# MQ Channel Throttler 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.

# Table of Contents

# 1 Introduction

## 1.1 Overview

***MQ Channel Throttler*** (MQCT) provides the ability to control/throttle the number of connection calls, open calls and/or messages (or bytes) that flow over a channel.  MQCT 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 and Server Connection channels of the MQ queue managers.

MQCT is a simple drop-in solution that provides throttling for MQ queue managers. The throttling can be configured for queue manager to queue manager channels or for client application to queue manager channels.

- Queue manager to queue manager throttling means all messages (or bytes) flowing over a channel between 2 queue managers will be throttled.

- Client application to queue manager throttling means application-level connection calls, open calls and/or messages (or bytes) flowing between a MQ client application and queue manager will be throttled.

MQCT can be used to throttle the following MQ API calls: MQCONN, MQCONNX, MQOPEN, MQGET, MQPUT and MQPUT1.

The MQCT can be configured as a channel message exit or as a channel send/receive exit.

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

Major Features of MQCT:

- Easy to set up and configure
- 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 can be throttled (controlled)
- Standard MQ feature, GET-with-Convert, is supported
- Provides high-level logging capability for throttling process

## 1.3 Context Diagram (Logical View)

## 1.4 Prerequisites

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

### 1.4.1 Operating System

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

#### 1.4.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

#### 1.4.1.2 HP-UX IA64

- HP-UX v11.23 or higher

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

- IBM i V6R1 or higher

#### 1.4.1.4 Linux x86

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

#### 1.4.1.5 Linux x86_64 (64-bit)

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

#### 1.4.1.6 Linux on POWER

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

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

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

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

- Raspberry Pi OS v9 or higher

#### 1.4.1.9 Sun Solaris

- Solaris SPARC v10 & v11
- Solaris x86_64 v10 & v11

#### 1.4.1.10 Windows

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

### 1.4.2 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 |

### 1.4.3 Windows 32-bit

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

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

### 1.4.4 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

# 2 Installing MQ Channel Throttler

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

## 2.1 Windows 32-bit Installation

To install the channel exit on Windows, first unzip the **mqct-setup-32bit.zip** and then run the mqct_setup.exe file. Follow the on-screen instructions and the channel exit will be installed in the **C:\Capitalware\MQCT\** directory (default installation).

The user may copy or ftp the mqct.dll and mqct.ini files from one Windows server to another Windows server.

## 2.2 Windows 64-bit Installation (MQ v8.0 & higher)

To install the security exit on Windows, first unzip the **mqct.zip** and then run the ***mqct-setup-64bit.exe*** file from the ***Windows-Server*** directory. Follow the on-screen instructions and the exit will be installed in the **C:\Capitalware\MQCT\** directory (default installation).

The user may copy or ftp the mqct.dll and mqct.ini files from one Windows server to another Windows server.

## 2.3 Linux 32-bit Installation

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

- **Linux_x86/mqct_linux.tar**
- **RaspberryPi_ARM/mqct_raspberrypi_arm.tar**

Steps to install the server-side channel exit:

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

   ```
   cd /var/mqm/exits/
   tar -xvf  mqct_xxx.tar
   ```

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

   ```
   chmod +x setmqct.sh
   ./setmqct.sh
   ```

## 2.4 Unix and Linux 64-bit Installation

To install the 64-bit version of MQCT on Unix or Linux, first unzip the **mqct-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/mqct_aix71_64.tar** for AIX v7.1 or higher
- **HPUX_IA64/mqct_hpux64_IA64.tar**
- **Linux_x86_64/mqct_linux_x86_64.tar**
- **Linux_POWER/mqct_linux_power64.tar**
- **Linux_zSeries/64-bit/mqct_linux_zseries64.tar**
- **Solaris_SPARC/64-bit/mqct_solaris10_64.tar** for Solaris SPARC v10 or higher
- **Solaris_x86_64/mqct_solaris_x86_64.tar**

Steps to install the server-side channel exit:

1. ftp or copy the selected TAR file to the target platform to the ***/var/mqm/exits64/*** directory.
2. Un-tar the mqct_xxx.tar file into the ***/var/mqm/exits64/*** sub-directory (xxx is either aix, hpux, solaris or linux)

   ```
   cd /var/mqm/exits64/
   tar -xvf  mqct_xxx64.tar
   ```

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

   ```
   chmod +x setmqct.sh
   ./setmqct.sh
   ```

## 2.5 IBM i Installation

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

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

Steps to install the server-side channel exit:

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

   ```
   CRTSAVF FILE(QGPL/MQCT)
   ```

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

   ```
   ftp -s:mqct_iseries.ftp  iseries_hostname
   ```

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

   binary
   cd QGPL
   put mqct.savf

   quote SITE NAMEFMT 1

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

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

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

### 2.5.1 MQCT-GUI Installation

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

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

#### 2.5.1.1 MQCT-GUI Installation on Windows

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

#### 2.5.1.2 MQCT-GUI Installation on Unix, Linux or macOS

To install MQCT-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 mqctgui.zip
```

# 3 Configuring QMgr to QMgr Channels

This section describes how to configure the channel exit for queue manager to queue manager channels.

| Platform | | Directory | Exit Module Name |
|---|---|---|---|
| Windows | 32-bit<br>64-bit | C:\Capitalware\MQCT\ | mqct.dll |
| Linux/Unix | 32-bit | /var/mqm/exits/ | mqct |
| Linux/Unix | 64-bit | /var/mqm/exits64/ | mqct |

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

## 3.1 Message Exit Data - Directly Setting Configuration Values

If the user wishes to not use an IniFile then the configuration parameters can be specified in the MSGDATA attribute field.

Note: Once a channel is started, if the MQAdmin changes the value in the ***MSGDATA*** field, MQ will not provide the updated value to the exit until the channel is stopped and restarted. Hence, if the user wants to use the dynamic features of MQCT then they need to use an IniFile.

- 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.

**C=20/s**

Where '20/s' says to throttle 20 connection calls per second.

**O=30/s**

Where '30/s' says to throttle 30 open calls per second.

**M=10/s**

Where '10/s' says to throttle 10 messages per second.

**B=200MB/m**

Where '200/m' says to throttle 200MB bytes per minute.

**A=C**

Where 'C' is value for the APIType keyword. It supports 4 values [C/O/G/P]. Separate each value with a comma. i.e. A=C,O,P,G

**D=Y**

Where 'Y' enables LogMode of Debug.

Note: Use a semicolon ';' to separate the MQCT parameters in the MSGDATA attribute field.

Alternatively, the MQAdmin can specify an IniFile in the MSGDATA attribute.

- For Windows, the MSGDATA attribute would be:
  `C:\Capitalware\MQCT\mqct.ini`

- For IBM MQ 32-bit on Linux, the MSGDATA attribute would be:
  `/var/mqm/exits/mqct.ini`

- For IBM MQ 64-bit on Unix and Linux, the MSGDATA attribute would be:
  `/var/mqm/exits64/mqct.ini`

- For IBM MQ on IBM i, the MSGDATA attribute would be:
  `mqct.ini`

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

## 3.2 Message Exit Data - IniFile

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

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

### 3.2.1 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\MQCT\mqct(CT)') +
       MSGDATA('C:\Capitalware\MQCT\mqct.ini') +
       REPLACE
```

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

### 3.2.2 Relative Path

Relative pathing for the MSGDATA is supported on Linux, IBM i, Unix and Windows platforms. MQCT 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/mqct(CT)') +
       MSGDATA('mqct.ini') +
       REPLACE
```

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

**`/var/mqm/exits/`**

## 3.2.3 Environment Variables

### 3.2.3.1 Global Environment Variable

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

e.g. Unix

```
export MQCT_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/mqct(CT)') +
       MSGDATA('mqct.ini') +
       REPLACE
```

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

### 3.2.3.2 Queue Manager Specific Environment Variable

MQCT supports the use of the MQCT_HOME environment variable post-fixed with the queue manager name which holds the directory path information. MQCT_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 MQCT_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/mqct(CT)') +
       MSGDATA('mqct.ini') +
       REPLACE
```

Hence, MQCT 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.***

## 3.3 Sender Channel

This section describes the necessary entries to enable the channel exit. The MQAdministrator will need to update 2 fields of the SENDER Channel that the channel 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\MQCT\mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       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).

  `MQCT      MQCT`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

## 3.4 Receiver Channel

This section describes the necessary entries to enable the channel exit. The MQAdministrator will need to update 2 fields of the RECEIVER Channel that the channel 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\MQCT\mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RCVR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RCVR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RCVR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       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).

  `MQCT      MQCT`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RCVR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

## 3.5 Server Channel

This section describes the necessary entries to enable the channel exit. The MQAdministrator will need to update 2 fields of the SERVER Channel that the channel 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\MQCT\mqct(CT)`**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`**

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.5.4 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).

