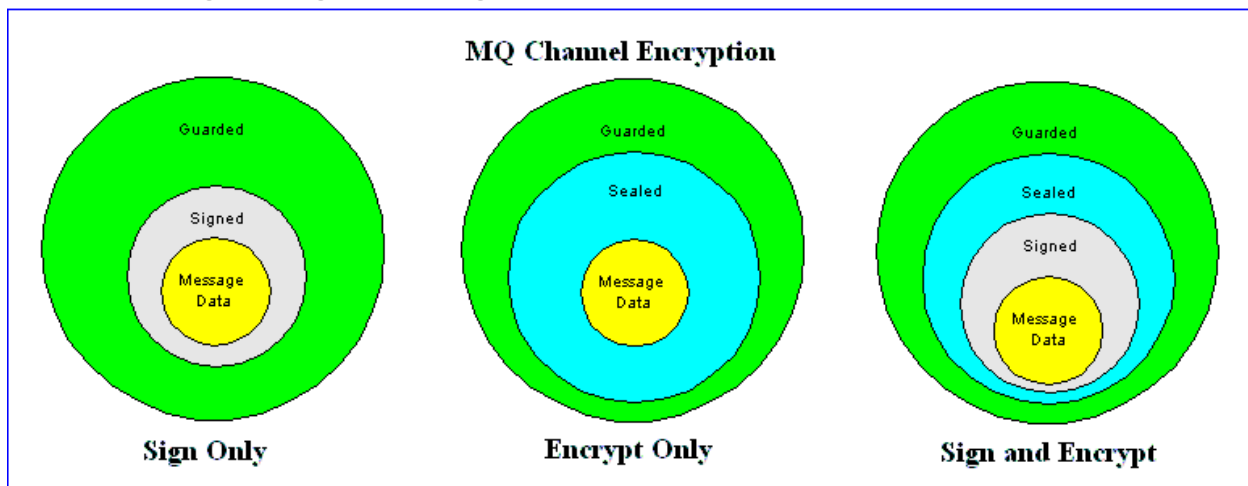
## 1.2 Executive Summary

The MQCE solution is an MQ encryption exit. It is available for a wide range of platforms: AIX, HP-UX, IBM i, Linux, Solaris and Windows.

Major Features of MQCE:

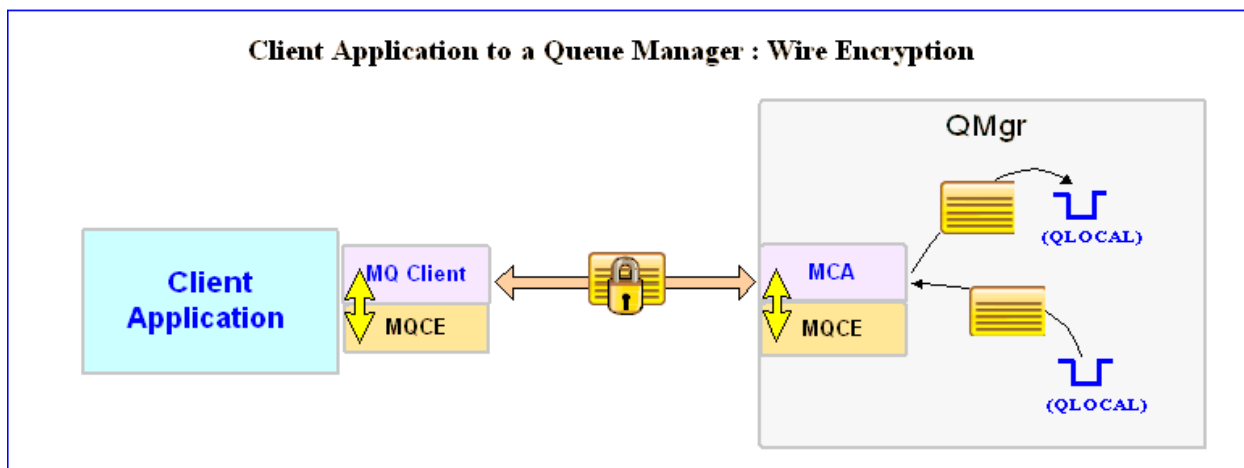
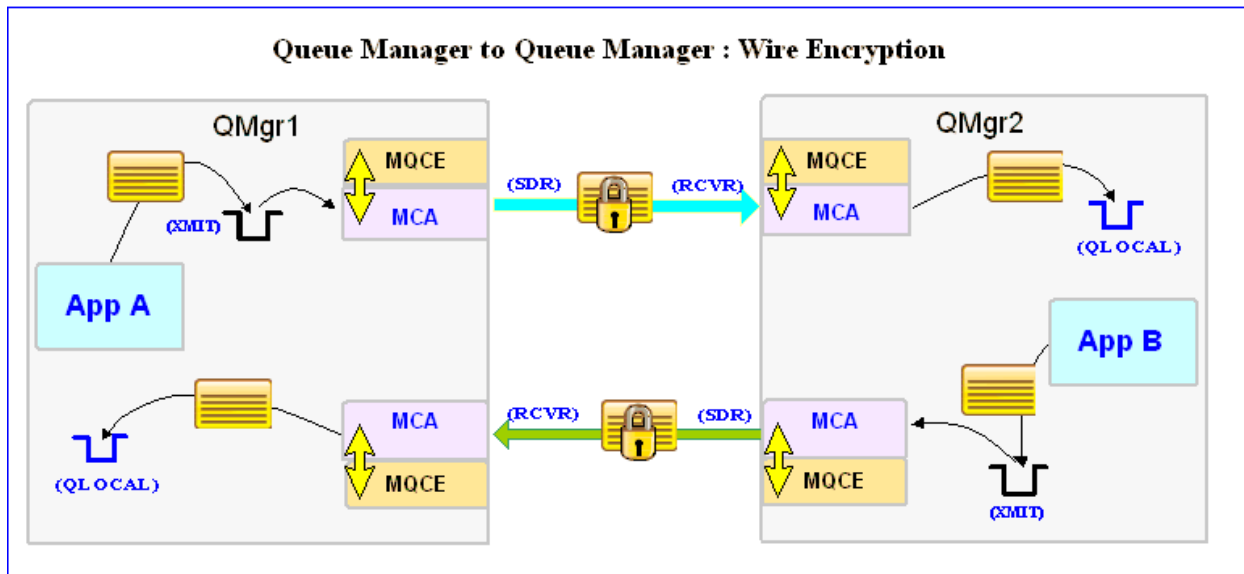
- Easy to set up and configure (unlike SSL)
- No application changes required
- Can be configured as either queue manager to queue manager or client application to queue manager solution
- For both modes, all message data flowing over a channel will be encrypted (nothing missed or forgotten)
- Secure encryption/decryption methodology using AES with 128, 192 or 256-bit keys
- Uses the SHA-2 to create a cryptographic hash function (digital signature)
- Standard MQ feature, GET-with-Convert, is supported
- Provides high-level logging capability for encryption / decryption processing

## 1.3 Message Diagram (Logical View)





## 1.4 Context Diagram (Logical View)



## 1.5 Prerequisites

This section details the minimum supported software levels. These prerequisites apply to both client-side and server-side installations of MQ Channel Encryption.

### 1.5.1 Operating System

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

#### 1.5.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

#### 1.5.1.2 HP-UX IA64

- HP-UX v11.23 or higher

#### 1.5.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

#### 1.5.1.4 Linux x86

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

#### 1.5.1.5 Linux x86\_64 (64-bit)

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

#### 1.5.1.6 Linux on POWER

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

#### 1.5.1.7 Linux on zSeries (64-bit)

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

#### 1.5.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

#### 1.5.1.9 Sun Solaris

- Solaris SPARC v10 & v11
- Solaris x86\_64 v10 & v11

#### 1.5.1.10 Windows

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

## 1.5.2 IBM MQ

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

Operating System	MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
AIX v6.1 or higher	64-bit
HP-UX IA64 v11.23 or higher	64-bit
IBM i (OS/400)	64-bit
Linux x86	32-bit
Linux x86_64	64-bit
Linux on POWER	64-bit
Linux on zSeries	64-bit
Raspberry Pi ARM	32-bit
Solaris SPARC v10 & v11	64-bit
Solaris x86_64 v10 & v11	64-bit
Windows 2008, 2012, 2016, 7, 8, 8.1 & 10	32-bit & 64-bit

## 1.5.3 Windows 32-bit

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

- Microsoft Visual C++ 2010 Redistributable Package (x86)  
[https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist\\_x86.exe](https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist_x86.exe)

## 1.5.4 Windows 64-bit

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

- Microsoft Visual C++ 2010 Redistributable Package (x64)  
[https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist\\_x64.exe](https://download.microsoft.com/download/1/6/5/165255E7-1014-4D0A-B094-B6A430A6BFFC/vcredist_x64.exe)

## 2 Installing MQ Channel Encryption

This section describes how to install Capitalware's MQ Channel Encryption. For Windows, it is available as a Windows installer package called: **mqce-setup.exe**. When the user runs this package, it will install the server-side encryption exit.

### 2.1 Channel Exit

The following files are the platform specific server-side encryption exits and the required initialization file (IniFile).

