

MQCCI for z/OS 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: April 2022.
© Copyright Capitalware Inc. 2018, 2022.

Table of Contents

1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 EXECUTIVE SUMMARY.....	1
1.3 PREREQUISITES.....	2
1.3.1 <i>Operating System</i>	2
1.3.2 <i>IBM MQ</i>	2
2 INSTALLING MQ CHANNEL CONNECTION INSPECTOR FOR Z/OS.....	3
2.1 CHANNEL SECURITY EXIT.....	3
2.1.1 <i>z/OS Installation</i>	3
2.1.2 <i>z/MQCCI DataSets</i>	4
2.1.3 <i>z/OS CHIN JCL</i>	5
3 CONFIGURING CHANNEL SECURITY EXIT.....	7
3.1 SECURITY USER DATA (SCYDATA).....	7
3.1.1 <i>SCYDATA with DD Name</i>	7
3.1.2 <i>SCYDATA with DD Name and Member Name</i>	8
3.2 SVRCONN CHANNEL.....	9
3.2.1 <i>z/OS</i>	9
4 INIFILE KEYWORDS.....	10
4.1 LOGGING.....	10
4.2 ACTIVE.....	10
4.3 CSVFILE.....	11
4.4 CSVQUEUE.....	12
4.5 ADDDISCMESSAGE.....	13
4.6 INCLUDETAG.....	13
4.7 INCLUDEDURATIONTIME.....	13
4.8 USEPURECSV.....	14
4.9 SHOWCD.....	15
4.10 SHOWCXP.....	17
4.11 LICENSEFILE.....	18
4.12 LICENSE KEY.....	18
5 APPENDIX A – Z/MQCCI INIFILE.....	19
6 APPENDIX B – Z/MQCCI UPGRADE PROCEDURES.....	22
7 APPENDIX C – CAPITALWARE PRODUCT DISPLAY VERSION.....	24
7.1 EXAMPLES.....	24
7.1.1 <i>z/OS</i>	24
8 APPENDIX D – SUPPORT.....	25
9 APPENDIX E – SUMMARY OF CHANGES.....	26
10 APPENDIX F – LICENSE AGREEMENT.....	27

1 Introduction

1.1 Overview

MQ Channel Connection Inspector for z/OS (z/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.

z/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).

z/MQCCI will operate with IBM MQ v6.0, v7.0, v7.1, v8.0, v9.0, v9.1 and v9.2 for z/OS environments. It works with Server Connection, Receiver, Requester, Sender, Server, Cluster-Sender and Cluster-Receiver channels of IBM MQ queue manager.

z/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.

z/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.

1.2 Executive Summary

The MQ Channel Connection Inspector for z/OS solution is comprised of a channel security exit.

The channel security exit is available as:

- z/OS load-module

The major features of MQ Channel Connection Inspector for z/OS 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 details the minimum supported software levels. These prerequisites apply to the server-side installations of MQ Channel Connection Inspector for z/OS.

1.3.1 Operating System

MQ Channel Connection Inspector for z/OS can be installed on any of the following supported servers:

1.3.1.1 IBM z/OS

- IBM z/OS v1.4 or higher

1.3.2 IBM MQ

- IBM MQ for z/OS v6.0, v7.0, v7.1, v8.0, v9.0, v9.1 and v9.2

2 Installing MQ Channel Connection Inspector for z/OS

This section describes how to install Capitalware's MQ Channel Connection Inspector for z/OS.

2.1 Channel Security Exit

The following files are the platform specific channel security exits and the required initialization file (IniFile).

2.1.1 z/OS Installation

To install the z/MQCCI for z/OS, first unzip the **mqcci_zos-setup.zip**. The zip file contains 2 z/OS XMIT prepared datasets.