  `MQCT      MQCT`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(SVR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

## 3.6 Requester Channel

This section describes the necessary entries to enable the channel exit. The MQAdministrator will need to update 2 fields of the REQUESTER Channel that the channel 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\MQCT\mqct(CT)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RQSTR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RQSTR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)**

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RQSTR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.6.4 IBM i

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

- MSGEXIT

  ```
  MQCT      MQCT
  ```

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE( RQSTR ) +
       TRPTYPE( TCP ) +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

## 3.7 Cluster Sender Channel

This section describes the necessary entries to enable the channel exit. The MQAdministrator will need to update 2 fields of the CLUSSDR Channel that the channel 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\MQCT\mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       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/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.7.4 IBM i

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

- MSGEXIT

  `MQCT      MQCT`

```
DEFINE CHANNEL ('MQA1.TO.MQB1') CHLTYPE(CLUSSDR) +
       TRPTYPE(TCP) +
       XMITQ( 'MQB1.XMIT' ) +
       CONNAME( '127.0.0.1(1415)') +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

## 3.8 Cluster Receiver Channel

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

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

### 3.8.1 Windows

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

- MSGEXIT
  `C:\Capitalware\MQCT\mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE(CLUSRCVR) +
       TRPTYPE( TCP ) +
       MSGEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.8.2 Linux 32-bit

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

- MSGEXIT
  `/var/mqm/exits/mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE(CLUSRCVR) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits/mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.8.3 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/mqct(CT)`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE(CLUSRCVR) +
       TRPTYPE( TCP ) +
       MSGEXIT('/var/mqm/exits64/mqct(CT)') +
       MSGDATA('M=20/s') +
       REPLACE
```

### 3.8.4 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).

  `MQCT      MQCT`

```
DEFINE CHANNEL( 'MQB1.TO.MQA1')  CHLTYPE(CLUSRCVR) +
       TRPTYPE( TCP ) +
       MSGEXIT('MQCT      MQCT      ') +
       MSGDATA('M=20/s') +
       REPLACE
```

# 4 Configuring Client to QMgr Channels

This section describes how to configure the channel exit for client to queue manager channels.

| Platform | | Directory | Exit Module Name |
|---|---|---|---|
| Windows | 32-bit<br>64-bit | C:\Capitalware\MQCT\ | mqct.dll |
| Linux/Unix | 32-bit | /var/mqm/exits/ | mqct |
| Linux/Unix | 64-bit | /var/mqm/exits64/ | mqct |

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

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

## 4.1 User Data - Directly Setting Configuration Values

If the user wishes to not use an IniFile then the configuration parameters can be specified in the SENDDATA and RCVDATA attribute fields.

Note: Once a channel is started, if the MQAdmin changes the value in the SENDDATA and/or RCVDATA fields, MQ will not provide the updated values to the exit until the channel is stopped and restarted. Hence, if the user wants to use the dynamic features of MQCT then they need to use an IniFile.

- SENDDATA and 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.
**C=20/s**
Where '20/s' says to throttle 20 connection calls per second.

**O=30/s**
Where '30/s' says to throttle 30 open calls per second.

**M=10/s**
Where '10/s' says to throttle 10 messages per second.

**B=200MB/m**
Where '200/m' says to throttle 200MB bytes per minute.

**A=C**
Where 'C' is value for the APIType keyword. It supports 4 values [C/O/G/P]. Separate each value with a comma. i.e. A=C,O,P,G

**D=Y**
Where 'Y' enables LogMode of Debug.

Use a semicolon ';' to separate the MQCT parameters in the SENDDATA and RCVDATA attribute fields.

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

- For Windows, the SENDDATA and RCVDATA attribute fields would be:
  `C:\Capitalware\MQCT\mqct.ini`
- For IBM MQ 32-bit on Linux, the SENDDATA and RCVDATA attribute fields would be:
  `/var/mqm/exits/mqct.ini`

- For IBM MQ 64-bit on Unix and Linux, the SENDDATA and RCVDATA attribute fields would be:
  `/var/mqm/exits64/mqct.ini`