- **mqce.dll** is the encryption exit DLL for Windows that will be invoked by the MQ component.
- **mqce** is the encryption exit shared library for Unix or Linux that will be invoked by the MQ Server component.
- **MQCE** is the encryption exit for IBM i that will be invoked by the MQ Server component.
- **MQCEJ.jar** is the Java client-side encryption exit that will encrypt the message data that will be invoked by the MQ Client component. It requires Java v1.3 or higher.
- **mqcedn.dll** is the .NET client-side encryption exit that will encrypt the message data that will be invoked by the MQ Client component.
- **mqce.ini** is a sample initialization file for the server-side encryption exit.
- **mqce.sample.MQA1.mqsc** is a sample 'MQA1' MQSC script to define MQ channels with the message exit.
- **mqce.sample.MQB1.mqsc** is a sample 'MQB1' MQSC script to define MQ channels with the message exit.
- **rotatelog.sh** is a simple Unix shell script roll the log file to a backup file.
- **rotatelog.bat** is a simple Windows batch file to roll the log file to a backup file.
- **setmqce.sh** is a simple Unix shell script to set the appropriate file permissions.

## 2.1.1 Windows Installation

To install the encryption exit on Windows, first unzip the **mqce-setup.zip** and then select the appropriate release of MQCE.

- **Java/MQCEJ.jar**
- **mqce-setup.exe** – for Windows 32-bit MQ installations i.e. MQ v7.5 or lower
- **mqce-64-setup-MQv8.exe** – for Windows 64-bit MQ installations i.e. MQ v8 or higher

### 2.1.1.1 Windows 32-bit

Run the **mqce-setup-32bit.exe** file. Follow the on-screen instructions and the encryption exit will be installed in the **C:\Capitalware\MQCE\** directory (default installation). Note: The 64-bit DLLs of MQCE are included in the **{MQCE\_Install\_Path}\64\** directory.

### 2.1.1.2 Windows 64-bit

Run the **mqce-setup-64bit.exe** file. Follow the on-screen instructions and the encryption exit will be installed in the **C:\Capitalware\MQCE\** directory (default installation). Note: The 32-bit DLLs of MQCE are included in the **{MQCE\_Install\_Path}\32\** directory.

## 2.1.2 Linux 32-bit Installation

To install the 32-bit version of MQCE on Linux, first unzip the **mqce-setup.zip** and then select the appropriate TAR file for the target platform. You will find 1 TAR files in the original ZIP file:

- **Java/MQCEJ.jar**
- **Linux\_x86/mqce\_linux.tar**
- **RaspberryPi\_ARM/mqce\_raspberrypi\_arm.tar**

Steps to install the server-side encryption exit:

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

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

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

```
chmod +x setmqce.sh  
./setmqce.sh
```

### 2.1.3 Unix and Linux 64-bit Installation

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

- **Java/MQCEJ.jar**
- **AIX/64-bit/mqce\_aix71\_64.tar** for AIX v7.1 or higher
- **HPUX\_IA64/mqce\_hpux64\_IA64.tar**
- **Linux\_x86\_64/mqce\_linux\_x86\_64.tar**
- **Linux\_POWER/mqce\_linux\_power64.tar**
- **Linux\_zSeries/64-bit/mqce\_linux\_zseries64.tar**
- **Solaris\_SPARC/64-bit/mqce\_solaris10\_64.tar** for Solaris SPARC v10 or higher
- **Solaris\_x86\_64/mqce\_solaris\_x86\_64.tar**

Steps to install the server-side encryption exit:

1. ftp or copy the selected TAR file to the target platform to the */var/mqm/* directory.
2. Un-tar the *mqce\_XXX.tar* file into the */var/mqm/exits/* and */var/mqm/exits64/* sub-directories (XXX is either aix, hpux, solaris or linux)

```
cd /var/mqm/  
tar -xvf mqce_XXX64.tar
```

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

```
chmod +x setmqce.sh  
./setmqce.sh
```

## 2.1.4 IBM i Installation

To install the MQCE on IBM i, first unzip the **mqce-setup.zip** and then select the files in the IBM i directory.

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

Steps to install the channel exit:

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

```
CRTSAVF FILE(QGPL/MQCE)
```

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

```
ftp -s:mqce_iseries.ftp iseries_hostname
```

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

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

```
RSTLIB SAVLIB(MQCE) DEV(*SAVF) SAVF(QGPL/MQCE)  
CLRSAVF FILE(QGPL/MQCE)  
CHGOBJOWN OBJ(MQCE) OBJTYPE(*LIB) NEWOWN(QMQM)  
qsh  
cd /QIBM/UserData/mqm/  
tar -xvf mqce_iseries.tar  
chown -R QMQM mqce  
rm mqce_iseries.tar
```

### 2.1.5 IBM APAR IY91269

An issue was discovered with IBM's IBM MQ for Windows v6.0.2.1. This issue affects the use of any client-side Send or Receive exits including MQCE. IBM has fixed the issue. The fix will be included in the IBM MQ for Windows v6.0.2.2 and higher releases.

<http://www-1.ibm.com/support/docview.wss?uid=swg1IY91269>

A copy of the fix has been included in the directory, **APAR\MQ\_V6\IY91269**, which can be found on the MQCE CD and in the MQCE download file. Please follow the directions in the README.txt file found in the APAR\_IY91269 directory.

### 2.1.6 IBM APAR IZ87819

Issues were discovered with IBM's MQ Java and MQ Explorer v7.0.1.0 and v7.0.1.3. The issue affect the use of any Java MQ application using send/receive exits. IBM has fixed the issue.

<http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg1IZ87819>

***Warning: Please exit all Java applications and MQ Explorer before applying the fix.***

If you are using MQ Explorer v7.0.1.0 or v7.0.1.3, you will need to apply APAR IZ87819. A copy of the fixed JAR file has been included in the directory, **APAR\MQ\_V7\IZ87819**, which can be found on the MQCE CD and in the MQCE download file.

To apply the fix, execute fix\_MQ\_v7.bat script, go to C:\Capitalware\MQCE and then run:

**C:\Capitalware\MQCE\APAR\MQ\_V7\fix\_MQ\_v7.bat**



## 2.1.7 MQCE-GUI Installation

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

5. **MQCE-GUI/mqcegui-wthjre.exe** (for Windows)
6. **MQCE-GUI/mqcegui.zip** (for Unix, Linux or macOS)

### 2.1.7.1 MQCE-GUI Installation on Windows

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

### 2.1.7.2 MQCE-GUI Installation on Unix, Linux or macOS

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

i.e. Do the following command:

```
unzip mqcegui.zip
```

### 3 Configuring QMgr to QMgr Channels

This section describes how to configure the encryption exit.

For normal operation of the MQCE solution, configuration parameters can be specified in the MSGDATA attribute field.

- MSGDATA

#### **L=0000-AAAA-BBBBBBBB**

Where '0000-AAAA-BBBBBBBB' is the license key supplied by Capitalware Inc. It is best if the user uses the Capitalware supplied License file rather than explicitly setting the license key.

#### **K=256**

Where '256' is the key size. Valid values are 128, 192 or 256.

#### **P=E**

Where 'E' is value for the Perform keyword. Perform supports 3 values [S / E / B].

#### **D=Y**

Where 'Y' enables LogMode of Debug.

Note: Use a semicolon to separate the MQCE parameters in the MSGDATA attribute field.

Alternatively, the MQ Admin can specify an IniFile in the MSGDATA attribute.

- For Windows, the MSGDATA attribute would be:  
**C:\Capitalware\MQCE\mqce.ini**
- For IBM MQ 32-bit on Unix and Linux, the MSGDATA attribute would be:  
**/var/mqm/exits/mqce.ini**
- For IBM MQ 64-bit on Unix and Linux, the MSGDATA attribute would be:  
**/var/mqm/exits64/mqce.ini**
- For IBM MQ on IBM i, the MSGDATA attribute would be:  
**mqce.ini**

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.1 Message Exit Data (MSGDATA)

*Message Exit Data (MSGDATA) field must NOT exceed 32 characters.* In order to work with this limitation, MQCE supports 3 ways to specify an IniFile path: absolute path, relative path and environment variable.

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

#### 3.1.1 Absolute Path

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

E.g. Windows

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
      TRPTYPE(TCP) +
      XMITQ('MQB1.XMIT') +
      CONNAME('127.0.0.1(1415)') +
      MSGEXIT('C:\Capitalware\MQCE\mqce(CE)') +
      MSGDATA('C:\Capitalware\MQCE\mqce.ini') +
      REPLACE
```

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

**C:\Capitalware\MQCE\**

#### 3.1.2 Relative Path

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

E.g. Unix

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
      TRPTYPE(TCP) +
      XMITQ('MQB1.XMIT') +
      CONNAME('127.0.0.1(1415)') +
      MSGEXIT('/var/mqm/exits/mqce(CE)') +
      MSGDATA('mqce.ini') +
      REPLACE
```

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

**/var/mqm/exits/**

### 3.1.3 Environment Variables

#### 3.1.3.1 Global Environment Variable

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

e.g. Unix

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

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ('MQB1.XMIT') +  
  CONNAME('127.0.0.1(1415)') +  
  MSGEXIT('/var/mqm/exits64/mqce(CE)') +  
  MSGDATA('mqce.ini') +  
  REPLACE
```

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

#### 3.1.3.2 Queue Manager Specific Environment Variable

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

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

```
export MQCE_HOME_MQA1=/really/long/path/MQHA/QMgrName/data2/
```

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ('MQB1.XMIT') +  
  CONNAME('127.0.0.1(1415)') +  
  MSGEXIT('/var/mqm/exits64/mqce(CE)') +  
  MSGDATA('mqce.ini') +  
  REPLACE
```

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

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

## 3.2 Sender Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the SDR Channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.2.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('C:\Capitalware\MQCE\mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.2.2 Linux 32-bit

