MQCCI 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 Connection Inspector (MQCCI) is a new solution that allows a company to track and/or audit what information a client application or remote queue manager is exchanging with the local queue manager when a connection is made.

MQCCI is comprised of an MQ Channel Security Exit. A channel security exit is ONLY invoked/called by the queue manager's MCA (Message Channel Agent) for MQCONN/X and MQDISC API calls (so it is very light-weight).

MQCCI will operate with IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix, IBM i and Linux environments. It works with Server Connection, Receiver, Requester, Sender, Server, Cluster-Sender and Cluster-Receiver channels of IBM MQ queue manager.

MQCCI is designed to provide the MQAdmin with any or all of the fields of the MQCD and MQCXP structures in a "human readable" format. Human readable implies that it will convert binary fields into their MQ defined names (i.e. SSLClientAuth=MQSCA_REQUIRED). The user can control the fields that are outputted for each of the following MQ structures: MQCD and MQCXP.

MQCCI is designed to output 1 line per API call (1 long line). The output information is written to plain text CSV (Comma Separate Value) files. The user can choose to have the output information written to a local or remote queue rather than to a file.

On AIX, HP-UX, Linux, Solaris and Windows, MQCCI 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 MQ Channel Connection Inspector solution contains a channel security exit.

The channel security exit is available in 3 forms:

- Windows DLL
- Shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The major features of MQ Channel Connection Inspector are as follows:

- Output information related to MQCONN/X and MQDISC API calls for remote connections
- MQCCI is designed to provide the user with all of the information in "human readable" format.
- Provides the output files in an easy to manage CSV (Comma Separated Value) format
- The user can select to have the CSV data written to a queue rather than a file

1.3 Prerequisites

This section provides the minimum supported software levels. These prerequisites apply to server-side installations of MQ Channel Connection Inspector.

1.3.1 Operating System

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

1.3.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.3.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.3.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.3.1.4 Linux x86

- Red Hat Enterprise Linux v5, v6, v7
- SuSE Linux Enterprise Server v10, v11, v12

1.3.1.5 Linux x86_64 (64-bit)

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

1.3.1.6 Linux on POWER

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

1.3.1.7 Linux on zSeries (64-bit)

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

1.3.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.3.1.9 Sun Solaris

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

1.3.1.10 Windows

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

1.3.2 IBM MQ

- IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 (both 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

1.3.3 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 Connection Inspector

This section describes how to install Capitalware's MQ Channel Connection Inspector.

2.1 Windows 32-bit Installation

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

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

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

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

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

2.3 Linux 32-bit Installation

To install the 32-bit version of MQCCI on Linux, first unzip the **mqcci.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/mqcci_linux.tar**
- **RaspberryPi_ARM/mqcci_raspberrypi_arm.tar**

Steps to install the channel security exit:

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

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

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

```
chmod +x setmqcci.sh
./setmqcci.sh
```

2.4 Unix and Linux 64-bit Installation

To install the 64-bit version of MQCCI on Unix or Linux, first unzip the **mqcci.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/mqcci_aix71_64.tar** for AIX v7.1 or higher
- **HPUX_IA64 /mqcci_hpux64_ia64.tar**
- **Linux_x86_64/mqcci_linux_x86_64.tar**
- **Linux_POWER/mqcci_linux_power64.tar**
- **Linux_zSeries/64-bit/mqcci_linux_zseries64.tar**
- **Solaris_SPARC/64-bit/mqcci_solaris10_64.tar** for Solaris SPARC v10 or higher
- **Solaris_x86_64/mqcci_solaris_x86_64.tar**

Steps to install the channel security exit:

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

   ```
   cd /var/mqm/exits64/
   tar -xvf  mqcci_xxx64.tar
   ```

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

   ```
   chmod +x setmqcci.sh
   ./setmqcci.sh
   ```

2.5 IBM i Installation

To install the MQCCI on IBM i, first unzip the **mqcci.zip** and then select the files in the IBM i (IBM i) directory.

- **mqcci.savf** is the IBM i 'Save File' that contains the library with the security exit.

- **mqcci_iseries.tar** is the IBM i IFS TAR file that contains a sample initialization file for the channel security exit and sample MQSC script to define MQ channels with the security exits.