- For IBM i, the SENDDATA and RCVDATA attribute fields would be:
  `mqct.ini`

***Note: SENDDATA and RCVDATA must NOT exceed 32 characters.***

## 4.2 User Data - IniFile

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

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

### 4.2.1 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\MQCT\mqct(CT)') +
       SENDDATA('C:\Capitalware\MQCT\mqct.ini') +
       RCVEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       RCVDATA('C:\Capitalware\MQCT\mqct.ini') +
       REPLACE
```

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

### 4.2.2 Relative Path

Relative pathing for the SENDDATA or RCVDATA is supported on Linux, IBM i, Unix and Windows platforms. MQCT 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/mqct(CT)') +
       SENDDATA('mqct.ini') +
       RCVEXIT('/var/mqm/exits/mqct(CT)') +
       RCVDATA('mqct.ini') +
       REPLACE
```

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

**`/var/mqm/exits/`**

## 4.2.3 Environment Variables

### 4.2.3.1 Global Environment Variable

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

e.g. Unix

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

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

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

### 4.2.3.2 Queue Manager Specific Environment Variable

MQCT supports the use of the MQCT_HOME environment variable post-fixed with the queue manager name which holds the directory path information. MQCT_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 MQCT_HOME_MQA1=/really/long/path/MQHA/QMgrName/data2/
```

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

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

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

## 4.3 SVRCONN Channel

This section describes the necessary entries to enable the server-side channel exit. The MQAdministrator will need to update 2 fields of the SVRCONN Channel that the server-side channel 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\MQCT\mqct(CT)`
- RCVEXIT
  `C:\Capitalware\MQCT\mqct(CT)`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SENDEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       SENDDATA('M=20/s') +
       RCVEXIT('C:\Capitalware\MQCT\mqct(CT)') +
       RCVDATA('M=20/s') +
       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/mqct(CT)`
- RCVEXIT
  `/var/mqm/exits/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SENDEXIT('/var/mqm/exits/mqct(CT)') +
       SENDDATA('M=20/s') +
       RCVEXIT('/var/mqm/exits/mqct(CT)') +
       RCVDATA('M=20/s') +
       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/mqct(CT)`
- RCVEXIT
  `/var/mqm/exits64/mqct(CT)`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SENDEXIT('/var/mqm/exits64/mqct(CT)') +
       SENDDATA('M=20/s') +
       RCVEXIT('/var/mqm/exits64/mqct(CT)') +
       RCVDATA('M=20/s') +
       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).

  `MQCT      MQCT`

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

  `MQCT      MQCT`

```
DEFINE CHANNEL ('MQA1.APP.CH01') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SENDEXIT('MQCT      MQCT      ') +
       SENDDATA('M=20/s') +
       RCVEXIT('MQCT      MQCT      ') +
       RCVDATA('M=20/s') +
       REPLACE
```

# 5 IniFile Keywords

## 5.1 Logging

This section describes the necessary entries to enable MQCT to write log information. To enable and control logging, you need 6 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\MQCT\mqct.log
  ```

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

  For IBM MQ 64-bit on Unix and Linux:
  ```
  LogFile=/var/mqm/exits64/mqct.log
  ```

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

- **WriteToSystemLog** specifies whether or not MQCT will 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. Setting ' WriteToSystemLog' to 'Y' (Yes) will activate this feature; otherwise, there will be no logging to the server's logging system. The default value is N.

  The location of the Unix / Linux syslog output can be found for each operating system as follows:

  | | |
  |---|---|
  | AIX: | /var/log/messages |
  | HP-UX: | /var/adm/syslog/syslog.log |
  | Linux: | /var/log/messages |
  | Solaris: | /var/adm/messages |

  The Windows Event log can be viewed using the Windows Event Viewer.

  Note: If the user did not install MQCT server-side security exit using the windows installer then the user will need to run the AddRegistryEntries.bat file to add the registry entries in order to view the MQCT log messages in the Event Viewer.

- **WriteToEventQueue** specifies whether or not MQCT will write an event message containing the log entry information to the event queue. The default value is N.

  WriteToEventQueue provides the ability to write custom MQ Events to System Channel Event Queue to allow MQCT to be tied into an MQ Monitoring tool.

    - 9501 for channel 'sleep' event message

- **EventQueueName** specifies the event queue name. The default value is 'SYSTEM.ADMIN.CHANNEL.EVENT'.

```
LogMode=N
LogFile=C:\Capitalware\MQCT\mqct.log
RotateLogDaily=Y
WriteToSystemLog=Y
WriteToEventQueue=Y
EventQueueName=SYSTEM.ADMIN.CHANNEL.EVENT
```

## 5.2 Active

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

```
Active=Y
```

## 5.3 APIType

APIType specifies the throttling to be against MQCONN, MQCONNX, MQOPEN, MQGET, MQPUT and/or MQPUT1. APIType supports 4 values [ C / O / G / P]. The default value is 'G,P'.