On Unix and Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.2.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits64/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.2.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

**MQCE**      **MQCE**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ( 'MQB1.XMIT' ) +  
  CONNAME( '127.0.0.1(1415)' ) +  
  MSGEXIT('MQCE      MQCE      ') +  
  MSGDATA('K=256') +  
  REPLACE
```

### 3.3 Receiver Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the RCVR Channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

#### 3.3.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RCVR ) +  
TRPTYPE( TCP ) +  
MSGEXIT( 'C:\Capitalware\MQCE\mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

#### 3.3.2 Linux 32-bit

On Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RCVR ) +  
TRPTYPE( TCP ) +  
MSGEXIT( '/var/mqm/exits/mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

#### 3.3.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RCVR ) +  
TRPTYPE( TCP ) +  
MSGEXIT( '/var/mqm/exits64/mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

### 3.3.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

**MQCE**      **MQCE**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RCVR ) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( 'MQCE      MQCE      ' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```



## 3.4 Server Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the SVR Channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.4.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('C:\Capitalware\MQCE\mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.4.2 Linux 32-bit

On Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.4.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits64/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.4.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

**MQCE**      **MQCE**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +  
  TRPTYPE(TCP) +  
  XMITQ( 'MQB1.XMIT' ) +  
  CONNAME( '127.0.0.1(1415)' ) +  
  MSGEXIT('MQCE      MQCE      ') +  
  MSGDATA('K=256') +  
  REPLACE
```

## 3.5 Requester Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the RQSTR channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.5.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RQSTR ) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( 'C:\Capitalware\MQCE\mqce(CE)' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```

### 3.5.2 Linux 32-bit

On Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RQSTR ) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( '/var/mqm/exits/mqce(CE)' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```

### 3.5.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RQSTR ) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( '/var/mqm/exits64/mqce(CE)' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```

### 3.5.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT

**MQCE**      **MQCE**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE( RQSTR ) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( 'MQCE            MQCE            ' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```

## 3.6 Cluster Sender Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the CLUSSDR Channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.6.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('C:\Capitalware\MQCE\mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.6.2 Linux 32-bit

On Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.6.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
  TRPTYPE(TCP) +
  XMITQ('MQB1.XMIT') +
  CONNAME('127.0.0.1(1415)') +
  MSGEXIT('/var/mqm/exits64/mqce(CE)') +
  MSGDATA('K=256') +
  REPLACE
```

### 3.6.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT

**MQCE**      **MQCE**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
  TRPTYPE(TCP) +
  XMITQ( 'MQB1.XMIT' ) +
  CONNAME( '127.0.0.1(1415)' ) +
  MSGEXIT('MQCE            MQCE            ') +
  MSGDATA('K=256') +
  REPLACE
```

## 3.7 Cluster Receiver Channel

This section describes the necessary entries to enable the encryption exit. The MQ Administrator will need to update 2 fields of the CLUSRCVR Channel that the encryption exit will be applied to.

*Note: The Message Exit Data (MSGDATA) field must NOT exceed 32 characters.*

### 3.7.1 Windows

On Windows, MSGEXIT and MSGDATA will contain the following values assuming a default install.

- MSGEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1') CHLTYPE(CLUSRCVR) +  
TRPTYPE( TCP ) +  
MSGEXIT( 'C:\Capitalware\MQCE\mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

### 3.7.2 Linux 32-bit

On Linux, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1') CHLTYPE(CLUSRCVR) +  
TRPTYPE( TCP ) +  
MSGEXIT( '/var/mqm/exits/mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

### 3.7.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT  
**/var/mqm/exits64/mqce(CE)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1') CHLTYPE(CLUSRCVR) +  
TRPTYPE( TCP ) +  
MSGEXIT( '/var/mqm/exits64/mqce(CE)' ) +  
MSGDATA( 'K=256' ) +  
REPLACE
```

### 3.7.4 IBM i

On IBM i, MSGEXIT and MSGDATA will contain the following values assuming a default install:

- MSGEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

**MQCE**      **MQCE**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE(CLUSRCVR) +  
  TRPTYPE( TCP ) +  
  MSGEXIT( 'MQCE      MQCE      ' ) +  
  MSGDATA( 'K=256' ) +  
  REPLACE
```



## 4 Configuring SVRCONN and CLNTCONN Channels

This section describes how to configure the encryption exit.

For normal operation of the MQCE solution, configuration parameters can be specified in the SENDDATA and RCVDATA attribute fields.

- SENDDATA
- RCVDATA

### **L=0000-AAAA-BBBBBBBB**

Where '0000-AAAA-BBBBBBBB' is the license key supplied by Capitalware Inc. It is best if the user uses the Capitalware supplied License file rather than explicitly setting the license key.

### **K=256**

Where '256' is the key size. Valid values are 128, 192 or 256.

### **P=E**

Where 'E' is value for the Perform keyword. Perform supports 3 values [S / E / B].

### **D=Y**

Where 'Y' enables LogMode of Debug.

Note: Use a semicolon to separate the MQCE parameters in the SENDDATA and RCVDATA attribute fields.

Alternatively, the MQ Admin can specify an IniFile in the SENDDATA and RCVDATA attribute fields.

- For Windows, the SENDDATA and RCVDATA attribute fields would be:  
**C:\Capitalware\MQCE\mqce.ini**
- For Unix and Linux for IBM MQ 32-bit, the SENDDATA and RCVDATA attribute fields would be:  
**/var/mqm/exits/mqce.ini**
- For Unix and Linux for IBM MQ 64-bit, the SENDDATA and RCVDATA attribute fields would be:  
**/var/mqm/exits64/mqce.ini**
- For IBM i for IBM MQ, the SENDDATA and RCVDATA attribute fields would be:  
**mqce.ini**

***Note: The Send / Receive Exit Data field must NOT exceed 32 characters.***

## 4.1 User Data (SENDDATA and RCVDATA)

*User Data (SENDDATA and RCVDATA) field must NOT exceed 32 characters.* In order to work with this limitation, MQCE supports 3 ways to specify an IniFile path: absolute path, relative path and environment variable.

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

### 4.1.1 Absolute Path

Absolute pathing (specifying the complete path) for the SENDDATA and/or RCVDATA works on Linux, Unix and Windows platforms.

E.g. Windows

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
  TRPTYPE(TCP) +
  SENDEXIT('C:\Capitalware\MQCE\mqce(CE)') +
  SENDDATA('C:\Capitalware\MQCE\mqce.ini') +
  RCVEXIT('C:\Capitalware\MQCE\mqce(CE)') +
  RCVDATA('C:\Capitalware\MQCE\mqce.ini') +
  REPLACE
```

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

### 4.1.2 Relative Path

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

E.g. Unix

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
  TRPTYPE(TCP) +
  SENDEXIT('/var/mqm/exits/mqce(CE)') +
  SENDDATA('mqce.ini') +
  RCVEXIT('/var/mqm/exits/mqce(CE)') +
  RCVDATA('mqce.ini') +
  REPLACE
```

Hence, MQCE will use the following path as the IniFile path:  
**/var/mqm/exits/**

### 4.1.3 Environment Variables

#### 4.1.3.1 Global Environment Variable

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

e.g. Unix

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

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('/var/mqm/exits64/mqce(CE)') +  
  SENDDATA('mqce.ini') +  
  RCVEXIT('/var/mqm/exits64/mqce(CE)') +  
  RCVDATA('mqce.ini') +  
  REPLACE
```

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

#### 4.1.3.2 Queue Manager Specific Environment Variable

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

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

```
export MQCE_HOME_MQA1=/really/long/path/MQHA/QMgrName/data/
```

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('/var/mqm/exits64/mqce(CE)') +  
  SENDDATA('mqce.ini') +  
  RCVEXIT('/var/mqm/exits64/mqce(CE)') +  
  RCVDATA('mqce.ini') +  
  REPLACE
```

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

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

## 4.2 Server Connection Channel

This section describes the necessary entries to enable the server-side encryption exit. The MQ Administrator will need to update 2 fields of the SVRCONN Channel that the server-side encryption exit will be applied to.

*Note: The Send / Receive Exit Data field must NOT exceed 32 characters.*

### 4.2.1 Windows

On Windows, SENDEXIT, SENDDATA, RCVEXIT and RCVDATA will contain the following values assuming a default install.

- SENDEXIT  
**C:\Capitalware\MQCE\mqce(CE)**
- RCVEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('C:\Capitalware\MQCE\mqce(CE)') +  
  SENDDATA('K=256') +  
  RCVEXIT('C:\Capitalware\MQCE\mqce(CE)') +  
  RCVDATA('K=256') +  
  REPLACE
```

### 4.2.2 Linux 32-bit

On Linux, SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT  
**/var/mqm/exits/mqce(CE)**
- RCVEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('/var/mqm/exits/mqce(CE)') +  
  SENDDATA('K=256') +  
  RCVEXIT('/var/mqm/exits/mqce(CE)') +  
  RCVDATA('K=256') +  
  REPLACE
```

### 4.2.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT  
`/var/mqm/exits64/mqce(CE)`
- RCVEXIT  
`/var/mqm/exits64/mqce(CE)`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('/var/mqm/exits64/mqce(CE)') +  
  SENDDATA('K=256') +  
  RCVEXIT('/var/mqm/exits64/mqce(CE)') +  
  RCVDATA('K=256') +  
  REPLACE
```

### 4.2.4 IBM i

On IBM i, SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

`MQCE`      `MQCE`

- RCVEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

- `MQCE`      `MQCE`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +  
  TRPTYPE(TCP) +  
  SENDEXIT('MQCE      MQCE      ') +  
  SENDDATA('K=256') +  
  RCVEXIT('MQCE      MQCE      ') +  
  RCVDATA('K=256') +  
  REPLACE
```

### 4.3 Client Connection Channel

This section describes the necessary entries to enable the server-side encryption exit. The MQ Administrator will need to update 2 fields of the CLNTCONN Channel that the server-side encryption exit will be applied to.