Steps to install the channel security exit:

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

```
CRTSAVF FILE(QGPL/MQCCI)
```

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

```
ftp -s:mqcci_iseries.ftp  iseries_hostname
```

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

binary
cd QGPL
put mqcci.savf

quote SITE NAMEFMT 1

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

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

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

2.5.1 MQCCI-GUI Installation

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

- **MQCCI-GUI/mqccigui-wthjre.exe** (for Windows)
- **MQCCI-GUI/mqccigui.zip** (for Unix, Linux or macOS)

2.5.1.1 MQCCI-GUI Installation on Windows

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

2.5.1.2 MQCCI-GUI Installation on Unix, Linux or macOS

To install MQCCI-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 mqccigui.zip
```

3 Configuring Channel Security Exit

This section describes how to configure the channel security exit.

3.1 Security User Data (SCYDATA)

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

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

3.1.1 Absolute Path

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

E.g. Windows

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('C:\Capitalware\MQCCI\mqcci(CCI)') +
       SCYDATA('C:\Capitalware\MQCCI\mqcci.ini') +
       REPLACE
```

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

3.1.2 Relative Path

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

E.g. Unix

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits/mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

Hence, MQCCI will use the following path as the IniFile path:
`/var/mqm/exits/`

3.1.3 Environment Variables

3.1.3.1 Global Environment Variable

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

e.g. Unix

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

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

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

3.1.3.2 Queue Manager Specific Environment Variable

