

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



Capitalware Inc.  
1673 Richmond Street, Suite 524  
London, Ontario N6G2N3  
Canada  
sales@capitalware.biz  
<http://www.capitalware.biz>



# 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 WebSphere MQ.....	5
<b>2 INSTALLING MQ CHANNEL ENCRYPTION.....</b>	<b>6</b>
2.1 SERVER-SIDE SECURITY EXIT.....	6
2.1.1 Windows Installation.....	6
2.1.2 Unix and 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
<b>3 CONFIGURING QMGR TO QMGR CHANNELS.....</b>	<b>11</b>
3.1 MESSAGE EXIT DATA (MSGDATA).....	12
3.1.1 Absolute Path.....	12
3.1.2 Relative Path.....	12
3.1.3 MQCE_HOME Environment Variable.....	13
3.2 SENDER CHANNEL.....	14
3.2.1 Windows .....	14
3.2.2 Unix and Linux 32-bit.....	14
3.2.3 Unix and Linux 64-bit.....	15
3.2.4 IBM i.....	15
3.3 RECEIVER CHANNEL.....	16
3.3.1 Windows .....	16
3.3.2 Unix and Linux 32-bit.....	16
3.3.3 Unix and Linux 64-bit.....	17
3.3.4 IBM i.....	17
3.4 SERVER CHANNEL.....	18
3.4.1 Windows .....	18
3.4.2 Unix and Linux 32-bit.....	18
3.4.3 Unix and Linux 64-bit.....	19
3.4.4 IBM i.....	19
3.5 REQUESTOR CHANNEL.....	20
3.5.1 Windows .....	20
3.5.2 Unix and Linux 32-bit.....	20
3.5.3 Unix and Linux 64-bit.....	21
3.5.4 IBM i.....	21
3.6 CLUSTER SENDER CHANNEL.....	22

3.6.1	Windows .....	22
3.6.2	Unix and Linux 32-bit.....	22
3.6.3	Unix and Linux 64-bit.....	23
3.6.4	IBM i.....	23
3.7	CLUSTER RECEIVER CHANNEL.....	24
3.7.1	Windows .....	24
3.7.2	Unix and Linux 32-bit.....	24
3.7.3	Unix and Linux 64-bit.....	25
3.7.4	IBM i.....	25
<b>4</b>	<b>CONFIGURING SVRCONN AND CLNTCONN CHANNELS.....</b>	<b>26</b>
4.1	USER DATA (SENDDATA AND RCVDATA).....	27
4.1.1	Absolute Path.....	27
4.1.2	Relative Path.....	27
4.1.3	MQCE_HOME Environment Variable.....	28
4.2	SERVER CONNECTION CHANNEL.....	29
4.2.1	Windows .....	29
4.2.2	Unix and Linux 32-bit.....	29
4.2.3	Unix and Linux 64-bit.....	30
4.2.4	IBM i.....	30
4.3	CLIENT CONNECTION CHANNEL.....	31
4.3.1	Windows .....	31
4.3.2	Unix and Linux 32-bit.....	31
4.3.3	Unix and Linux 64-bit.....	32
4.3.4	IBM i.....	32
4.4	CONFIGURING SEND/RECEIVE EXIT IN MQ EXPLORER.....	33
4.4.1	Local One Time Setup.....	33
4.4.2	Creating a Client Channel Definition Table Entry.....	33
4.5	JAVA BASED APPLICATIONS.....	35
4.5.1	Java Code Samples.....	35
4.5.2	Java Run-Time Settings .....	35
4.6	CONFIGURING MQCEJ FOR USE IN J2EE APPLICATION SERVER.....	36
4.6.1	Batch or Quiet mode for J2EE based applications.....	36
<b>5</b>	<b>INIFILE KEYWORDS.....</b>	<b>38</b>
5.1	LOGGING.....	38
5.2	KEYSIZE.....	39
5.3	PERFORM.....	39
5.4	PASSPHRASE.....	40
5.5	LICENSEFILE.....	41
5.6	LICENSE KEY.....	41
<b>6</b>	<b>MISCELLANEOUS.....</b>	<b>42</b>
6.1	WINDOWS.....	42
6.2	UNIX AND LINUX.....	43
6.3	IBM I.....	43
6.4	SERVER-SIDE LOG FILE.....	43