- **C** means that MQCT will throttle for MQCONN and MQCONNX calls
- **O** means that MQCT will throttle for MQOPEN call
- **G** means that MQCT will throttle for MQGET call
- **P** means that MQCT will only throttle for MQPUT and MQPUT1 calls

Separate each value with a comma.

```
APIType=C,O,G,P
```

## 5.4 ConnectionRate

ConnectionRate specifies the maximum number of connection calls to be allowed over the channel in a given period of time.

- **25/s** means that MQCT will allow 25 connections over the channel per second
- **50/m** means that MQCT will allow 50 connections over the channel per minute

Note: Only used if APIType is set to C.

```
ConnectionRate=25/s
```

## 5.5 OpenRate

OpenRate specifies the maximum number of open calls to be allowed over the channel in a given period of time.

- **30/s** means that MQCT will allow 30 open calls over the channel per second
- **60/m** means that MQCT will allow 60 open calls over the channel per minute

Note: Only used if APIType is set to O.

```
OpenRate=30/s
```

## 5.6 ByteRate

ByteRate specifies the maximum number of bytes to be allowed over the channel in a given period of time.

- **25KB/s** means that MQCT will allow 25 Kilobytes of data over the channel per second
- **40MB/m** means that MQCT will allow 40 Megabytes of data over the channel per minute

Note: Only used if APIType is set to P and/or G. Also, if ByteRate is used then MessageRate is ignored.

```
ByteRate=100MB/m
```

## 5.7 MessageRate

MessageRate specifies the maximum number of messages to be allowed over the channel in a given period of time.

- **55/s** means that MQCT will allow 55 messages over the channel per second
- **100/m** means that MQCT will allow 100 messages over the channel per minute

Note: Only used if APIType is set to P and/or G. Also, if ByteRate is used then MessageRate is ignored.

```
MessageRate=55/s
```

## 5.8 UseCombinedGetPutRates

UseCombinedGetPutRates specifies whether or not the throttles should be based on the combined total of gets and puts. UseCombinedGetPutRates supports 2 values [Y / N]. The default value is Y.

Note: Only used if APIType is set to P and/or G.

```
UseCombinedGetPutRates=Y
```

## 5.9 LicenseFile

This section will describe how to have a file that contains all of the user's MQCT 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\MQCT\mqct_licenses.ini
```

For IBM MQ 32-bit on Unix and Linux:

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

For IBM MQ 64-bit on Unix and Linux:

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

For IBM MQ on IBM i:

```
LicenseFile=/QIBM/UserData/mqm/mqct/mqct_licenses.ini
```

## 5.10 License Key

This section will describe how to license MQ Channel Throttler 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 MQCT up and running in a very quick manner.

## 6.1 Windows

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

***Sample MQSC scripts***
The '***mqct.sample.mqsc***' file is a sample MQSC script to update the 2 system defined channels with the MQCT channel exit information.

***Rotate log script***
The '***rotatelog.bat***' file is a Windows batch script to rotate (backup) the mqct.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 '***mqct.ini***' file is a basic MQCT IniFile. It has the standard IniFile parameters that the user may need to use or update. The '***mqct.ini.readme***' file is a plain text help file with a description of the parameters.

***Sample MQSC scripts***
The '***mqct.sample.MQA1.mqsc***' and '***mqct.sample.MQB1.mqsc***' files are sample MQSC scripts to be used as reference channels with the MQCT channel exit.

***Rotate log script***
The '***rotatelog.sh***' file is a Unix / Linux shell script to rotate (backup) the mqct.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 mqct.log /var/mqm/exits/  > /tmp/mqct.log.run 2 > &1
```

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

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

## 6.3 IBM i

***Sample IniFile***

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

***Sample MQSC scripts***

The '***mqct.sample.mqsc***' file is a sample MQSC script to update the 2 system defined channels with the MQCT channel 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.

```
2015/04/22 15:48:42 MQCT I: [P] QMgr='MQWT1' ConName='192.168.10.115' ChlName='TEST.CHL' is delayed for 1000ms
2015/04/22 15:49:03 MQCT I: [P] QMgr='MQWT1' ConName='192.168.10.115' ChlName='TEST.CHL' is delayed for 1000ms
```