- **MQCCI.LOAD.ZOS** is the XMIT dataset that contains the z/OS load-module.
- **MQCCI.SYSIN.ZOS** is the XMIT dataset 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. ftp the z/OS XMIT prepared datasets to the z/OS LPAR.

```
ftp -s:mqcci.ftp z/OS_hostname
```

```
your-z/OS-userid  
your-z/OS-password  
  
binary  
quote SITE recfm=fb lrecl=80 blksize=3120  
put MQCCI.LOAD.ZOS  
put MQCCI.SYSIN.ZOS  
quit
```

If the user receives the following error message then they will need to pre-allocate the z/OS datasets:

```
ftp> put MQCCI.LOAD.ZOS  
200 Port request OK.  
550-SVC99 RETURN CODE=4 S99INFO=0 S99ERROR=38656 HEX=9700 S99ERSN code X'000003F3'.  
550 Unable to create data set xxxxx.MQCCI.LOAD.ZOS for STOR command.  
ftp> put MQCCI.SYSIN.ZOS  
200 Port request OK.  
550-SVC99 RETURN CODE=4 S99INFO=0 S99ERROR=38656 HEX=9700 S99ERSN code X'000003F3'.  
550 Unable to create data set xxxxx.MQCCI.SYSIN.ZOS for STOR command.
```

To pre-allocating the XMIT datasets go to option 3.2 of ISPF and allocate both datasets: MQCCI.LOAD.ZOS and MQCCI.SYSIN.ZOS.

Use the following dataset attributes when allocating both datasets:

Space	
Units	BLOCKS
Primary Quantity	40
Secondary Quantity	40
Directory Blocks	0
DCB Parameters	
RECFM	FB
LRECL	80
BLKSIZE	3120
DsnType	Blank

After the user has pre-allocated the datasets, they can redo the ftp commands.

2. Log on to z/OS LPAR and issue the following TSO RECEIVE commands:

```
TSO RECEIVE INDATASET(MQCCI.LOAD.ZOS)
TSO RECEIVE INDATASET(MQCCI.SYSIN.ZOS)
```

After issuing the above commands, the following product datasets will appear:

- **+HLQ+.CPTLWARE.MQCCI.LOAD** is the dataset that contains the z/OS load-module.
- **+HLQ+.CPTLWARE.MQCCI.SYSIN** is a dataset that contains a sample initialization file for the channel security exit and sample MQSC script to define MQ channels with the security exits.

2.1.2 z/MQCCI DataSets

z/MQCCI solution is comprised of 2 datasets: +HLQ+.CPTLWARE.MQCCI.LOAD and +HLQ+.CPTLWARE.MQCCI.SYSIN.

2.1.2.1 +HLQ+.CPTLWARE.MQCCI.LOAD

- **MQCCI** is the actual security exit z/OS load-module that will be invoked by the MQ Server component.

2.1.2.2 +HLQ+.CPTLWARE.MQCCI.SYSIN

- **MQCCIINI** is a sample initialization file for the channel security exit.
- **CCIMQSC** is a sample MQSC script to define MQ channels with the security exits.

2.1.3 z/OS CHIN JCL

This section describes the required JCL for z/MQCCI.

2.1.3.1 CSQXLIB DDName

The z/MQCCI load-module needs to be put in the executable path for the CHINIT started-task. There are 2 options for achieving this:

1. Add the dataset to the CSQXLIB concatenation of the CHINIT's CSQXLIB.

```
//CSQXLIB DD DISP=SHR,DSN=+MQHLQ+. +QMGRNAME+. USERAUTH  
// DD DISP=SHR,DSN=+HLQ+. CPTLWARE.MQCCI.LOAD
```

2. Or copy the z/MQCCI load-module to your existing MQ exit / link-edited parameter dataset. Here is a sample JCL to copy the z/MQCCI load-module:

```
//COPY1 EXEC PGM=IEBCOPY,REGION=1024K  
//SYSPRINT DD SYSOUT=*  
//SYSUT3 DD DSN=&&SYSUT3,UNIT=SYSDA,DISP=(,DELETE),  
// SPACE=(CYL,(5,1))  
//SYSUT4 DD DSN=&&SYSUT4,UNIT=SYSDA,DISP=(,DELETE),  
// SPACE=(CYL,(5,1))  
//*  
//IN DD DISP=SHR,DSN=+HLQ+. CPTLWARE.MQCCI.LOAD  
//*  
//OUT DD DISP=SHR,DSN=+MQHLQ+. +QMGRNAME+. USERAUTH  
//*  
//SYSIN DD *  
COPYMOD OUTDD=OUT,INDD=((IN,R))  
S M=MQCCI  
/*
```

2.1.3.2 MQCCIIN DDName

MQCCIIN is the DDName that points to a dataset containing the IniFile parameters.