MQCCI supports the use of the MQCCI_HOME environment variable post-fixed with the queue manager name which holds the directory path information. MQCCI_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 MQCCI_HOME_MQA1=/really/long/path/MQHA/QMgrName/data2/
```

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

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

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

3.2 SVRCONN Channel

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

Platform		Directory	Exit Module Name
Windows	32-bit 64-bit	C:\Capitalware\MQCCI\	mqcci.dll
Linux/Unix	32-bit	/var/mqm/exits/	mqcci
Linux/Unix	64-bit	/var/mqm/exits64/	mqcci

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

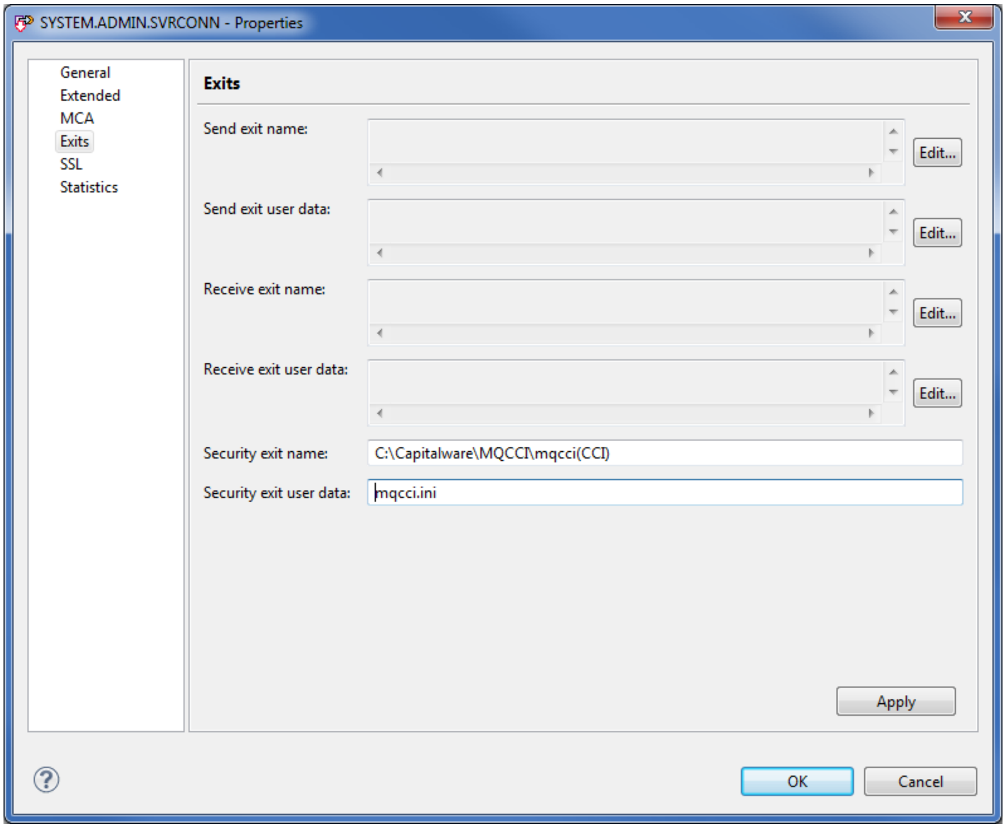
Note: The Security Exit Data (SCYDATA) field must NOT exceed 32 characters.

3.2.1 Windows

On Windows, SCYEXIT and SCYDATA will contain the following values assuming a default install.

- SCYEXIT
  `C:\Capitalware\MQCCI\mqcci(CCI)`
- SCYDATA
  `mqcci.ini`

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('C:\Capitalware\MQCCI\mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```



Security exit name: C:\Capitalware\MQCCI\mqcci(CCI)

Security exit user data: mqcci.ini

3.2.2 Linux 32-bit

On Unix and Linux, SCYEXIT and SCYDATA will contain the following values assuming a default install:

- SCYEXIT
  `/var/mqm/exits/mqcci(CCI)`
- SCYDATA
  `mqcci.ini`

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits/mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

3.2.3 Unix and Linux 64-bit

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

- SCYEXIT
  `/var/mqm/exits64/mqcci(CCI)`
- SCYDATA
  `mqcci.ini`

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqcci(CCI)') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

3.2.4 IBM i

On IBM i, SCYEXIT and SCYDATA will contain the following values assuming a default install:

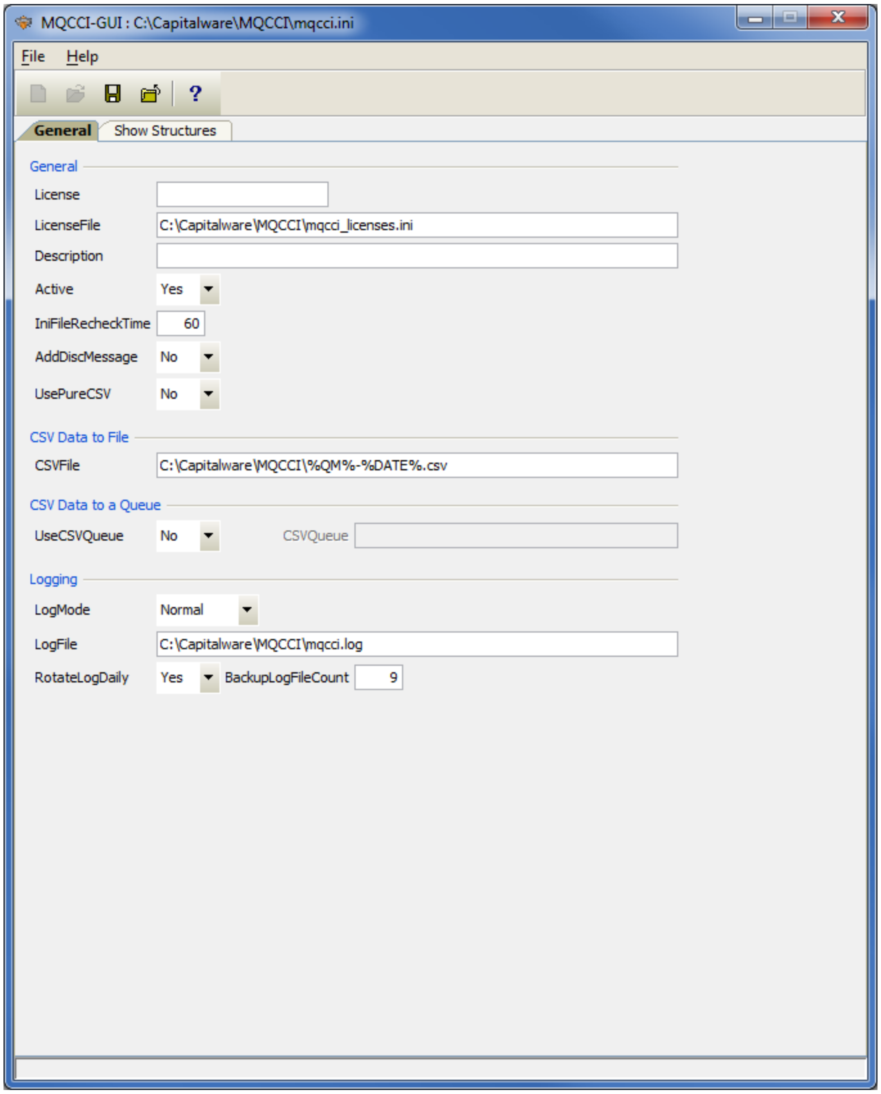
- SCYEXIT is made up of 10 characters for program name (padded with blanks) followed by 10 characters for the LIBRARY name (padded with blanks).
  `MQCCI     MQCCI`

- SCYDATA
  `mqcci.ini`