# 7 Appendix A – mqct.ini file

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

```
LogMode=N
LogFile=C:\Capitalware\MQCT\mqct.log
APIType=C,P
ConnectionRate=4/s
MessageRate=25/s
License=
```

**Note: Keywords are case sensitive**.

| Keyword | Description of Server-side keywords |
|---|---|
| **Active** | **Active** specifies if MQCT is enabled or disabled.  **Active** supports 2 values [Y / N].  The default value is Y.<br><br>e.g.<br>**Active**=N |
| APIType | **APIType** specifies the throttling to be against MQCONN, MQCONNX, MQOPEN, MQGET, MQPUT and/or MQPUT1. APIType supports 4 values [ C / O /  G / P].  The default value is 'G,P'.<br><br>• **C** means that MQCT will throttle for MQCONN and MQCONNX calls<br>• **O** means that MQCT will throttle for MQOPEN call<br>• **G** means that MQCT will throttle for MQGET call<br>• **P** means that MQCT will only throttle for MQPUT and MQPUT1 calls<br><br>Separate each value with a comma.<br>e.g.<br>APIType=C,O,G,P |
| ByteRate | **ByteRate** specifies the maximum number of bytes to be allowed over the channel in a given period of time.<br><br>e.g.<br>ByteRate=25KB/s<br><br>Note: Only used if APIType is set to P and/or G.  Also, if ByteRate is used then MessageRate is ignored. |

| Keyword | Description of Server-side keywords |
|---|---|
| ConnectionRate | **ConnectionRate** specifies the maximum number of connection calls to be allowed over the channel in a given period of time.<br>e.g.<br>ConnectionRate=35/s<br>Note: Only used if APIType is set to C. |
| EventQueueName | **EventQueueName** specifies the name of the event queue. The default value is 'SYSTEM.ADMIN.CHANNEL.EVENT'.<br>e.g.<br>EventQueueName= SYSTEM.ADMIN.CHANNEL.EVENT<br>Note: Only used if WriteToEventQueue is set to 'Y'. |
| IniFileRecheckTime | **IniFileRecheckTime** specifies the amount, in seconds, before the IniFile is checked whether it has changed or not. The default value is 60.<br>e.g.<br>**IniFileRecheckTime**=60 |
| License | **License** 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).<br>e.g.<br>License=0000-AAAA-BBBBBBBB |
| LicenseFile | **LicenseFile** specifies the location of License file that contains all of the customer's license keys.<br>The following are the default values for LicenseFile:<br>For Windows:<br>LicenseFile=C:\Capitalware\MQCT\mqct_licenses.ini<br>For IBM MQ 32-bit on Unix and Linux:<br>LicenseFile=/var/mqm/exits/mqct_licenses.ini<br>For IBM MQ 64-bit on Unix and Linux:<br>LicenseFile=/var/mqm/exits64/mqct_licenses.ini<br>For IBM MQ on IBM i:<br>LicenseFile=/QIBM/UserData/mqm/mqct/mqme_licenses.ini<br>e.g.<br>LicenseFile=/var/mqm/exits64/mqct_licenses.ini |

| Keyword | Description of Server-side keywords |
|---|---|
| LogFile | **LogFile** specifies the location of the log file. The default is as follows:<br>For Windows:<br>LogFile=C:\Capitalware\MQCT\mqct.log<br>For IBM MQ 32-bit on Unix and Linux:<br>LogFile=/var/mqm/exits/mqct.log<br>For IBM MQ 64-bit on Unix and Linux:<br>LogFile=/var/mqm/exits64/mqct.log<br>For IBM MQ on IBM i:<br>LogFile=/QIBM/UserData/mqm/mqct/mqct.log |
| LogMode | **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.<br>e.g.<br>LogMode=Q |
| MessageRate | **MessageRate** specifies the maximum number of messages to be allowed over the channel in a given period of time.<br>e.g.<br>MessageRate=55/s<br>Note: Only used if APIType is set to P and/or G. Also, if ByteRate is used then MessageRate is ignored. |
| OpenRate | **OpenRate** specifies the maximum number of open calls to be allowed over the channel in a given period of time.<br>e.g.<br>OpenRate=35/s<br>Note: Only used if APIType is set to O. |
| RotateLogDaily | **RotateLogDaily** specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y.<br>e.g.<br>RotateLogDaily=Y |