Add the following line to the CHINIT's JCL.

```
//MQCCIIN DD DISP=SHR,DSN=+HLQ+. CPTLWARE.MQCCI.SYSIN(MQCCIINI)
```

2.1.3.3 CSVFILE DDName - Optional

CSVFILE is the DDName that points to a dataset containing the UserId proxy values.

Add the following line to the CHINIT's JCL:

```
//CSVFILE DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCCI.CSVFILE
```

To allocate the CSVFILE dataset go to option 3.2 of ISPF and allocate a dataset using the following dataset attributes:

Space	
Units	MEGABYTE
Primary Quantity	100
Secondary Quantity	200
Directory Blocks	0
DCB Parameters	
RECFM	VB
LRECL	4096
BLKSIZE	32760
DsnType	LIBRARY

Note: The MQAdmin can use any DD for outputting the CSV data. i.e. SYSPRINT. Hence, the MQAdmin would not need to manage the dataset if SYSPRINT DD is used.

3 Configuring Channel Security Exit

This section describes how to configure the channel security exit.

3.1 Security User Data (SCYDATA)

MQCCI supports 2 ways to specify an IniFile via the Security User Data (SCYDATA) field: DD Name and DD Name with a Member Name.

3.1.1 SCYDATA with DD Name

In this case, only the DD Name is used to specify the IniFile. The DD Name provided in the SCYDATA field must match the DD Name in the CHIN's JCL. The DD statement's DSN keyword can contain either a fully qualified Partition DataSet with the Member name or a Sequential DataSet.

3.1.1.1 SCYDATA with DD Name using Partition DataSet

The CHIN's DD Name references the DSN keyword which contains the fully qualified Partition DataSet Name (highlighted in **red**) and member name (highlighted in **blue**). Since the Member Name is included in the CHIN'S DD DSN keyword, do not put the Member Name in the SCYDATA field.

e.g.

```
SCYDATA('DDName')
```

CHIN JCL using Partition DataSet

```
//MQCCIIN DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCCI.SYSIN(MQCCIINI)
```

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

3.1.1.2 SCYDATA with DD Name using Sequential DataSet

The CHIN's DD Name specifies a DSN which will contain the Sequential DataSet. As seen below, the DD Name in the SCYDATA field matches the DD Name in the CHIN's JCL.

e.g.

```
SCYDATA('DDName')
```

CHIN JCL using Sequential DataSet

```
//MQCCIIN DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCCI.SYSIN.SEQ
```

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

3.1.2 SCYDATA with DD Name and Member Name

In this case, both the DD Name (highlighted in **red**) and the Member Name (highlighted in **blue**) are used to specify the IniFile since the DSN keyword of the DD statement only contains the Partition DataSet Name. In other words, the user specifies the Member Name as a parameter to the SCYDATA field. This is a dynamic configuration that allows for different IniFiles for different channels.

e.g.

```
SCYDATA('DDName(MemberName)')
```

CHIN JCL using Partition DataSet

```
//MQCCIIN DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCCI.SYSIN
```

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

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.

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

3.2.1 z/OS

On z/OS, SCYEXIT and SCYDATA will contain the following values assuming a default install.

- SCYEXIT
MQCCI
- SCYDATA
MQCCIIN

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

4 IniFile Keywords

This section describes IniFile keywords.

4.1 Logging

This section describes the necessary entries to enable z/MQCCI to write log information. To enable and control logging, you need 3 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 z/OS:

LogFile=SYSPRINT

3. **UseFormFeed** specifies that a FormFeed command be issued once a day at midnight. UseFormFeed supports 2 values [Y / N]. The default value is N.

```
LogMode=N
LogFile=SYSPRINT
```

4.2 Active

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

```
Active=Y
```

4.3 CSVFile

This section specifies the DD to output the CSV record to.

The default value for CSVFile is CSVFILE.