<b>7 APPENDIX A – MQCE.INI FILE.....</b>	<b>44</b>
<b>8 APPENDIX B – MQCE UPGRADE PROCEDURES.....</b>	<b>47</b>
<i>8.1.1 Windows Upgrade.....</i>	<i>47</i>
<i>8.1.2 Unix and Linux 32-bit Upgrade.....</i>	<i>47</i>
<i>8.1.3 Unix and Linux 64-bit Upgrade.....</i>	<i>48</i>
<i>8.1.4 IBM i Upgrade.....</i>	<i>48</i>
<b>9 APPENDIX C - CLIENT CHANNEL DEFINITION TABLE EDITOR.....</b>	<b>49</b>
<b>10 APPENDIX D – ENCRYPTION AND DIGITAL SIGNATURE.....</b>	<b>51</b>
<b>11 APPENDIX E – SUPPORT.....</b>	<b>52</b>
<b>12 APPENDIX F – SUMMARY OF CHANGES.....</b>	<b>53</b>
<b>13 APPENDIX G – LICENSE AGREEMENT.....</b>	<b>54</b>
<b>14 APPENDIX H – NOTICES.....</b>	<b>56</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 WebSphere MQ (WMQ) resources. MQCE operates with WMQ v5.3, v6.0 and v7.0 (and MQSeries v5.2) in Windows, Unix, IBM i (OS/400) and Linux environments. It operates with Sender, Receiver, Server, Requestor, Cluster-Sender, Cluster-Receiver, Server Connection and Client Connection channels of the WMQ queue managers.

MQCE is a simple drop-in solution that provides cryptographic protection for WMQ 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 WMQ client application and queue manager will be encrypted.

The MQCE can be configured as a queue manager channel message exit or as a channel sender/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.

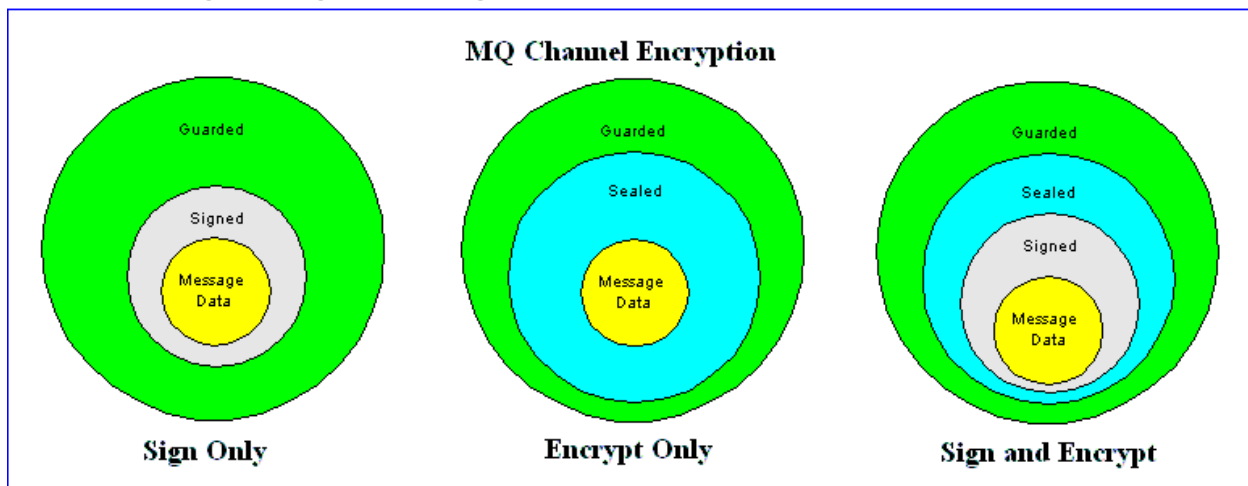
## 1.2 Executive Summary

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.