| Keyword | Description of Server-side keywords |
|---|---|
| UseCombinedGetPutRates | **UseCombinedGetPutRates** specifies whether or not the throttles should be based on the combined total of gets and puts. UseCombinedGetPutRates supports 2 values [Y / N]. The default value is Y.<br>e.g.<br>UseCombinedGetPutRates=Y |
| WriteToEventQueue | **WriteToEventQueue** specifies if MQCT will write an event message containing the log entry information to an event queue. WriteToEventQueue supports 2 values [Y / N]. The default value is N.<br>e.g.<br>WriteToSystemLog =Y |
| WriteToSystemLog | **WriteToSystemLog** specifies if MQCT 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.<br>The Unix / Linux syslog output can be found for each operating system as follows:<br>• AIX: /var/log/messages<br>• HP-UX: /var/adm/syslog/syslog.log<br>• Linux: /var/log/messages<br>• Solaris: /var/adm/messages<br>e.g.<br>WriteToSystemLog =Y |

# 8 Appendix B – MQCT Upgrade Procedures

To upgrade an existing installation of MQCT 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 MQCT exit or completely stop the queue manager.
- Backup all MQCT IniFiles in the MQCT install directory
- If MQCT was installed using the Windows Installer then
  - Click the ***Start*** -> ***All Programs*** -> ***Control Panel*** -> ***Add or Remove Programs***, select MQCT from the list and click the ***Remove*** button then follow the prompts to remove it
  - Run the ***mqct-setup.exe*** file from the ***Windows*** directory to install the new version
- Otherwise copy the following files (latest version) to the MQCT install directory:
  - mqct.dll
  - rotatelog.bat
- Restore the MQCT IniFiles if they were altered / deleted.
- Start all of the channels using the MQCT 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 MQCT exit or completely stop the queue manager.
- Backup all MQCT IniFiles in the MQCT 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  mqct_linux.tar
  ```
- Run the script as follows:

  ```
  ./setmqct.sh
  ```
- Restore the MQCT IniFiles if they were altered / deleted.
- Delete the MQCT tar file
- Start all of the channels using the MQCT 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 MQCT exit or completely stop the queue manager.
- Backup all MQCT IniFiles in the MQCT 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  mqct_aix.tar
```

- Run the script as follows:

```
./setmqct.sh
```

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

### 8.1.4 IBM i Upgrade

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

```
ftp -s:mqct_iseries.ftp  iseries_hostname
```

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

binary
cd QGPL
put mqct.savf

quote SITE NAMEFMT 1

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

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

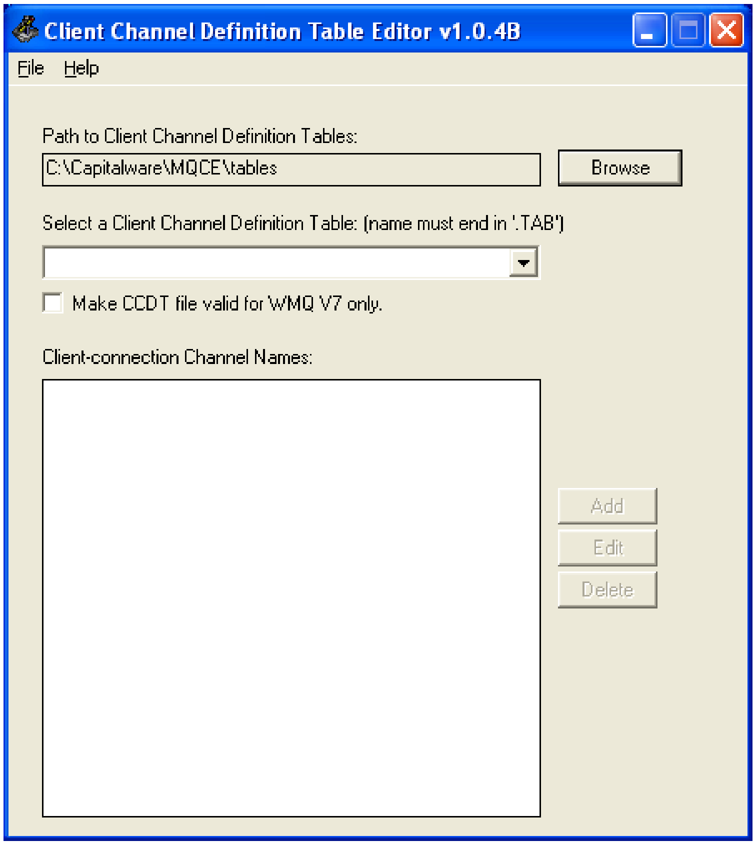
```
RSTLIB SAVLIB(MQCT) DEV(*SAVF) SAVF(QGPL/MQCT)
CLRSAVF FILE(QGPL/MQCT)
CHGOBJOWN  OBJ(MQCT) OBJTYPE(*LIB)  NEWOWN(QMQM)
qsh
cd /QIBM/UserData/mqm/
tar -xvf mqct_iseries.tar
chown -R QMQM mqct
rm mqct_iseries.tar
```

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

# 9 Appendix C – Client Channel Definition Table Editor

MQCT 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.