```
DEFINE CHANNEL ('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCONN) +
       TRPTYPE(TCP) +
       SCYEXIT('MQCCI     MQCCI     ') +
       SCYDATA('mqcci.ini') +
       REPLACE
```

3.3 MQCCI-GUI

This section briefly describes the new graphical program called MQCCI-GUI. MQCCI-GUI assists the user in creating and managing their MQCCI IniFiles. For more information, please see the ***MQCCI-GUI User Guide*** manual.



General

License

LicenseFile C:\Capitalware\MQCCI\mqcci_licenses.ini

Description

Active Yes

IniFileRecheckTime 60

AddDiscMessage No

UsePureCSV No

CSV Data to File

CSVFile C:\Capitalware\MQCCI\%QM%-%DATE%.csv

CSV Data to a Queue

UseCSVQueue No CSVQueue

Logging

LogMode Normal

LogFile C:\Capitalware\MQCCI\mqcci.log

RotateLogDaily Yes BackupLogFileCount 9

4 IniFile Keywords

This section describes IniFile keywords.

4.1 Logging

This section describes the necessary entries to enable MQCCI to record log information. To enable and control logging, there are 4 keywords in the IniFile:

1. **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 N.

2. **LogFile** specifies the location of the log file. The default is as follows:

   For Windows:
   ```
   LogFile=C:\Capitalware\MQCCI\mqcci.log
   ```

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

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

   For IBM MQ on IBM i:
   ```
   LogFile=/QIBM/UserData/mqm/mqcci/mqcci.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

3. **RotateLogDaily** specifies whether or not the log files will be rotated on a daily basis. Setting ' RotateLogDaily' to 'Y' (Yes) will activate this feature; otherwise, the log files will left as is. The default value is Y.

   The RotateLogDaily feature will 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 then the ninth backup log file will be delete and 8 becomes 9, 7 becomes 8, etc…

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

```
LogMode=N
LogFile=C:\Capitalware\MQCCI\mqcci.log
RotateLogDaily=Y
```

4.2 Active

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

```
Active=Y
```

4.3 CSVFile

This section specifies the filename pattern and path to the filename.

Token Replacement for LogFile keyword:

- %QM% - Substitutes the name of the queue manager
- %CHL% - Substitutes the name of the channel
- %DATE% - Substitutes the current date in the format YYYY_MM_DD

The default value for CSVFile is:

For Windows:

```
CSVFile=C:\Capitalware\MQCCI\%QM%-%DATE%.csv
```

For IBM MQ 32-bit on Unix and Linux:

```
CSVFile=/var/mqm/exits/%QM%-%DATE%.csv
```

For IBM MQ 64-bit on Unix and Linux:

```
CSVFile=/var/mqm/exits64/%QM%-%DATE%.csv
```

For IBM MQ on IBM i:

```
CSVFile=/QIBM/UserData/mqm/mqcci/%QM%-%DATE%.csv
```

The value for the CSVFile keyword MUST contain the %DATE% token. It is does not contain the %DATE% token then MQCCI will use the default value.

Note: CSVFile must NOT exceed 2048 characters.

```
CSVFile=%DATE%-%QM%-%CHL%.csv
```

4.4 CSVQueue

This section describes the necessary steps to enable the use of 'CSV Queue'.

The user can write their own program to read the messages from the 'CSV Queue' and to write the information somewhere (i.e. database).