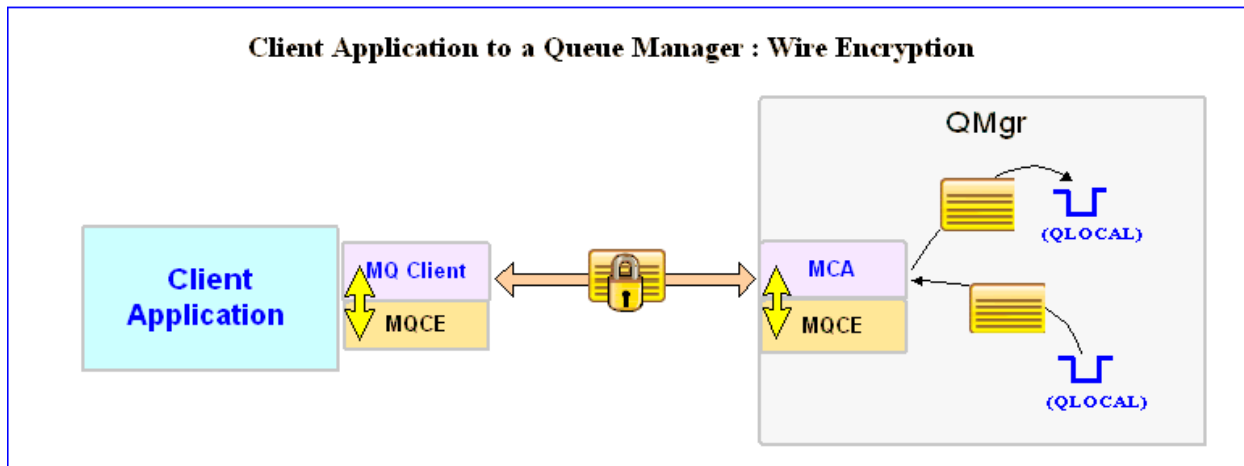
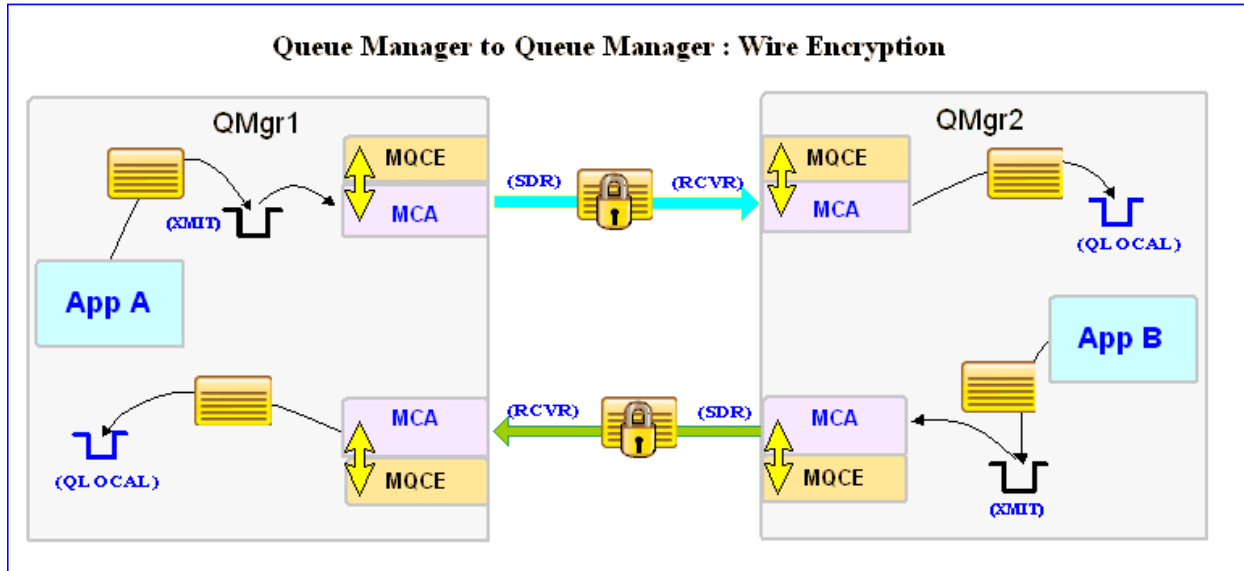
Major Features of MQCE:

- 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 5L version 5.1 or higher

#### 1.5.1.2 HP-UX IA64

- HP-UX v11.23 or higher

#### 1.5.1.3 HP-UX PA-RISC

- HP-UX v11.00 or higher

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

- IBM i V5R3 or higher

#### 1.5.1.5 Linux x86

- Linux kernel, version 2.4
- glibc version 2.2.5 or greater

Sample distributions:

- Red Hat Enterprise Linux v4, v5
- SUSE Linux Enterprise Server v9, v10, v11

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

Sample distributions:

- Red Hat Enterprise Linux v4, v5
- SUSE Linux Enterprise Server v9, v10, v11

#### 1.5.1.7 Linux on POWER

Sample distributions:

- Red Hat Enterprise Linux v4, v5
- SUSE Linux Enterprise Server v10, v11

#### 1.5.1.8 Linux on zSeries (32-bit)

- Linux kernel, version 2.4
- glibc version 2.2.5 or greater

Sample distributions:

- Red Hat Enterprise Linux v4, v5
- SUSE Linux Enterprise Server v10, v11

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

Sample distributions:

- Red Hat Enterprise Linux v4, v5
- SUSE Linux Enterprise Server v10, v11

### 1.5.1.10 Sun Solaris

- Solaris SPARC v8 or higher
- Solaris v10 x86\_64 (64-bit)

### 1.5.1.11 Windows

- Windows NT, 2000, 2003 or 2008 Server (32-bit)
- Windows XP Professional, Vista or 7 (32-bit)

## 1.5.2 WebSphere MQ

- WebSphere MQ v5.3 (or MQSeries v5.2)
- WebSphere MQ v6.0 and v7.0 (both 32-bit and 64-bit)

Operating System	WMQ v5.3 (or MQ 5.2)	WMQ v6.0 & v7.0
AIX v5.1 or higher	32-bit	64-bit
HP-UX IA64 v11.23 or higher	n/a	64-bit
HP-UX PA-RISC v11.00 or higher	32-bit	64-bit
IBM i (OS/400)	64-bit	64-bit
Linux x86	32-bit	32-bit
Linux x86_64	n/a	64-bit
Linux on POWER	n/a	64-bit
Linux on zSeries	32-bit	64-bit
Solaris SPARC v8 or higher	32-bit	64-bit
Solaris x86_64 v10	n/a	64-bit
Windows NT, 2000, 2003, XP Pro, Vista & 7	32-bit	32-bit

## 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 Server-side Security Exit

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

- **mqce.dll** is the actual encryption exit DLL for Windows that will be invoked by the MQ Server component.
- **mqce** is the actual encryption exit shared library for Unix or Linux that will be invoked by the MQ Server component.
- **MQCE** is the actual encryption exit for IBM i that will be invoked by the MQ Server component.
- **MQCEJ.jar** is the actual 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.
- **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.
- **AddRegistryEntries.bat** is a simple Windows batch file to add Event Log registry entries.
- **mqce.reg** is the registry entries for the Windows Event Log.
- **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.
- **setce.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 run the **mqce\_setup.exe** file. Follow the on-screen instructions and the encryption exit will be installed in the **C:\Capitalware\MQCE\** directory (default installation).

The user may copy or ftp the **mqce.dll**, **mqce.ini**, **AddRegistryEntries.bat** and **mqce.reg** files from one Windows server to another Windows server.

## 2.1.2 Unix and Linux 32-bit Installation

To install the 32-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 6 TAR files in the original ZIP file:

- **Java/MQCEJ.jar**
- **Linux\_x86/mqce\_linux.tar**
  - **Linux\_x86/glibc\_2.2.5/mqce\_linux\_glibc\_2.2.5.tar** (earlier releases of Linux)

Steps to install the server-side encryption exit:

1. ftp or copy the selected TAR file to the target platform to the */var/mqm/exits/* 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/exits/  
tar -xvf mqce_XXX.tar
```

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

```
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:

- **AIX/64-bit/mqce\_aix53\_64.tar**
- **AIX/64-bit/mqce\_aix61\_64.tar**
- **HPUX\_PA-RISC/64-bit/mqce\_hpux64.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\_solaris64.tar**
- **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/exits64/* directory.
2. Un-tar the mqce\_XXX.tar file into the */var/mqm/exits64/* sub-directory (XXX is either aix, hpux, solaris or linux)

```
cd /var/mqm/exits64/  
tar -xvf mqce_XXX64.tar
```

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

```
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 security exit.
- **mqce\_iseries.tar** is the IBM i IFS TAR file that contains a sample initialization file for the server-side security exit and sample MQSC script to define MQ channels with the security exits.

Steps to install the server-side security 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)  
CLRSVF 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

Recently, an issue was discovered with IBM's WebSphere 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 WebSphere 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\WMQ\_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\WMQ\_V7\IZ87819**, which can be found on the MQCE CD and in the MQCE download file.

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

**C:\Capitalware\MQCE\APAR\WMQ\_V7\fix\_WMQ\_v7.bat**

### 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 Unix and Linux for WebSphere MQ v5.3 or v6.0 (32-bit), the MSGDATA attribute would be:  
**/var/mqm/exits/mqce.ini**
- For Unix and Linux for WebSphere MQ v6.0 or v7.0 (64-bit), the MSGDATA attribute would be:  
**/var/mqm/exits64/mqce.ini**
- For IBM i for WebSphere MQ v6.0 or v7.0, the MSGDATA attribute would be:  
**mqce.ini**

***Note: Message Exit Data 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 MQCE\_HOME 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/**

## 3.2 Sender Channel

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

### 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 Unix and 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 RECEIVER Channel that the encryption exit will be applied to.

#### 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 Unix and 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( '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 SERVER Channel that the encryption exit will be applied to.

### 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 Unix and 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(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 Requestor Channel

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

### 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 Unix and 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( '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.

### 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 Unix and 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(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.

### 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 Unix and 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( '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 WebSphere MQ v5.3 or v6.0 (32-bit), the SENDDATA and RCVDATA attribute fields would be:  
**/var/mqm/exits/mqce.ini**
- For Unix and Linux for WebSphere MQ v6.0 or v7.0 (64-bit), the SENDDATA and RCVDATA attribute fields would be:  
**/var/mqm/exits64/mqce.ini**
- For IBM i for WebSphere MQ v6.0 or v7.0, the SENDDATA and RCVDATA attribute fields would be:  
**mqce.ini**

***Note: Message Exit Data 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 MQCE\_HOME 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/**

## 4.2 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.

### 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 Unix and Linux 32-bit

On Unix and 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: Send / Receive Exit Data 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 Unix and Linux 32-bit

On Unix and 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 WebSphere MQ installation directory.

i.e.

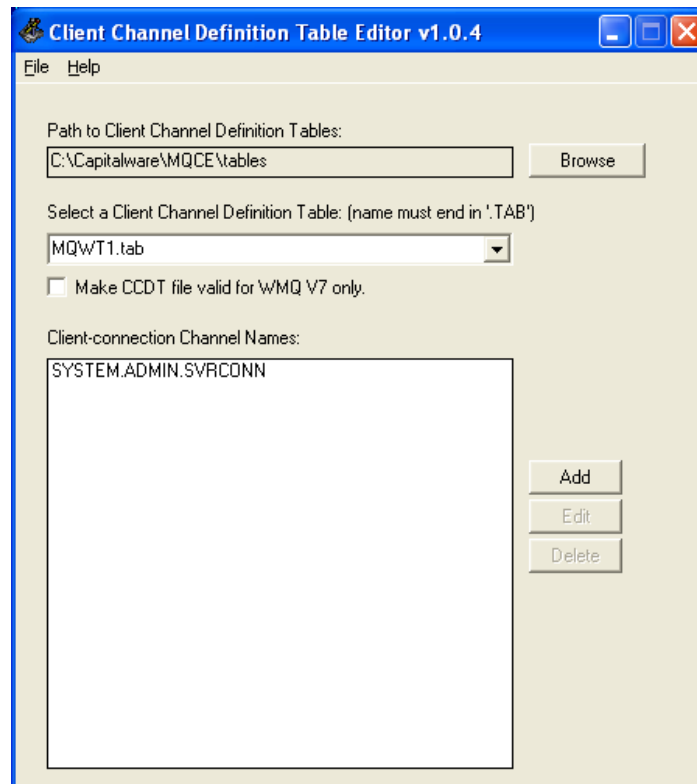
**C:\Program Files\IBM\websphere MQ\bin\runmqcfg\_rcp.cmd**

Add the following line to the runmqcfg\_rcp.cmd batch script just before the *start "WebSphere 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]

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\_LICENSE and 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 UserId and Password:

```
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 J2EE Application Server

This section describes the necessary steps to enable Security Exits in a J2EE Application Server like IBM's WebSphere Application Server or BEA's WebLogic Server.

### 4.6.1 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 License and PassPhrase values, do the following for the user-defined client-side security exit for authentication:

#### 4.6.1.1 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
```

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

Use WebSphere MQ's JMSAdmin command to define or alter a QCF (QueueConnectionFactory) or TCF (TopicConnectionFactory). The client-side security 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(tcfClient) 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(qcfClient) 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)
```

## 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 WebSphere MQ v5.3 or v6.0 (32-bit) on Unix and Linux:

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

For WebSphere MQ v6.0 or v7.0 (64-bit) on Unix and Linux:

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

For WebSphere MQ v6.0 or v7.0 on IBM i:

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

- **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 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, MQCE creates the digital signature using cryptographic hash function of SHA-2.

```
Perform=E
```

## 5.4 PassPhrase

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
- **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.

```
UsePP=Y  
PassPhrase=AeKWU31_wky6MZrL
```

## 5.5 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 WebSphere MQ v5.3, v6.0 or v7.0 (32-bit) on Unix and Linux:

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

For WebSphere MQ v6.0 or v7.0 (64-bit) on Unix and Linux:

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

For WebSphere MQ v6.0 or v7.0 on IBM i:

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

## 5.6 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*

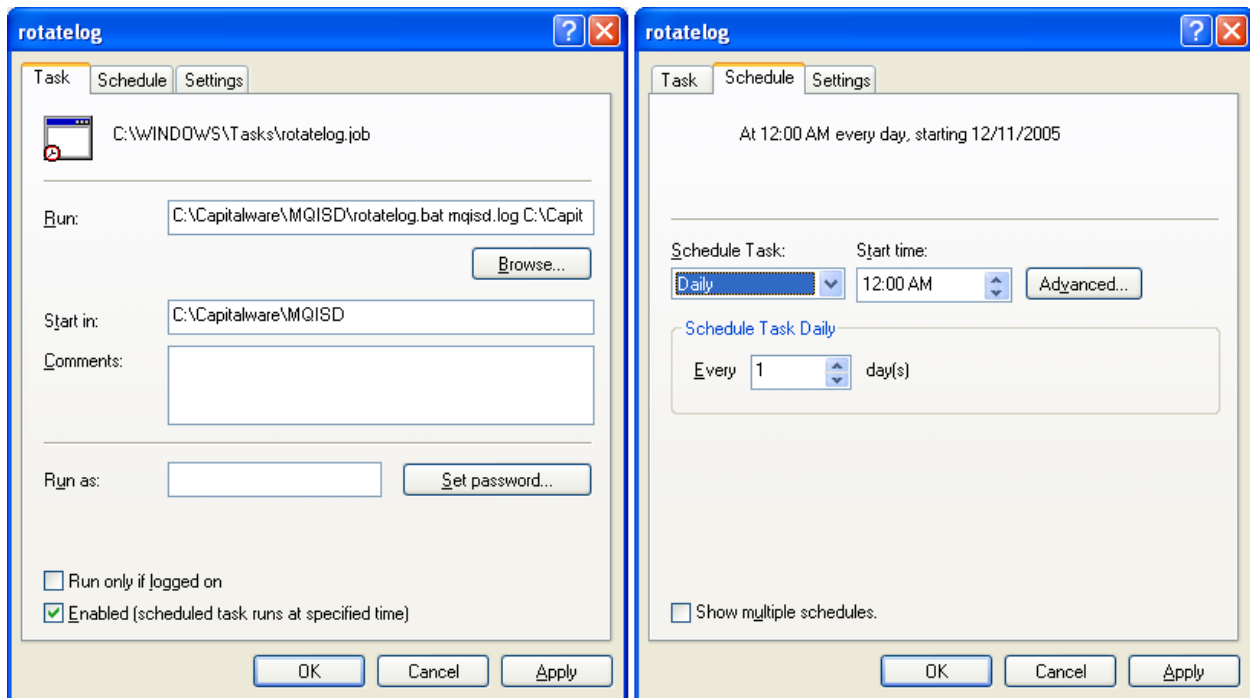
The '*mqce.ini*' file is a basic MQCE IniFile. It has the standard IniFile parameters that the user may need to use or update. The '*mqce.ini.readme*' file is a plain text help file with a description of the parameters.