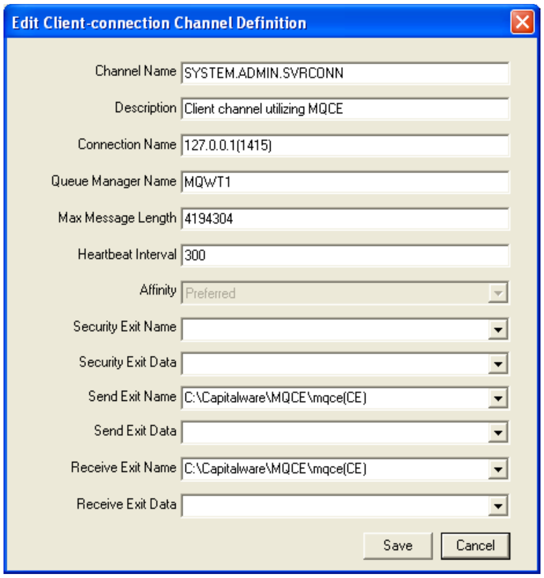
Path to Client Channel Definition Tables:
C:\Capitalware\MQCE\tables

Select a Client Channel Definition Table: (name must end in '.TAB')

□ Make CCDT file valid for WMQ V7 only.

Client-connection Channel Names:

- To start the Client Channel Definition Table Editor, click ***Start*** -> ***All Programs*** -> ***MQ Channel Throttler*** -> ***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.



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:
Security Exit Data:
Send Exit Name: C:\Capitalware\MQCE\mqce(CE)
Send Exit Data:
Receive Exit Name: C:\Capitalware\MQCE\mqce(CE)
Receive Exit Data:

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\MQCT\mqct(CT) | Use this value for native Windows applications. |
| /var/mqm/exits64/mqct(CT) | Use this value for native Unix/Linux 64-bit applications. |
| /var/mqm/exits/mqct(CT) | 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\MQCT\mqct.ini | Use this value for native Windows applications. |
| /var/mqm/exits64/mqct.ini | Use this value for native Unix/Linux 64-bit applications. |
| /var/mqm/exits/mqct.ini | Use this value for native Unix/Linux 32-bit applications. |

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\MQCT\tables\MQW1.TAB
```

# 10 Appendix D – Capitalware Product Display Version

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

`cwdspver`

## 10.1 Examples

### 10.1.1 Windows

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

```
cwdspver.exe
```

### 10.1.2 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
```

### 10.1.3 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
```

### 10.1.4 IBM i

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

```
CALL MQCT/CWDSPVER
```

# 11 Appendix E – Support

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

**By email at:**
support@capitalware.com

**By regular mail at:**

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

# 12 Appendix F – Summary of Changes

- MQ Channel Throttler v1.3.0
  - Added new keyword: UseCombinedGetPutRates - default value is 'Y'
  - 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 Throttler v1.2.0
  - Tuned the code that is called on entry
  - Tuned the logging code

- MQ Channel Throttler v1.1.0
  - Added extra checks for null pointers
  - Performed some tuning to some highly used functions
  - Fixed an issue with counting message/bytes for MQGET.
  - Added code to determine Endianness on startup.
  - Fixed an issue related to MQOD structure.
  - Fixed an issue related to MQPMO structure.
  - Fixed an issue related to MQGMO structure.
  - Fixed an issue related to MQMD structure.
  - Fixed an issue in the logging framework where a constant was being modified.

- MQ Channel Throttler v1.0.2
  - Added the Active keyword
  - Enhanced logging - the LogFile keyword now supports the following tokens: %QM%, %CHL%, %UID%, %PID% & %TID%
  - Fixed an issue with shared memory

- MQ Channel Throttler v1.0.1
  - 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 Throttler 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 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 mqct.ini, mqct_licenses.ini or any other form.

3. OTHER RESTRICTIONS. The registration notification provided, showing your 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, 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.