- **UseCSVQueue** specifies that the CSV records are to be written to te CSVQueue.
- **CSVQueue** specifies the name of the CSV queue.

```
UseCSVQueue=Y
CSVQueue=CW.MQCCI.CSVDATA.QUEUE
```

4.5 AddDiscMessage

AddDiscMessage specifies whether or not MQCCI will write a disconnect message when the client application closes the channel. The default value is ‘N’.

```
AddDiscMessage=Y
```

4.6 IncludeTag

IncludeTag specifies whether or not MQCCI will include a 16 character tag on both the “CONN” and “DISC” records. This allows for the matching of the DISC record to the CONN record. The default value is Yes. Note: Only used if AddDiscMessage is set to 'Y'.

```
IncludeTag=Y
```

4.7 IncludeDurationTime

IncludeDurationTime specifies whether or not MQCCI will include the duration time on the “DISC” record. The default value is No. Note: Only used if AddDiscMessage is set to 'Y'.

```
IncludeDurationTime=Y
```

4.8 UsePureCSV

UsePureCSV specifies whether or not MQCCI will write the CSV record in a pure CSV format. The default value is No.

```
UsePureCSV=Y
```

When UsePureCSV is set to 'N' then the CSV record is in the format of “field=value,field=value,field=value,etc.”. The “value” data will be converted to human-readable format. The CSV record will look like (the follow example is 1 line output):

```
2018/01/10 17:25:07.712373, CONN, Tag=F4Evlx0T6ComjD20, CD_QMgrName=MQWT1,
CD_ChannelName=TEST.CHL, CD_ConnectionName=127.0.0.1, , CD_ShortConnectionName=127.0.0.1,
CD_MaxMsgLength=4194304, CD_PutAuthority=MQPA_DEFAULT, CD_MCAUserIdentifier=roger,
CD_RemoteUserIdentifier=roger, CD_RemotePassword_Length=0, CD_SSLCipherSpec=,
CD_SSLClientAuth=MQSCA_REQUIRED, CD_CertificateLabel=, CXP_PartnerName=roger,
CXP_SSLCertUserid=, CXP_SecurityParms_AuthenticationType=MQCSP_AUTH_USER_ID_AND_PWD,
CXP_SecurityParms_UserId=roger, CXP_SecurityParms_Password_Length=8,
CXP_SharingConversations=TRUE, CXP_MCAUserSource=MQUSRC_MAP, CXP_RemoteProduct=MQJB,
CXP_RemoteVersion=0800,
```

When UsePureCSV is set to 'Y' then the CSV record will have the standard string values in double quotes and numeric values without double quotes. The CSV record will look like (the follow example is 1 line output):

```
2018/01/10 17:25:07.712373, "CONN", "9C1yef3FCR73e5Mw", "MQWT1", "MY.TEST.EXIT.CCI",
"127.0.0.1", , "127.0.0.1", 4194304, "MQPA_DEFAULT", "roger", "roger", 0, "", "MQSCA_REQUIRED",
"", "roger", "", "MQCSP_AUTH_USER_ID_AND_PWD", "roger", 8, "TRUE", "MQUSRC_MAP", "MQJB",
"0800",
```

4.9 ShowCD

The keyword ShowCD (MQCD - Channel Definition) contains the fields that the user wants MQCCI to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'CertificateLabel; ConnectionName; MCAUserIdentifier; MaxMsgLength; PutAuthority; RemoteUserIdentifier; RemotePassword; SSLCipherSpec; SSLClientAuth; SSLPeerName'.

The following fields are available:

- BatchDataLimit
- BatchHeartbeat
- BatchInterval
- BatchSize
- CertificateLabel
- ChannelMonitoring
- ChannelStatistics
- ChannelType
- ClientChannelWeight
- ClusterPtr
- ClustersDefined
- CLWLChannelPriority
- CLWLChannelRank
- CLWLChannelWeight
- ConnectionAffinity
- ConnectionName
- DataConversion
- DefReconnect
- Description
- DiscInterval
- ExitDataLength
- ExitNameLength
- HdrCompList
- HeartbeatInterval
- KeepAliveInterval
- LocalAddress
- LongMCAUserId
- LongRemoteUserId
- LongRetryCount
- LongRetryInterval
- MaxInstances
- MaxInstancesPerClient
- MaxMsgLength
- MCAName
- MCASecurityId
- MCAType
- MCAUserIdentifier
- ModeName
- MsgCompList
- MsgExit
- MsgExitPtr
- MsgExitsDefined
- MsgRetryCount
- MsgRetryExit
- MsgRetryInterval
- MsgRetryUserData
- MsgUserData
- MsgUserDataPtr
- NetworkPriority
- NonPersistentMsgSpeed
- Password *
- PropertyControl
- PutAuthority
- ReceiveExit
- ReceiveExitPtr
- ReceiveExitsDefined
- ReceiveUserData
- ReceiveUserDataPtr
- RemotePassword *
- RemoteSecurityId
- RemoteUserIdentifier
- SecurityExit
- SecurityUserData
- SendExit
- SendExitPtr
- SendExitsDefined
- SendUserData
- SendUserDataPtr
- SeqNumberWrap
- SharingConversations
- ShortConnectionName
- ShortRetryCount
- ShortRetryInterval
- SSLCipherSpec
- SSLClientAuth
- SSLPeerName
- TpName
- TransportType
- UseDLQ
- UserIdentifier
- Version
- XmitQName

* Only the length of the Password and RemotePassword will be outputted.

```
ShowCD=CertificateLabel;ConnectionName;SSLCipherSpec;SSLClientAuth;SSLPeerName
```

If the user does not want any fields generated, set the ShowCD keyword equal to 'NONE'.

```
ShowCD=NONE
```

Note: To generate audit information for all of the fields of MQCD, set the ShowCD keyword equal to '*'.

```
ShowCD=*
```

4.10 ShowCXP

The keyword ShowCXP (MQCXP - Channel Exit Parameter) contains the fields that the user wants MQCCI to generate in the audit information. Separate each field by using by a semi-colon (';'). The default value is 'MCAUserSource; PartnerName; RemoteProduct; RemoteVersion; SecurityParms; SSLCertUserid; SSLRemCertIssName; SharingConversations'.

The following fields are available:

- CapabilityFlags
- CurHdrCompression
- CurMsgCompression
- ExitData
- ExitId
- ExitNumber
- ExitReason
- ExitResponse2
- ExitResponse
- ExitSpace
- FAPLevel
- Feedback
- HeaderLength
- MaxSegmentLength
- MCAUserSource
- MsgRetryCount
- MsgRetryInterval
- MsgRetryReason
- PartnerName
- RemoteProduct
- RemoteVersion
- SecurityParms *
- SharingConversations
- SSLCertUserid
- SSLRemCertIssName
- Version

* SecurityParms' field is a pointer the MQCSP structure. Hence, adding this keyword to ShowCXP will cause MQCCI to output the following 3 fields from MQCSP: AuthenticationType, UserID and Password length.

```
ShowCXP=RemoteProduct;RemoteVersion;SSLCertUserid;SSLRemCertIssName
```

If the user does not want any fields generated, set the ShowCXP keyword equal to 'NONE'.

```
ShowCXP=NONE
```

Note: To generate audit information for all of the fields of MQCXP, set the ShowCXP keyword equal to '*'.

```
ShowCXP=*
```

4.11 LicenseFile

This section will describe how to have a file that contains all of the user's MQCCI 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 = 10S0-AAAA-BBBBBBBB
MQB1 = 10S0-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\MQCCI\mqcci_licenses.ini
```

For IBM MQ 32-bit on Unix and Linux:

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

For IBM MQ 64-bit on Unix and Linux:

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

For IBM MQ on IBM i:

```
LicenseFile=/QIBM/UserData/mqm/mqcci/mqcci_licenses.ini
```

4.12 License Key

This section will describe how to license MQ Channel Connection Inspector 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: 10S0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work).