*Note: The Send / Receive Exit Data field must NOT exceed 32 characters.*

#### 4.3.1 Windows

On Windows, SENDEXIT, SENDDATA, RCVEXIT and RCVDATA will contain the following values assuming a default install.

- SENDEXIT  
**C:\Capitalware\MQCE\mqce(CE)**
- RCVEXIT  
**C:\Capitalware\MQCE\mqce(CE)**

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(CLNTCONN) +  
  TRPTYPE(TCP) +  
  CONNAME('127.0.0.1(1414)') +  
  QMNAME('MQA1') +  
  SENDEXIT('C:\Capitalware\MQCE\mqce(CE)') +  
  SENDDATA('K=256') +  
  RCVEXIT('C:\Capitalware\MQCE\mqce(CE)') +  
  RCVDATA('K=256') +  
  REPLACE
```

#### 4.3.2 Linux 32-bit

On Linux, SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT  
**/var/mqm/exits/mqce(CE)**
- RCVEXIT  
**/var/mqm/exits/mqce(CE)**

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(CLNTCONN) +  
  TRPTYPE(TCP) +  
  CONNAME('127.0.0.1(1414)') +  
  QMNAME('MQA1') +  
  SENDEXIT('/var/mqm/exits/mqce(CE)') +  
  SENDDATA('K=256') +  
  RCVEXIT('/var/mqm/exits/mqce(CE)') +  
  RCVDATA('K=256') +  
  REPLACE
```

### 4.3.3 Unix and Linux 64-bit

On Unix and Linux (excluding Linux x86), SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT  
`/var/mqm/exits64/mqce(CE)`
- RCVEXIT  
`/var/mqm/exits64/mqce(CE)`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(CLNTCONN) +
  TRPTYPE(TCP) +
  CONNAME('127.0.0.1(1414)') +
  QMNAME('MQA1') +
  SENDEXIT('/var/mqm/exits64/mqce(CE)') +
  SENDDATA('K=256') +
  RCVEXIT('/var/mqm/exits64/mqce(CE)') +
  RCVDATA('K=256') +
  REPLACE
```

### 4.3.4 IBM i

On IBM i, SENDEXIT and SENDDATA will contain the following values assuming a default install:

- SENDEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

`MQCE`      `MQCE`

- RCVEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).

`MQCE`      `MQCE`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(CLNTCONN) +
  TRPTYPE(TCP) +
  CONNAME('127.0.0.1(1414)') +
  QMNAME('MQA1') +
  SENDEXIT('MQCE            MQCE            ') +
  SENDDATA('K=256') +
  RCVEXIT('MQCE            MQCE            ') +
  RCVDATA('K=256') +
  REPLACE
```

## 4.4 Configuring Send/Receive Exit in MQ Explorer

This section describes the necessary steps to enable send/receive exit in MQ Explorer for Windows or Linux. *Note: Make sure you have applied the APAR from section 2.1.6 (if applicable).*

### 4.4.1 Local One Time Setup

To use the MQCE send/receive exit with MQ Explorer, the user must do a one time setup. The user needs to add a statement to the runmqcfg\_rcp.cmd batch script. runmqcfg\_rcp.cmd is located in the bin directory of the IBM MQ installation directory.

i.e.

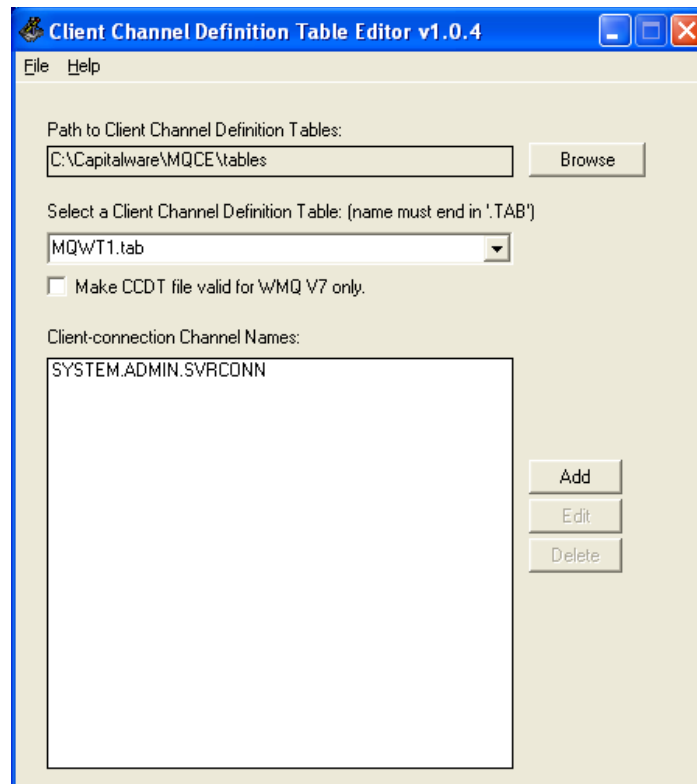
**C:\Program Files\IBM\IBM MQ\bin\runmqcfg\_rcp.cmd**

Add the following line to the runmqcfg\_rcp.cmd batch script just before the *start "IBM MQ Explorer" %AMQ\_EXPLORER%* line.

```
set AMQ_EXPLORER=%AMQ_EXPLORER% "-Dcom.ibm.mq.exitClasspath=C:/Capitalware/MQCE/MQCEJ.jar"
```

### 4.4.2 Creating a Client Channel Definition Table Entry

MQ Explorer requires the user to create a client channel definition table to use the MQCE send/receive exit. To enable user-defined send/receive exit for encryption, do the following steps:





1. Start the Client Channel Definition Table Editor (From the **Start** -> **All Programs** menu)
2. Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')
3. Click the Add button to insert a new CLNTCONN channel or click the Edit button to edit an existing CLNTCONN channel.

**Edit Client-connection Channel Definition**

Channel Name: SYSTEM.ADMIN.SVRCONN

Description: Client channel utilizing MQCE

Connection Name: 127.0.0.1(1415)

Queue Manager Name: MQWT1

Max Message Length: 4194304

Heartbeat Interval: 300

Affinity: Preferred

Security Exit Name: [Empty]

Security Exit Data: [Empty]

Send Exit Name: biz.capitalware.mqce.MQCEJ

Send Exit Data: [Empty]

Receive Exit Name: biz.capitalware.mqce.MQCEJ

Receive Exit Data: [Empty]

Buttons: Save, Cancel

4. For **Send Exit Name** and **Receive Exit Name**, select **biz.capitalware.mqce.MQCEJ** from the drop-down list.

A client channel definition table will be created in the 'tables' directory under the default install directory.

For the example above, a client channel definition table will be found (assuming a default install) at this location:

**C:\Capitalware\MQCE\tables\MQW1.TAB**

## 4.5 Java based Applications

For Windows, Unix or Linux, set MQCE\_PASPHRASE **OR** set MQCE\_FILE JVM arguments. Do not set both groups of JVM arguments.

### 4.5.1 Java Code Samples

This section describes how to code a Java application to invoke MQCE.

#### 4.5.1.1 Java Send Exit Code Sample

```
MQEnvironment.sendExit = new MQCEJ();
```

#### 4.5.1.2 Java Receive Exit Code Sample

```
MQEnvironment.receiveExit = new MQCEJ();
```

### 4.5.2 Java Run-Time Settings

To use JVM arguments to specify the License and PassPhrase or a file that will contain the License and PassPhrase values, do the following:

- Add the following JVM arguments to your java command-line parameters to specify the PassPhrase:

```
java -DMQCE_PASSPHRASE=A3JiYt9LWERUPKw com.acme.run.Thing
```

- Add the following JVM argument to your java command-line parameters to specify a file that will contain the License and PassPhrase values:

On Windows:

```
java -DMQCE_FILE=C:\Capitalware\MQCE\mqcej.ini com.acme.run.Thing
```

On Unix / Linux:

```
java -DMQCE_FILE=/home/user/mqcej.ini com.acme.run.Thing
```

## 4.6 Configuring MQCEJ for use in WebSphere Application Server

This section describes the necessary steps to enable Send/Receive Exits in IBM's WebSphere Application Server (WAS):

### 4.6.1 Updating WAS's JVM Classpath

The MQCEJ.jar file is intended for use by all applications deployed on WAS. Copy the MQCEJ.jar file to the *ws.ext.dirs* directory. As a result, the jar file will be loaded by the WAS extensions class loader.

On Windows, the *ws.ext.dirs* may be configured as

```
{WAS_Install_Path}\webSphere6\AppServer\lib\ext
```

On Unix / Linux, the *ws.ext.dirs* may be configured as

```
{WAS_Install_Path}/webSphere6/AppServer/lib/ext
```

### 4.6.2 Configuring WAS Admin Console

To add the **biz.capitalware.mqce.MQCEJ** class to your deployed WAS application, add the following custom property to your IBM MQ connection factory as show below:

Required:

Property Name	Value
SENDEXIT	biz.capitalware.mqce.MQCEJ
RECEXIT	biz.capitalware.mqce.MQCEJ

Optional:

Property Name	Value
SENDEXITINIT	K=256
RECEXITINIT	K=256

See Appendix F for more information about exit data values.