#### *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.MQAI.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 WebSphere MQ v5.3 or v6.0 (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 WebSphere MQ v6.0 (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 security exit information.

## 6.4 Server-side Log File

To verify that the process flow was successful, you can view the log file for the events that are generated.

```
2007/07/23 21:55:51 MQCE #00798 I: Message encrypted for Ch1Name='MQWT1' QMgr='MQWT1.TO.MQWT2' ConName='127.0.0.1(1416)'  
2007/07/23 21:55:51 MQCE #00798 I: Message decrypted for Ch1Name='MQWT2' QMgr='MQWT1.TO.MQWT2' ConName='127.0.0.1'
```

## 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
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>
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 WebSphere MQ v5.3, v6.0 or v7.0 (32-bit) on Unix and Linux: LicenseFile=/var/mqm/exits/mqce_licenses.ini</p> <p>For WebSphere MQ v6.0 or v7.0 (64-bit) on Unix and Linux: LicenseFile=/var/mqm/exits64/mqce_licenses.ini</p> <p>For WebSphere MQ v6.0 or v7.0 on IBM i: LicenseFile=/QIBM/UserData/mqm/mqce/mqme_licenses.ini</p> <p>e.g. LicenseFile=/var/mqm/exits64/mqce_licenses.ini</p>

Keyword	Description of Server-side keywords
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 WebSphere MQ v5.3 or v6.0 (32-bit) on Unix and Linux:  LogFile=/var/mqm/exits/mqce.log</p> <p>For WebSphere MQ v6.0 or v7.0 (64-bit) on Unix and Linux:  LogFile=/var/mqm/exits64/mqce.log</p> <p>For WebSphere MQ v6.0 or v7.0 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=QPriiTJmr4j7aQ2PW</p>
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 MQME will only sign the message</li> <li>• <b>E</b> means that MQME will only encrypt the message</li> <li>• <b>B</b> means that MQME 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>

Keyword	Description of Server-side keywords
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>
WriteToSystemLog	<p><b>WriteToSystemLog</b> specifies that MQCE write a log entry to the server's 'logging system'. On Windows, the server's 'logging system' is the Event Log and on Unix / Linux it is the syslog. WriteToSystemLog supports 2 values [Y / N]. The default value is N.</p> <p>The Unix / Linux syslog output can be found for each operating system as follows:</p> <ul style="list-style-type: none"> <li>➤ AIX:            /var/log/messages</li> <li>➤ HP-UX:         /var/adm/syslog/syslog.log</li> <li>➤ Linux:          /var/log/messages</li> <li>➤ Solaris:        /var/adm/messages</li> </ul> <p>e.g. WriteToSystemLog =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
  - AddRegistryEntries.bat
  - mqce.reg
  - rotatelog.bat
- Run AddRegistryEntries.bat batch file
- Restore the MQCE IniFiles if they were altered / deleted.
- Start all of the channels using the MQCE server-side security exit or restart the queue manager if it was previously stopped.