```
License=10S0-AAAA-BBBBBBBB
```

5 Appendix A – Summary of IniFile

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

```
LogMode=N
LogFile=C:\Capitalware\MQCCI\mqcci.log
CSVFile=%DATE%-%QM%-%CHL%.csv
UseCSVQueue=N
```

Note: Keywords are case sensitive.

Keyword	Description of Server-side keywords
Active	**Active** specifies if MQCCI is enabled or disabled. **Active** supports 2 values [Y / N]. The default value is Y. e.g. **Active**=N
AddDiscMessage	**AddDiscMessage** specifies whether or not MQCCI will write a disconnect message when the client application closes the channel. The default value is No. AddDiscMessage supports 2 values [Y / N]. The default value is N. e.g. AddDiscMessage=Y
BackupLogFileCount	**BackupLogFileCount** specifies the number of backup logfiles that the security exit will be keeping. The default value is 9. e.g. BackupLogFileCount=9
CSVFile	**CSVFile** specifies the location of the CSV file. The default is as follows: For Windows: CSVFile=C:\Capitalware\MQCCI\%QM%-%DATE%.csv For IBM MQ 32-bit on Linux: CSVFile=/var/mqm/exits/%QM%-%DATE%.csv For IBM MQ 64-bit on Unix and Linux: CSVFile=/var/mqm/exits64/%QM%-%DATE%.csv For IBM MQ on IBM i: CSVFile=/QIBM/UserData/mqm/mqcci/%QM%-%DATE%.csv

Keyword	Description of Server-side keywords
CSVQueue	**CSVQueue** specifies the name of the CSV queue. e.g. CSVQueue= CW.MQCCI.CSVDATA.QUEUE Note: Only used if UseCSVQueue is set to 'Y'.
IncludeTag	**IncludeTag** specifies whether or not MQCCI will include a 16 character tag on both the "CONN" and "DISC" records. IncludeTag supports 2 values [Y / N]. The default value is N. e.g. IncludeTag=Y Note: Only used if AddDiscMessage is set to 'Y'.
IncludeDurationTime	**IncludeDurationTime** specifies whether or not MQCCI will include the duration time on the "DISC" record. IncludeDurationTime supports 2 values [Y / N]. The default value is N. e.g. IncludeDurationTime=Y Note: Only used if AddDiscMessage is set to 'Y'.
License	**License** specifies the queue manager's license key. Your license will look something like: 10S0-AAAA-BBBBBBBB (Note: This is a sample license only and will NOT work). e.g. License=10S0-AAAA-BBBBBBB

Keyword	Description of Server-side keywords
LicenseFile	**LicenseFile** specifies the location of License file that contains all of the customer's license keys. The following are the default values for LicenseFile: For Windows: LicenseFile=C:\Capitalware\MQCCI\mqcci_licenses.ini For IBM MQ 32-bit on Unix and Linux: LicenseFile=/var/mqm/exits/mqcci_licenses.ini For IBM MQ 64-bit on Unix and Linux: LicenseFile=/var/mqm/exits64/mqcci_licenses.ini For IBM MQ on IBM i: LicenseFile=/QIBM/UserData/mqm/mqcci/mqcci_licenses.ini e.g. LicenseFile=/var/mqm/exits64/mqcci_licenses.ini
LogFile	**LogFile** specifies the location of the log file. The default is as follows: For Windows: LogFile=C:\Capitalware\MQCCI\mqcci.log For IBM MQ 32-bit on Linux: LogFile=/var/mqm/exits/mqcci.log For IBM MQ 64-bit on Unix and Linux: LogFile=/var/mqm/exits64/mqcci.log For IBM MQ on IBM i: LogFile=/QIBM/UserData/mqm/mqcci/mqcci.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 N. e.g. LogMode=N
RotateLogDaily	**RotateLogDaily** specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y. e.g. RotateLogDaily=Y

Keyword	Description of Server-side keywords
ShowCD	**ShowCD** (Channel Definition) contains the fields of the MQCD structure. The default value is 'CertificateLabel; ConnectionName; MCAUserIdentifier; MaxMsgLength; PutAuthority; RemoteUserIdentifier; SSLCipherSpec; SSLClientAuth; SSLPeerName'. e.g. **ShowCD**=CertificateLabel;ConnectionName;SSLCipherSpec;SSLClientAuth;SSLPeerName
ShowCXP	**ShowCXP** (Channel Exit Parameters) contains the fields of the MQCXP structure. The default value is 'MCAUserSource; PartnerName; RemoteProduct; RemoteVersion; SecurityParms; SSLCertUserid; SSLRemCertIssName; SharingConversations'. e.g. ShowCXP=RemoteProduct;RemoteVersion;SSLCertUserid;SSLRemCertIssName
UseCSVQueue	**UseCSVQueue** specifies that the CSV records are to be written to te CSVQueue. UseCSVQueue supports 2 values [Y / N]. The default value is N. e.g. UseCSVQueue=Y
UsePureCSV	**UsePureCSV** specifies whether or not MQCCI will write the CSV record in a pure CSV format. The default value is No. e.g. UsePureCSV=N

6 Appendix B – MQCCI Upgrade Procedures

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

6.1.1 Windows Upgrade

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

6.1.2 Linux 32-bit Upgrade

- Login under the mqm account
- Stop all of the channels using the MQCCI channel security exit or completely stop the queue manager.
- Backup all MQCCI IniFiles in the MQCCI 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  mqcci_linux.tar
  ```
- Run the script as follows:

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

6.1.3 Unix and Linux 64-bit Upgrade

- Stop all of the channels using the MQCCI channel security exit or completely stop the queue manager.
- Backup all MQCCI IniFiles in the MQCCI 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  mqcci_aix.tar
  ```
- Run the script as follows:
  ```
    ./setssx.sh
  ```
- Restore the MQCCI IniFiles if they were altered / deleted.
- Delete the MQCCI tar file
- Start all of the channels using the MQCCI channel security exit or restart the queue manager if it was previously stopped.

6.1.4 IBM i Upgrade

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

  ```
  ftp -s:mqcci_iseries.ftp  iseries_hostname
  ```

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

binary
cd QGPL
put mqcci.savf
quote SITE NAMEFMT 1
cd /QIBM/UserData/mqm/
put mqcci_iseries.tar
quit
```

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

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

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

7 Appendix C – Capitalware Product Display Version

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

`cwdspver`

7.1 Examples

7.1.1 Windows

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

```
cwdspver.exe
```

7.1.2 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
```

7.1.3 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
```

7.1.4 IBM i

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

```
CALL MQCCI/CWDSPVER
```

8 Appendix D – Support

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

By email at:
support@capitalware.com

By regular mail at:

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

9 Appendix E – Summary of Changes

- MQ Channel Connection Inspector v1.2.0
  - Enhanced the code for dumping the pointers passed into exit.
  - Fixed an issue in the subroutine that removes trailing blanks
  - Fixed issue when an invalid or expired license key is used
  - Fixed an issue with default exit path

- MQ Channel Connection Inspector v1.1.0
  - Fixed an issue with tracking QMgr to QMgr channels
  - Added support for IncludeTag keyword
  - Added support for IncludeDurationTime keyword
  - Tuned the code that is called on entry
  - Tuned the logging code

- MQ Channel Connection Inspector v1.0.0
  - Initial release.

10 Appendix F – 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 mqcci.ini, mqcci_licenses.ini or any other form.

3. OTHER RESTRICTIONS. The registration notification provided, showing your authorization code and this License is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent or lease the SOFTWARE, but you may transfer your rights under this License on a permanent basis, provided you transfer this License, the SOFTWARE and all accompanying printed materials, retain no copies, and the recipient agrees to the terms of this License. You may not reverse engineer, decompile, or disassemble the SOFTWARE, except to the extent the foregoing restriction is expressly prohibited by applicable law.

LIMITED WARRANTY

LIMITED WARRANTY. Capitalware Inc. warrants that the SOFTWARE will perform substantially in accordance with the accompanying printed material (if any) and on-line documentation for a period of 365 days from the date of receipt.

CUSTOMER REMEDIES. Capitalware Inc. entire liability and your exclusive remedy shall be, at Capitalware Inc. option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet this Limited Warranty and that is returned to Capitalware Inc.

with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

NO OTHER WARRANTIES. To the maximum extent permitted by applicable law, Capitalware Inc. disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE and any accompanying written materials.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall Capitalware Inc. be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use the SOFTWARE, even if Capitalware Inc. has been advised of the possibility of such damages.

11 Appendix G – Notices

Trademarks:

AIX, IBM, MQSeries, OS/2 Warp, OS/400, IBM i, 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.