## 4.7 Configuring Security Exit for JBoss V7 or higher

This section describes the necessary steps to enable Security Exits in JBoss V7 or higher.

### 4.7.1 Updating standalone.xml (or standalone-full.xml)

#### 4.7.1.1 MQCEJ Information

In the *standalone.xml* (or *standalone-full.xml*), add the following property to the *<system-properties>* section (This will tell the resource adapter's classloader where the exit classes are located.):

On Windows:

```
<property name="com.ibm.mq.cfg.ClientExitPath.JavaExitsClasspath" value="C:/Capitalware/MQCE/MQCEJ.jar"/>
```

On Unix / Linux:

```
<property name="com.ibm.mq.cfg.ClientExitPath.JavaExitsClasspath" value="/var/mqm/exits64/MQCEJ.jar"/>
```

#### 4.7.1.2 MQ Connectivity Information

In the *standalone.xml* (or *standalone-full.xml*), add the following property to the *<connection-definition>* section.

```
<config-property name="channel">SYSTEM.DEF.SVRCONN</config-property>  
<config-property name="hostName">localhost</config-property>  
<config-property name="transportType">CLIENT</config-property>  
<config-property name="queueManager">ExampleQM</config-property>  
<config-property name="port">1414</config-property>
```

## 4.7.2 Define the activation-config properties

Next, the user needs to define the activation-config properties. It can be done either using jboss-ejb3.xml file or annotation.

### 4.7.2.1 Using the jboss-ejb3.xml file

```
<jee:activation-config-property>
  <jee:activation-config-property-name>sendExit</jee:activation-config-property-name>
  <jee:activation-config-property-value>biz.capitalware.mqce.MQCEJ</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config-property>
  <jee:activation-config-property-name>sendExitInit</jee:activation-config-property-name>
  <jee:activation-config-property-value>K=256</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config-property>
  <jee:activation-config-property-name>receiveExit</jee:activation-config-property-name>
  <jee:activation-config-property-value>biz.capitalware.mqce.MQCEJ</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config-property>
  <jee:activation-config-property-name>receiveExitInit</jee:activation-config-property-name>
  <jee:activation-config-property-value></jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config>
  <jee:activation-config-property>
    <jee:activation-config-property-name>channel</jee:activation-config-property-name>
    <jee:activation-config-property-value>CHANNEL.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destination</jee:activation-config-property-name>
    <jee:activation-config-property-value>QUEUE.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destinationType</jee:activation-config-property-name>
    <jee:activation-config-property-value>javax.jms.Queue</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>hostName</jee:activation-config-property-name>
    <jee:activation-config-property-value>HOST.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
</jee:activation-config>
```

## 4.7.2.2 Using Annotation

```
@MessageDriven(
    activationConfig = {
        @ActivationConfigProperty(propertyName = "sendExit", propertyValue = "biz.capitalware.mqce.MQCEJ"),
        @ActivationConfigProperty(propertyName = "sendExitInit", propertyValue = "K=256")

        @ActivationConfigProperty(propertyName = "receiveExit", propertyValue = "biz.capitalware.mqce.MQCEJ"),
        @ActivationConfigProperty(propertyName = "receiveExitInit", propertyValue = "")

        @ActivationConfigProperty(propertyName = "channel", propertyValue="CHANNEL.NAME"),
        @ActivationConfigProperty(propertyName = "destination", propertyValue = "QUEUE.NAME"),
        @ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue"),
        @ActivationConfigProperty(propertyName = "hostName", propertyValue = "HOST.NAME"),
        @ActivationConfigProperty(propertyName = "port", propertyValue = "1414"),
        @ActivationConfigProperty(propertyName = "transportType", propertyValue = "CLIENT"),

    })

@ResourceAdapter(value="wmq.jmsra.rar")
```

## 4.8 Configuring MQCEJ for use in J2EE Application Server

This section describes the necessary steps to enable channel exit in a J2EE Application Server like BEA's WebLogic Server.

### 4.8.1 Batch or Quiet mode for J2EE based applications

To run in batch or quiet mode, the user can explicitly set the value of the License and PassPhrase in the channel's SendExitInit / ReceiveExitInit field or specify a file in the SendExitInit / ReceiveExitInit field.

To explicitly set the PassPhrase value, do the following for the user-defined client-side channel exit for authentication:

#### 4.8.1.1 Updating Application Server's JVM Classpath

##### *Windows:*

The JAR file is located at (assuming a default install of `C:\Capitalware\MQCE`):

```
SET CLASSPATH=C:\Capitalware\MQCE\MQCEJ.jar;%CLASSPATH%
```

##### *Unix and Linux (32-bit):*

The JAR file is located at (assuming a default install of `/var/mqm/exits/`):

```
export CLASSPATH=/var/mqm/exits/MQCEJ.jar;%CLASSPATH%
```

##### *Unix and Linux (64-bit):*

The JAR file is located at (assuming a default install of `/var/mqm/exits64/`):

```
export CLASSPATH=/var/mqm/exits64/MQCEJ.jar:$CLASSPATH
```

#### 4.8.1.2 Updating Application's JMS binding file

Use IBM MQ's JMSAdmin command to define or alter a QCF (QueueConnectionFactory) or TCF (TopicConnectionFactory). The client-side channel exit also works with the XA versions of QCF and TCF (i.e. XAQCF and XATCF). In the SendExitInit / ReceiveExitInit field, include the License and PassPhrase information as follows:

```
define tcf(tcClient) qmgr(MY.QMGR)
channel(SYSTEM.DEF.SVRCONN) hostname(MYHOSTNAME) port(1414)
transport(CLIENT) SENDEXIT(biz.capitalware.mqce.MQCEJ)
SENDEXITINIT(K=256)
RCVEXIT(biz.capitalware.mqce.MQCEJ)
RCVEXITINIT(K=256)
```

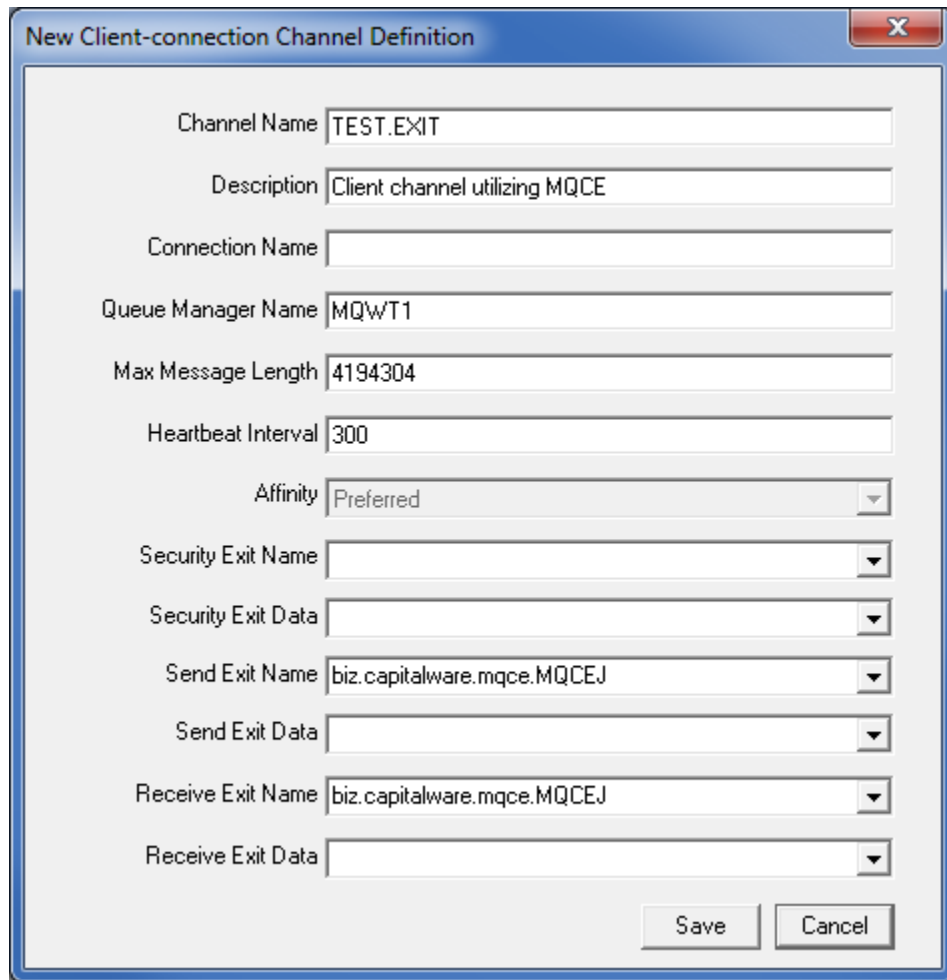
or

```
define qcf(qcClient) qmgr(MY.QMGR)
channel(SYSTEM.DEF.SVRCONN) hostname(MYHOSTNAME) port(1414)
transport(CLIENT) SENDEXIT(biz.capitalware.mqce.MQCEJ)
SENDEXITINIT(K=256)
RCVEXIT(biz.capitalware.mqce.MQCEJ)
RCVEXITINIT(K=256)
```



### 4.8.1.3 Updating Application's CCDT file

The MQAdmin can use CCDTE program to update the CCDT file and add the send and receive exit fields. Use the dropdown to select: biz.capitalware.mqce.MQCEJ



The screenshot shows a dialog box titled "New Client-connection Channel Definition". It contains the following fields and values:

- Channel Name: TEST.EXIT
- Description: Client channel utilizing MQCE
- Connection Name: (empty)
- Queue Manager Name: MQWT1
- Max Message Length: 4194304
- Heartbeat Interval: 300
- Affinity: Preferred
- Security Exit Name: (empty)
- Security Exit Data: (empty)
- Send Exit Name: biz.capitalware.mqce.MQCEJ
- Send Exit Data: (empty)
- Receive Exit Name: biz.capitalware.mqce.MQCEJ
- Receive Exit Data: (empty)

Buttons: Save, Cancel

Or the MQAdmin can use the runmqsc program and issue an MQSC command as follows:

```
ALTER CHANNEL(TEST.EXIT) CHLTYPE(CLNTCONN) +
TRPTYPE(TCP) +
SENDEXIT('biz.capitalware.mqce.MQCEJ') +
RCVEXIT('biz.capitalware.mqce.MQCEJ')
```

## 5 IniFile Keywords

### 5.1 Logging

This section describes the necessary entries to enable MQCE to write log information. To enable and control logging, you need 3 keywords in the IniFile:

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

For Windows:

```
LogFile=C:\Capitalware\MQCE\mqce.log
```

For IBM MQ 32-bit on Linux:

```
LogFile=/var/mqm/exits/mqce.log
```

For IBM MQ 64-bit on Unix and Linux:

```
LogFile=/var/mqm/exits64/mqce.log
```

For IBM MQ on IBM i:

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

Token Replacement for LogFile keyword:

- **%QM%** - Substitutes the name of the queue manager
  - **%CHL%** - Substitutes the name of the channel
  - **%UID%** - Substitutes the UserID
  - **%PID%** - Substitutes the Process ID
  - **%TID%** - Substitutes the Thread ID
- **RotateLogDaily** specifies whether or not the log files will be rotated on a daily basis. A Y value for 'RotateLogDaily' will activate this feature; otherwise, the log files will left as is. The default value is Y.

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

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

## 5.2 KeySize

KeySize specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256. The default value is 128.

```
keySize=128
```

## 5.3 Perform

Perform indicates which functionality that MQCE will perform. Perform supports 3 values [S / E / B]. The default value is E.

- **S** means that MQCE will only sign the message
- **E** means that MQCE will only encrypt the message
- **B** means that MQCE will sign and encrypt the message

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

```
Perform=E
```

## 5.4 AllSegments

AllSegments specifies whether or not all TSH segments will be encrypted. AllSegments supports 2 values [Y / N]. The default value is N.

- **Y** means that MQCE will encrypt all TSH segments that are sent
- **N** means that MQCE will only encrypt the TSH segments that contain message data

```
AllSegments=N
```

## 5.5 EncPassPhrase, PassPhrase and UsePP

To enable the use of the user's own PassPhrase, you need 2 keywords in the IniFile:

- **UsePP** allows the use of a user specified PassPhrase
- **EncPassPhrase** specifies an encrypted PassPhrase for this IniFile that will be used for message encryption and/or decryption. See Appendix C for details on creating the encrypted PassPhrase.
- **PassPhrase** specifies the actual PassPhrase that will be used for the message encryption and/or decryption (can be 16, 24 or 32 characters/digits in length).

What not to use for your PassPhrase:

- A famous quotation from literature, holy books, etc.
- Something easily guessed by intuition

What to use for your PassPhrase:

- A random selection of characters and numbers
- Use a mix of upper and lower characters
- Use special characters like slash, dot, comma, ampersand, etc.

Encrypted PassPhrase (See Appendix C for details on creating the encrypted passphrase.):

```
UsePP=Y  
EncPassPhrase=jXzFNlKKwZ52wsQ3CUwqWUBpDaoVRDnLMDkNqhVEOcswMA
```

Plain text PassPhrase:

```
UsePP=Y  
PassPhrase=AekWU31_wky6MZrL
```

Note: If EncPassPhrase keyword is specified then the PassPhrase keyword is ignored.

## 5.6 LicenseFile

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

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

**QMgrName = License\_Key**

### Example:

```
MQA1 = 10C0-AAAA-BBBBBBBB  
MQB1 = 10C0-XXXX-CCCCCCCC
```

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

The following are the default values for LicenseFile:

For Windows:

**LicenseFile=C:\Capitalware\MQCE\mqce\_licenses.ini**

For IBM MQ 32-bit on Unix and Linux:

**LicenseFile=/var/mqm/exits/mqce\_licenses.ini**

For IBM MQ 64-bit on Unix and Linux:

**LicenseFile=/var/mqm/exits64/mqce\_licenses.ini**

For IBM MQ on IBM i:

**LicenseFile=/QIBM/UserData/mqm/mqce/mqce\_licenses.ini**

## 5.7 License Key

This section will describe how to license MQ Channel Encryption to a particular queue manager.

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

Your license will look something like: 0000-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).

```
License=10C0-AAAA-BBBBBBBB
```

## 6 Miscellaneous

This section describes the extra files that were included to help the user get MQCE up and running in a very quick manner.

### 6.1 Windows

#### *Sample IniFile*

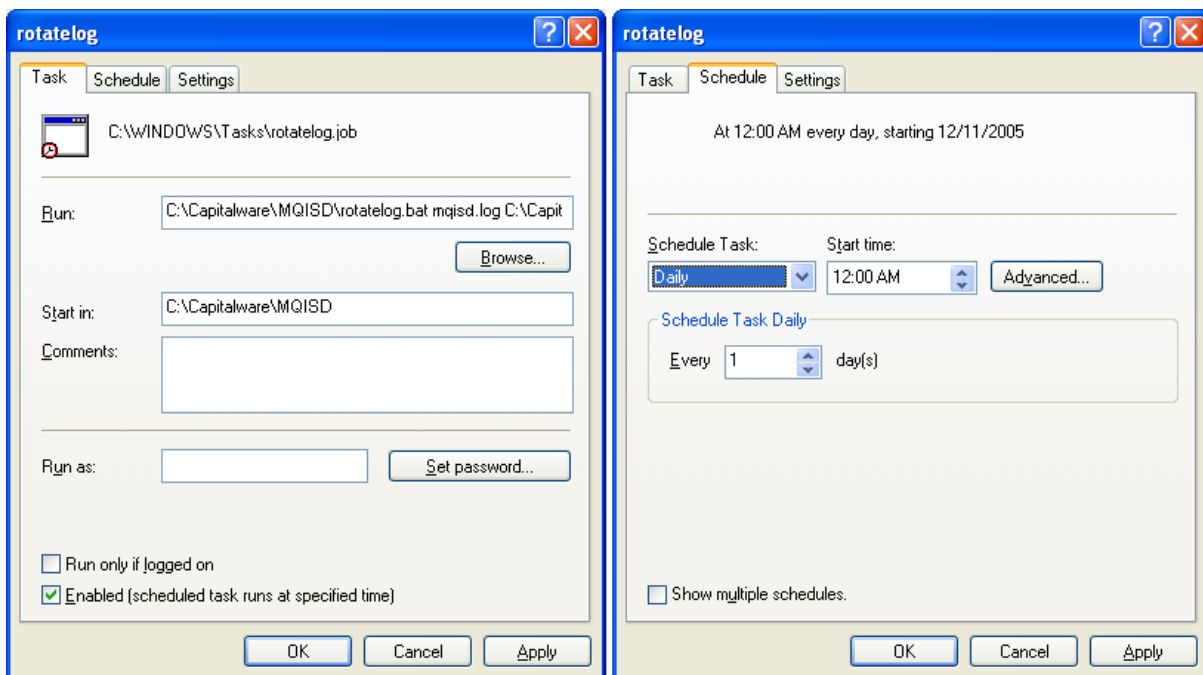
The '*mqce.ini*' file is a basic MQCE IniFile. It has the standard IniFile parameters that the user may need to use or update. The '*mqce.ini.readme*' file is a plain text help file with a description of the parameters.

#### *Sample MQSC scripts*

The '*mqce.sample.mqsc*' file is a sample MQSC script to update the 2 system defined channels with the MQCE encryption exit information.

#### *Rotate log script*

The '*rotatelog.bat*' file is a Windows batch script to rotate (backup) the mqce.log file. Actually, it is generic in implementation; hence, it can be used to rotate any log file that the user wishes to be rotated. The batch script requires 2 parameters: log file name and the directory of log file.



## 6.2 Unix and Linux

### *Sample IniFile*

The '*mqce.ini*' file is a basic MQCE IniFile. It has the standard IniFile parameters that the user may need to use or update. The '*mqce.ini.readme*' file is a plain text help file with a description of the parameters.

### *Sample MQSC scripts*

The '*mqce.sample.MQA1.mqsc*' and '*mqce.sample.MQB1.mqsc*' files are sample MQSC scripts to be used as reference channels with the MQCE encryption exit.

### *Rotate log script*

The '*rotatelog.sh*' file is a Unix / Linux shell script to rotate (backup) the mqce.log file. Actually, it is generic in implementation; hence, it can be used to rotate any log file that the user wishes to be rotated. The shell script requires 2 parameters: log file name and the directory of log file.

Sample daily CRON entry for IBM MQ 32-bit on Unix and Linux:

```
0 0 * * * /var/mqm/exits/rotatelog.sh mqce.log /var/mqm/exits/ > /tmp/mqce.log.run 2 > &1
```

Sample daily CRON entry for IBM MQ 64-bit on Unix and Linux:

```
0 0 * * * /var/mqm/exits64/rotatelog.sh mqce.log /var/mqm/exits64/ > /tmp/mqce.log.run 2 > &1
```

## 6.3 IBM i

### *Sample IniFile*

The '*mqce.ini*' file is a basic MQCE IniFile. It has the standard IniFile parameters that the user may need to use or update. The '*mqce.ini.readme*' file is a plain text help file with a description of the parameters.