For z/OS:

CSVFile=CSVFILE

Note: The MQAdmin can use any DD for outputting the CSV data. i.e. SYSPRINT or SYSOUT

Note: CSVFile must NOT exceed 2048 characters.

```
CSVFile=CSVFILE
```

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 z/MQCCI will write a disconnect message when the client application closes the channel. The default value is No.

```
AddDiscMessage=Y
```

4.6 IncludeTag

IncludeTag specifies whether or not z/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 z/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 z/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, QMgrName=MQWT1, ChlName=TEST.EXIT, CD_MaxMsgLength=4194304,
CD_PutAuthority=MQPA_DEFAULT, CD_MCAUserIdentifier=barney, CD_ConnectionName=10.10.10.10,
CD_RemoteUserIdentifier=barney, CXP_RemoteProduct=MQCC, 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", "MQWT1", "TEST.EXIT", 4194304, "MQPA_DEFAULT", "barney",
"10.10.10.10", "barney", "MQCC", "0800",
```

4.9 ShowCD

The keyword ShowCD (MQCD - Channel Definition) contains the fields that the user wants z/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:

<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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 z/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:

<ul style="list-style-type: none">• CapabilityFlags• CurHdrCompression• CurMsgCompression• ExitData• ExitId• ExitNumber• ExitReason• ExitResponse2• ExitResponse• ExitSpace• FAPLevel• Feedback• HeaderLength	<ul style="list-style-type: none">• 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 z/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'sz/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 z/OS DD:

LicenseFile=MQCCIFIL

4.12 License Key

This section will describe how to license MQ Channel Connection Inspector for z/OS 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 – z/MQCCI IniFile

The sample IniFile below is the z/MQCCIINI file supplied for z/OS. The IniFile supports the following keywords and their values:

```
LogMode=N
LogFile=SYSPRINT
CSVFile=CSVFILE
UseCSVQueue=N
SequenceNumberFlag=N
```

Note: Keywords are case sensitive.

Keyword	Description of Server-side keywords
Active	<p>Active specifies if z/MQCCI is enabled or disabled. Active supports 2 values [Y / N]. The default value is Y.</p> <p>e.g. Active=N</p>
AddDiscMessage	<p>AddDiscMessage specifies whether or not z/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.</p> <p>e.g. AddDiscMessage=Y</p>
CSVFile	<p>CSVFile specifies the location of the CSV file. The default is as follows:</p> <p>For Windows: CSVFile=C:\Capitalware\MQCCI\%QM%-%DATE%.csv</p> <p>For IBM MQ 32-bit on Linux: CSVFile=/var/mqm/exits/%QM%-%DATE%.csv</p> <p>For IBM MQ 64-bit on Unix and Linux: CSVFile=/var/mqm/exits64/%QM%-%DATE%.csv</p> <p>For IBM MQ on IBM i: CSVFile=/QIBM/UserData/mqm/mqcci/%QM%-%DATE%.csv</p>
CSVQueue	<p>CSVQueue specifies the name of the CSV queue.</p> <p>e.g. CSVQueue= CW.MQCCI.CSVDATA.QUEUE</p> <p>Note: Only used if UseCSVQueue is set to 'Y'.</p>

Keyword	Description of Server-side keywords
IncludeTag	<p>IncludeTag specifies whether or not z/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.</p> <p>e.g. IncludeTag=Y</p> <p>Note: Only used if AddDiscMessage is set to 'Y'.</p>
IncludeDurationTime	<p>IncludeDurationTime specifies whether or not z/MQCCI will include the duration time on the "DISC" record. IncludeDurationTime supports 2 values [Y / N]. The default value is N.</p> <p>e.g. IncludeDurationTime=Y</p> <p>Note: Only used if AddDiscMessage is set to 'Y'.</p>
License	<p>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).</p> <p>e.g. License=10S0-AAAA-BBBBBBBB</p>
LicenseFile	<p>LicenseFile specifies the location of License file that contains all of the customer's license keys.</p> <p>The following are the default values for LicenseFile:</p> <p>For z/OS DD: LicenseFile=MQCCIFIL</p> <p>e.g. LicenseFile=MQCCIFIL</p>
LogFile	<p>LogFile specifies the location of the log file. The default is as follows:</p> <p>For z/OS: LogFile=SYSPRINT</p>
LogMode	<p>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.</p> <p>e.g. LogMode=V</p>

Keyword	Description of Server-side keywords
SequenceNumberFlag	<p>SequenceNumberFlag is a z/OS (OS/390) only flag. It states whether or not there are sequence numbers in columns 72 to 80. SequenceNumberFlag supports 2 values [Y / N]. The default value is N.</p> <p>e.g. SequenceNumberFlag = Y</p>
ShowCD ShowCXP	<p>ShowCD (Channel Definition) contains the fields of the MQCD structure. The default value is 'CertificateLabel; ConnectionName; MCAUserIdentifier; MaxMsgLength; PutAuthority; RemoteUserIdentifier; SSLCipherSpec; SSLClientAuth; SSLPeerName'.</p> <p>e.g. ShowCD=CertificateLabel;ConnectionName;SSLCipherSpec;SSLClientAuth;SSLPeerName</p> <p>ShowCXP (Channel Exit Parameters) contains the fields of the MQCXP structure. The default value is 'MCAUserSource; PartnerName; RemoteProduct; RemoteVersion; SecurityParms; SSLCertUserid; SSLRemCertIssName; SharingConversations'.</p> <p>e.g. ShowCXP=RemoteProduct;RemoteVersion;SSLCertUserid;SSLRemCertIssName</p>
UseCSVQueue	<p>UseCSVQueue specifies that the CSV records are to be written to te CSVQueue. UseCSVQueue supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UseCSVQueue=Y</p>
UsePureCSV	<p>UsePureCSV specifies whether or notz/MQCCI will write the CSV record in a pure CSV format. The default value is No.</p> <p>e.g. UsePureCSV=N</p>

6 Appendix B – z/MQCCI Upgrade Procedures

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

1. Stop all of the channels using the z/MQCCI channel security exit or stop the queue manager's CHIN (channel initiator).
2. ftp the z/OS XMIT prepared datasets to the z/OS LPAR.

ftp -s:mqcci.ftp z/OS_hostname