### 8.1.2 Unix and 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:  
**./setce.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:  
**./setce.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:

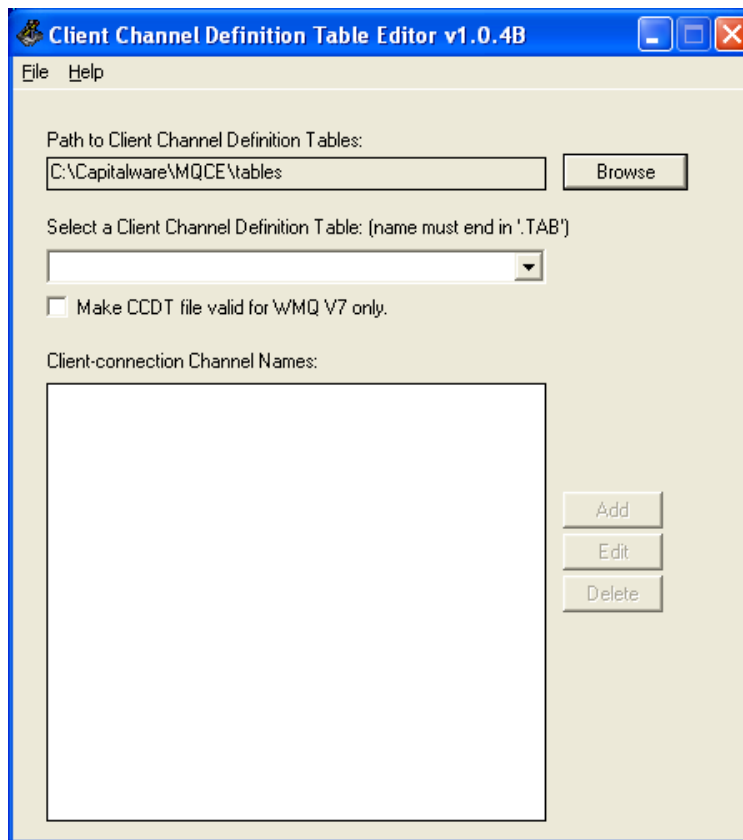
```
RSTLIB SAVLIB(MQCE) DEV(*SAVF) SAVF(QGPL/MQCE)  
CLRSVF FILE(QGPL/MQCE)  
CHGOBJOWN OBJ(MQCE) OBJTYPE(*LIB) NEWOWN(QMQM)  
qsh  
cd /QIBM/UserData/mqm/  
tar -xvf mqce_iseriess.tar  
chown -R QMQM mqce  
rm mqce_iseriess.tar
```

- 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 - Client Channel Definition Table Editor

MQCE client-side security 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 WebSphere MQ Server or WebSphere 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 3 predefined values as shown below:

Values	Description
C:\Capitalware\MQCE\mqce(CE)	Use this value for native 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**

## 10 Appendix D – Encryption and Digital Signature

MQ Channel Encryption Solution uses the Advanced Encryption Standard (AES) to encrypt the message data, which flows between WebSphere MQ (WMQ) resources. For the digital signature, MQME 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.*

## **11 Appendix E – Support**

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

**Online Help Desk Ticketing System at**  
[www.capitalware.biz/phpst/](http://www.capitalware.biz/phpst/)

**By email at:**  
[support@capitalware.biz](mailto:support@capitalware.biz)

**By regular mail at:**

Capitalware Inc.  
Attn: MQCE Support  
1673 Richmond Street, Suite 524  
London, Ontario N6G2N3  
Canada

## 12 Appendix F – Summary of Changes

- MQ Channel Encryption v2.0.0
  - Added support for digital signature SHA-2.
  - Added program **cwdsprver** to display the product version number
  
- MQ Channel Encryption v1.0.0
  - Initial release.

## 13 Appendix G – 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 user, or group of users for Multiple User Licenses and Site Licenses. This Agreement requires that each user of the SOFTWARE be Licensed, either individually, or as part of a group. A Multi-User License provides for a specified number of users to use this SOFTWARE at any time. This does not provide for concurrent user Licensing. Each user of this SOFTWARE must be covered either individually, or as part of a group Multi-User 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 users only.

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 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.

## 14 Appendix H – Notices

### Trademarks:

AIX, IBM, MQSeries, OS/2 Warp, OS/400, iSeries, MVS, OS/390, WebSphere, WebSphere 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.