### *Sample MQSC scripts*

The '*mqce.sample.mqsc*' file is a sample MQSC script to update the 2 system defined channels with the MQCE channel exit information.

## 7 Appendix A – mqce.ini file

The sample IniFile below is the mqce.ini file supplied for Windows. The IniFile supports the following keywords and their values:

```
LogMode=N
LogFile=C:\Capitalware\MQCE\mqce.log
KeySize=128
License=
```

**Note: Keywords are case sensitive.**

Keyword	Description of Server-side keywords
AllSegments	<p><b>AllSegments</b> specifies whether or not all TSH segments will be encrypted. AllSegments set to 'N' means only message data segments will be encrypted. AllSegments supports 2 values [Y / N]. The default value is N.</p> <p>e.g. AllSegments=Y</p>
EncPassPhrase	<p><b>EncPassPhrase</b> specifies the encrypted PassPhrase for the message encryption and/or decryption. See Appendix C for details on creating the encrypted PassPhrase.</p> <p>e.g. EncPassPhrase=jXzFNIKKwZ52wsQ3CUwqWUBpDaoVRDnLMDkNqhVEOcsWMA</p> <p>Note: Only used if UsePassPhrase is set to 'Y'.</p>
KeySize	<p><b>KeySize</b> specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256. The default value is 128.</p> <p>e.g. KeySize=128</p>
License	<p><b>License</b> specifies the queue manager's license key. Your license will look something like: 0000-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).</p> <p>e.g. License=0000-AAAA-BBBBBBBB</p>



Keyword	Description of Server-side keywords
LicenseFile	<p><b>LicenseFile</b> specifies the location of License file that contains all of the customer's license keys.</p> <p>The following are the default values for LicenseFile:</p> <p>For Windows: LicenseFile=C:\Capitalware\MQCE\mqce_licenses.ini</p> <p>For IBM MQ 32-bit on Unix and Linux: LicenseFile=/var/mqm/exits/mqce_licenses.ini</p> <p>For IBM MQ 64-bit on Unix and Linux: LicenseFile=/var/mqm/exits64/mqce_licenses.ini</p> <p>For IBM MQ on IBM i: LicenseFile=/QIBM/UserData/mqm/mqce/mqce_licenses.ini</p> <p>e.g. LicenseFile=/var/mqm/exits64/mqce_licenses.ini</p>
LogFile	<p><b>LogFile</b> specifies the location of the log file. The default is as follows:</p> <p>For Windows: LogFile=C:\Capitalware\MQCE\mqce.log</p> <p>For IBM MQ 32-bit on Unix and Linux: LogFile=/var/mqm/exits/mqce.log</p> <p>For IBM MQ 64-bit on Unix and Linux: LogFile=/var/mqm/exits64/mqce.log</p> <p>For IBM MQ on IBM i: LogFile=/QIBM/UserData/mqm/mqce/mqce.log</p>
LogMode	<p><b>LogMode</b> specifies what type of logging the user wishes to have. LogMode supports 4 values [Q / N / V / D] where Q is Quiet, N is Normal, V is Verbose and D is Debug. The default value is Q.</p> <p>e.g. LogMode=Q</p>
PassPhrase	<p><b>PassPhrase</b> specifies a user supplied PassPhrase. The PassPhrase can be one of three sizes: 16, 24 or 32 characters/digits in length.</p> <p>e.g. PassPhrase=QPriiTJmr4j7aQ2P</p>

Keyword	Description of Server-side keywords
Perform	<p><b>Perform</b> indicates what functionality that MQCE will perform. Perform supports 3 values [S / E / B]. The default value is E.</p> <ul style="list-style-type: none"> <li>• <b>S</b> means that MQCE will only sign the message</li> <li>• <b>E</b> means that MQCE will only encrypt the message</li> <li>• <b>B</b> means that MQCE will sign and encrypt the message</li> </ul> <p>When signing the message, MQCE creates the digital signature using cryptographic hash function of SHA-2.</p> <p>e.g. Perform=E</p>
RotateLogDaily	<p><b>RotateLogDaily</b> specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y.</p> <p>e.g. RotateLogDaily=Y</p>
UsePP	<p><b>UsePP</b> allows the user to specify their own PassPhrase. UsePP supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UsePP=Y</p>

## 8 Appendix B – MQCE Upgrade Procedures

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

### 8.1.1 Windows Upgrade

- Stop all of the channels using the MQCE exit or completely stop the queue manager.
- Backup all MQCE IniFiles in the MQCE install directory
- If MQCE was installed using the Windows Installer then
  - Click the **Start -> All Programs -> Control Panel -> Add or Remove Programs**, select MQCE from the list and click the **Remove** button then follow the prompts to remove it
  - Run the **mqce-setup.exe** file from the **Windows** directory to install the new version
- Otherwise copy the following files (latest version) to the MQCE install directory:
  - mqce.dll
  - rotatelog.bat
- Restore the MQCE IniFiles if they were altered / deleted.
- Start all of the channels using the MQCE channel exit or restart the queue manager if it was previously stopped.

### 8.1.2 Linux 32-bit Upgrade

- Login under the mqm account
- Stop all of the channels using the MQCE exit or completely stop the queue manager.
- Backup all MQCE IniFiles in the MQCE install directory
- Copy the appropriate tar file to the **/var/mqm/exits/** directory
- Un-tar the contents of the tar file.  
i.e. For AIX, do the following command:  
**tar -xvf mqce\_aix.tar**
- Run the script as follows:  
**./setmqce.sh**
- Restore the MQCE IniFiles if they were altered / deleted.
- Delete the MQCE tar file
- Start all of the channels using the MQCE exit or restart the queue manager if it was previously stopped.

### 8.1.3 Unix and Linux 64-bit Upgrade

- Stop all of the channels using the MQCE exit or completely stop the queue manager.
- Backup all MQCE IniFiles in the MQCE install directory
- Copy the appropriate tar file to the `/var/mqm/exits64/` directory
- Un-tar the contents of the tar file.  
i.e. For AIX, do the following command:  
**tar -xvf mqce\_aix.tar**
- Run the script as follows:  
**./setmqce.sh**
- Restore the MQCE IniFiles if they were altered / deleted.
- Delete the MQCE tar file
- Start all of the channels using the MQCE exit or restart the queue manager if it was previously stopped.

### 8.1.4 IBM i Upgrade

- Stop all of the channels using the MQCE exit or completely stop the queue manager.
- Backup all MQCE IniFiles in the MQCE install directory
- ftp the IBM i files to the IBM i server as follows:

```
ftp -s:mqce_iseriess.ftp iseries_hostname
```

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

binary
cd QGPL
put mqce.savf

quote SITE NAMEFMT 1

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

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

```
RSTLIB SAVLIB(MQCE) DEV(*SAVF) SAVF(QGPL/MQCE)
CLRSVF FILE(QGPL/MQCE)
CHGOBJOWN OBJ(MQCE) OBJTYPE(*LIB) NEWOWN(QMQM)
qsh
cd /QIBM/UserData/mqm/
tar -xvf mqce_iseriess.tar
chown -R QMQM mqce
rm mqce_iseriess.tar
```

- Restore the MQCE IniFiles if they were altered / deleted.
- Start all of the channels using the MQCE exit or restart the queue manager if it was previously stopped.

## 9 Appendix C - Encrypt PassPhrase

The Encrypt PassPhrase ('enc\_pp') program is used to encrypt the PassPhrase for EncPassPhrase keyword.

Syntax:

**enc\_pp plain\_text\_PassPhrase**

Where :

- plain\_text\_PassPhrase is the user's PassPhrase to be encrypted

The plain text PassPhrase can be one of three sizes: 16, 24 or 32 characters/digits in length. No spaces. *The user MUST make sure that the PassPhrase length matches the KeySize, otherwise, MQCE will discard the EncPassPhrase if there is a mismatch.*

KeySize	PassPhrase length
128	16 characters
192	24 characters
256	32 characters

The enc\_pp program outputs the encrypted PassPhrase to the user's screen as follows:

**Encrypted PassPhrase: jXzFNIKKwZ52wsQ3CUwqWUBpDaoVRDnLMDkNqhVEOcsWMA**

Copy the 46 characters beginning and place them in the IniFile for the EncPassPhrase keyword.

e.g.