```
your-z/OS-userid
your-z/OS-password

binary
quote SITE recfm=fb lrecl=80 blksize=3120
put MQCCI.LOAD.ZOS
quit
```

If the user receives the following error message then they will need to pre-allocate the z/OS datasets:

```
ftp> put MQCCI.LOAD.ZOS
200 Port request OK.
550-SVC99 RETURN CODE=4 S99INFO=0 S99ERROR=38656 HEX=9700 S99ERSN code X'000003F3'.
550 Unable to create data set xxxxx.MQCCI.LOAD.ZOS for STOR command.
ftp> put MQCCI.SYSIN.ZOS
200 Port request OK.
550-SVC99 RETURN CODE=4 S99INFO=0 S99ERROR=38656 HEX=9700 S99ERSN code X'000003F3'.
550 Unable to create data set xxxxx.MQCCI.SYSIN.ZOS for STOR command.
```

To pre-allocating the XMIT datasets go to option 3.2 of ISPF and allocate both dataset: MQCCI.LOAD.ZOS

Use the following dataset attributes when allocating the dataset:

Space	
Units	BLOCKS
Primary Quantity	40
Secondary Quantity	40
Directory Blocks	0
DCB Parameters	
RECFM	FB
LRECL	80
BLKSIZE	3120
DsnType	Blank

After the user has pre-allocated the dataset, the user can redo the ftp commands.

- Log on to z/OS LPAR and issue the following TSO RECEIVE command:

TSO RECEIVE INDATASET(MQCCI.LOAD.ZOS)

After issuing the above command, the following product dataset will appear:

+HLQ+.CPTLWARE.MQCCI.LOAD is the dataset that contains the z/OS load-module.

- Start all of the channels using the z/MQCCI channel security exit or restart the queue manager's CHIN.

7 Appendix C – Capitalware Product Display Version

z/MQCCI includes a program to display the product version number.

7.1 Examples

7.1.1 z/OS

To use the CWDSPVER program on z/OS, use the following JCL:

```
//CWDSPVER EXEC PGM=CWDSPVER,  
//SYSPRINT DD SYSOUT=*  
//STEPLIB DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCCI.LOAD
```

8 Appendix D – Support

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

By email at:

support@capitalware.com

By regular mail at:

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

9 Appendix E – Summary of Changes

- MQ Channel Connection Inspector for z/OS 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 for z/OS 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 for z/OS 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, iSeries, MVS, OS/390, REXX, ISPF, TSO, 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.