**EncPassPhrase=jXzFNIKKwZ52wsQ3CUwqWUBpDaoVRDnLMDkNqhVEOcsWMA**

### 9.1 Examples

#### 9.1.1 Windows

To use the *enc\_pp* program on Windows, open a Command prompt and change the directory to **C:\CapitaWare\MQCE\**

```
enc_pp.exe QPriiTJmr4j7aQ2P
```

### 9.1.2 Linux 32-bit

To use the *enc\_pp* program on Linux for MQ 32-bit, open a shell prompt and change directory to [/var/mqm/exits/](#)

```
enc_pp QPriiTJmr4j7aQ2P
```

### 9.1.3 Unix or Linux 64-bit

To use the *enc\_pp* program on Unix/Linux for MQ 64-bit, open a shell prompt and change directory to [/var/mqm/exits64/](#)

```
enc_pp QPriiTJmr4j7aQ2P
```

### 9.1.4 IBM i

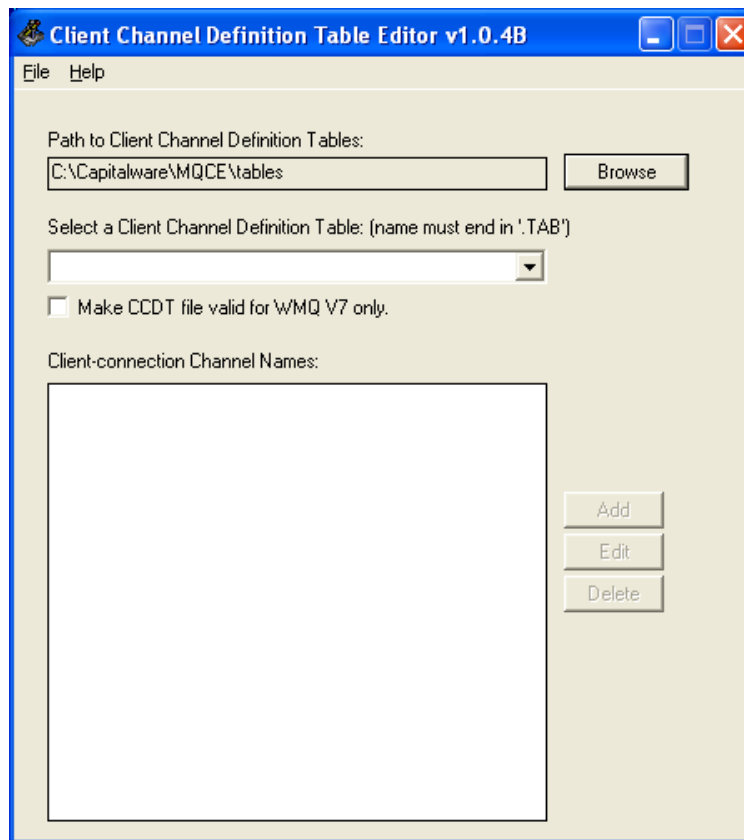
To use the *enc\_server* program on IBM i for MQ, open a shell prompt (**QSH**) and change directory to [/QIBM/UserData/mqm/mqce/](#)

```
enc_pp QPriiTJmr4j7aQ2P
```

## 10 Appendix D - Client Channel Definition Table Editor

MQCE client-side channel exit installation package includes a new tool called: *Client Channel Definition Table Editor*. The Client Channel Definition Table Editor is a Windows GUI program that enables the user to quickly create a Client Channel Definition Table or to edit an existing table in order to add, update and delete CLNTCONN channels.

The Client Channel Definition Table Editor does not require IBM MQ Server or IBM MQ Client to be installed on the PC. The Client Channel Definition Table Editor uses SupportPac MO72 to perform the adding, updating and deleting of CLNTCONN channels of an MQ client channel definition table.



- To start the Client Channel Definition Table Editor, click **Start -> All Programs -> MQ Channel Encryption -> Client Channel Definition Table Editor**
- Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name **MUST** end in '.tab')
- Click the **Add** button to insert a new CLNTCONN channel or click the **Edit** button to edit an existing CLNTCONN channel.

For the *Send/Receive Exit Name* field, the user can input their own data or use 1 of the 4 predefined values as shown below:

Values	Description
C:\Capitalware\MQCE\mqce(CE)	Use this value for native Windows applications.
C:\Capitalware\MQCE\mqcedn.dll(Capitalware.MQCEDN)	Use this value for .NET Windows applications.
/var/mqm/exits64/mqce(CE)	Use this value for native Unix/Linux 64-bit applications.
/var/mqm/exits/mqce(CE)	Use this value for native Unix/Linux 32-bit applications.

For the *Send/Receive Exit Data* field, the user can input their own data or use 1 of the 4 predefined values as shown below:

Values	Description
C:\Capitalware\MQCE\mqce.ini	Use this value for native Windows applications.
/var/mqm/exits64/mqce.ini	Use this value for native Unix/Linux 64-bit applications.
/var/mqm/exits/mqce.ini	Use this value for native Unix/Linux 32-bit applications.
K=256	Explicitly use KeySize 256-bit

A client channel definition table will be created in the 'tables' directory under the default install directory. For the example above, a client channel definition table will be found (assuming a default install) at this location:

**C:\Capitalware\MQCE\tables\MQW1.TAB**



## 11 Appendix E – Capitalware Product Display Version

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

**cwdspver**

### 11.1 Examples

#### 11.1.1 Windows

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

```
cwdspver.exe
```

#### 11.1.2 Linux 32-bit

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

```
./cwdspver
```

#### 11.1.3 Unix and Linux 64-bit

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

```
./cwdspver
```

#### 11.1.4 IBM i

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

```
CALL MQCE/CWDSPVER
```

## 12 Appendix F - Explicitly Setting Values in MSGDATA, RCVDATA & SENDDATA

An alternative to setting values in an IniFile, the user can explicitly set information in channel's MSGDATA, RCVDATA & SENDDATA fields. There are 4 values that can be explicitly set.

- **P** (Perform) indicates what functionality that MQCE will perform. Perform supports 3 values [S / E / B].
- **K** (Key Size) specifies the AES key size used for the encryption / decryption of the message data. Valid values are 128, 192 or 256.
- **D** (Debug Logging) sets the LogMode value to 'D'.
- **L** (License) specifies the queue manager's license key (server-side only)

```
DEFINE CHANNEL( 'MQB1.TO.MQA1' ) CHLTYPE(SDR) +  
  TRPTYPE( TCP ) +  
  XMITQ( 'MQB1.XMIT' ) +  
  CONNAME( '127.0.0.1(1415)' ) +  
  MSGEXIT( '/var/mqm/exits64/mqce(CE)' ) +  
  MSGDATA( 'P=E;K=256' ) +  
  REPLACE
```

Note: Separate each value with a semicolon (;).

## 13 Appendix G - Encryption and Digital Signature

MQ Channel Encryption Solution uses the Advanced Encryption Standard (AES) to encrypt the message data, which flows between IBM MQ resources. For the digital signature, MQCE uses the SHA-2 to create a cryptographic hash function for the message data..

### *Wikipedia*

*the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor,[3] the Data Encryption Standard (DES).*

*AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable (see Advanced Encryption Standard process for more details). It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce. It is available in many different encryption packages. AES is the first publicly accessible and open cipher approved by the NSA for top secret information*

### *Wikipedia*

*SHA-2 is a set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, SHA-512) designed by the National Security Agency (NSA) and published in 2001 by the NIST as a U.S. Federal Information Processing Standard. SHA stands for **Secure Hash Algorithm**. SHA-2 includes a significant number of changes from its predecessor, SHA-1. SHA-2 consists of a set of four hash functions with digests that are 224, 256, 384 or 512 bits.*

## **14 Appendix H - Support**

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

**By email at:**

support@capitalware.com

**By regular mail at:**

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

## 15 Appendix I - Summary of Changes

- MQ Channel Encryption v3.4.0
  - Enhanced the code for dumping the pointers passed into exit.
  - Fixed an issue in the subroutine that removes trailing blanks
  - Fixed issue when an invalid or expired license key is used
  - Fixed an issue with default exit path
- MQ Channel Encryption v3.3.0
  - Tuned the code that is called on entry
  - Tuned the logging code
- MQ Channel Encryption v3.2.0
  - Added EncPassPhrase keyword to support the use of encrypted PassPhrase.
  - Added 'enc\_pp' program that will create an encrypted PassPhrase.
  - Addition debug logging information added
  - Fixed an issue in the logging framework where a constant was being modified.
- MQ Channel Encryption v3.1.3
  - Addition debug logging information added
- MQ Channel Encryption v3.1.2
  - Enhanced logging - the LogFile keyword now supports the following tokens:  
%QM%, %CHL%, %UID%, %PID% & %TID%
- MQ Channel Encryption v3.1.1
  - Added check for flags in the CWME header
  - Fixed an issue on Windows with freeing environment variable memory (error with FreeEnvironmentStrings Windows API call)
  - Fixed an issue with using "size\_t" variable type when it should have been "int"
- MQ Channel Encryption v3.1.0
  - Added keyword 'AllSegments' to signal that all segments are to be encrypted and/or signed.
  - Added extra bounds check for incoming message
  - Removed an extra buffer, hence, reduced memory allocation
  - Improved the IniFile processing speed.
  - Fixed an issue with Enterprise License key not being loaded from a License file.
  - Tested with MQ v8.0
  - Tested with Windows 8/8.1
- MQ Channel Encryption v3.0.0
  - Added support non-default install for MQ v7.1 & higher multi-install feature on Linux, Unix and Windows
  - Tested with MQ v7.5

- Altered MQCE header because of incorrect exit data length received from MQ - PMR 58128
  - Increased the accepted IniFile parameter length from 1024 to 2048 characters
  - New 64-bit native DLL (64\mqce.dll) exit for 64-bit client applications
  - New MQCE .NET DLL (mqcedn.dll) exit for 32-bit .NET client applications
  - New MQCE .NET DLL (64\mqcedn.dll) exit for 64-bit .NET client applications
  - New MQCE Programming Guide manual
  - Fixed a bug in the in-memory Ini parser
  - Fixed an issue with BackupLogFileCount
- MQ Channel Encryption v2.0.0
    - Added support for digital signature SHA-2.
    - Added program **cwdspper** to display the product version number
  - MQ Channel Encryption v1.0.0
    - Initial release.

## 16 Appendix J - 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 mqce.ini, mqce\_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.



## 17 Appendix K - 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.