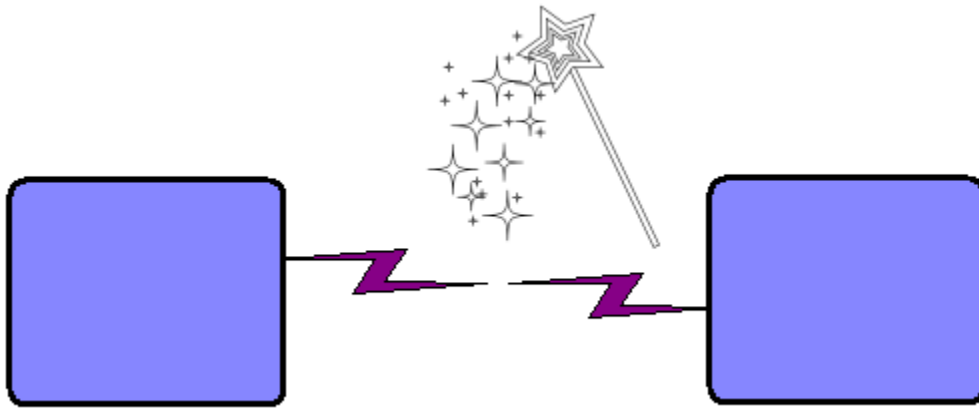


MQ Channel Auto Creation Manager for z/OS Installation and Operation Manual



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

Last Updated: July 2020.
© Copyright Capitalware Inc. 2010, 2020.

Table of Contents

1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 EXECUTIVE SUMMARY.....	2
1.3 PREREQUISITES.....	2
1.3.1 <i>Operating System</i>	2
1.3.2 <i>IBM MQ</i>	2
2 INSTALLING Z/MQCACM.....	3
2.1 MQCACM.....	3
2.1.1 <i>z/OS Installation</i>	3
2.1.2 <i>z/MQCACM DataSets</i>	4
2.1.3 <i>z/OS CHIN JCL</i>	5
3 CONFIGURING Z/MQCACM.....	6
3.1 INIFILE.....	6
3.1.1 <i>DD Name using Partition DataSet</i>	6
3.1.2 <i>DD Name using Sequential DataSet</i>	6
3.2 CONFIGURING z/MQCACM.....	7
3.2.1 <i>z/OS</i>	7
4 MQCACM INIFILE KEYWORDS.....	8
4.1 LOGGING.....	8
4.2 ALLOW OR RESTRICT THE INCOMING IP ADDRESS.....	9
4.3 ALLOW OR RESTRICT THE INCOMING NAMING STANDARD.....	10
4.4 ALLOWCLUSRCVR.....	11
4.5 ALLOWCLUSDR.....	11
4.6 SETCONNECTIONNAME.....	11
4.7 SETMCAUSER.....	12
4.8 SETMESSAGEEXIT.....	12
4.9 SETRECEIVEEXIT.....	12
4.10 SETSECURITYEXIT.....	13
4.11 SETSENDEXIT.....	13
5 Z/MQCACM CHANNEL AUTO-DEFINITION INIFILE.....	14
5.1 CHANNEL AUTO-DEFINITION INIFILE SEARCH ORDER.....	14
5.2 CHANNEL AUTO-DEFINITION INIFILE KEYWORDS.....	14
5.2.1 <i>ConName</i>	14
5.2.2 <i>Partner</i>	14
5.2.3 <i>MCAUser</i>	14
5.2.4 <i>MsgExit and MsgData</i>	15
5.2.5 <i>RcvExit and RcvData</i>	15
5.2.6 <i>ScyExit and ScyData</i>	15
5.2.7 <i>SendExit and SendData</i>	16
6 MQCACM LOG FILE.....	17
6.1 <i>z/OS</i>	17

7 APPENDIX A – Z/MQCACM INIFILE.....	18
8 APPENDIX B – MQCACM CHANNEL AUTO-DEFINITION INIFILE SUMMARY...21	21
9 APPENDIX C – Z/MQCACM UPGRADE PROCEDURES.....	24
10 APPENDIX D – CAPITALWARE PRODUCT DISPLAY VERSION.....	26
10.1 EXAMPLES.....	26
10.1.1 z/OS.....	26
11 APPENDIX E – SUPPORT.....	27
12 APPENDIX F – SUMMARY OF CHANGES.....	28
13 APPENDIX G – LICENSE AGREEMENT.....	29
14 APPENDIX H – NOTICES.....	33

1 Introduction

1.1 Overview

MQ Channel Auto Creation Manager for z/OS (z/MQCACM) is an MQ Channel Auto-Definition (MQ CHAD) exit that allows a company to control and restrict incoming connection requests to auto-create a channel. z/MQCACM is invoked when a request is received to start an undefined Cluster-Receiver or Cluster-Sender channel. z/MQCACM can modify or clear the supplied default channel definition values for an instance of the channel, so that there is no exit incompatibility (cross-platform or otherwise).

z/MQCACM will operate with IBM MQ v5.3.1, v6.0, v7.0, v7.1, v8.0, v9.0, v9.1 and v9.2 for z/OS environments. It works with Cluster-Receiver and Cluster-Sender channels of IBM MQ queue manager.

z/MQCACM has the ability to allow or restrict the incoming IP address from auto-creating a channel. z/MQCACM compares the incoming IP address against a regular expression pattern using a regular expression parser. If a match is found, the channel auto-creation is allowed; otherwise, the connection request is denied.

z/MQCACM has the ability to allow or restrict the incoming auto-create channel request based on the company's naming standard. z/MQCACM compares the incoming channel name against a regular expression pattern (Naming Standard) using a regular expression parser. If a match is found, the channel auto-creation is allowed; otherwise, the connection request is denied.

z/MQCACM has the ability to define, override or clear the supplied default channel definition values related to Message, Security, Receive and Send Exits. Hence, the appropriate platform specific values can be set for the given exit.

Cluster-Sender and Cluster-Receiver channels do not require the queue manager's CHAD parameter to be enabled for the channel auto-creation to happen. For 3rd party security exits to work with MQ clustering, an MQ CHAD exit is required to override the supplied default channel definition values. Hence, z/MQCACM can be used with 3rd party security exits to explicitly set or clear the supplied default channel definition values.

z/MQCACM can be used to prevent a 'Denial-Of-Service' (DOS) attack against a queue manager by setting all 2 'Allow' keywords (AllowClusRcvr and AllowClusSdr) to 'N'.

Note: IBM MQ for z/OS does not support Channel Auto-Definition for Receiver and Server-Connection channel types.

The z/MQCACM solution is licensed as '*Licensed As Free*' meaning a user can freely download the software and use it (i.e. Same as how IBM licenses MQ Client software.). If the user wants support for z/MQCACM then they can purchase a support subscription from Capitalware.

1.2 Executive Summary

z/MQCACM is an MQ CHAD exit

z/MQCACM is available as:

- z/OS load-module

The major features of z/MQCACM are as follows:

- Prevent Denial-Of-Service (DOS) attacks
- Allows or restricts the incoming IP address against a regular expression pattern
- Allows or restricts the auto-create channel based on company's naming standards
- Ability to define, override or clear fields related Message, Security, Receive and Send Exits
- Ability to work with 3rd party security exits to explicitly set or clear supplied default channel values for Cluster-Sender channels.
- Provides logging capability for all successful and unsuccessful attempts to auto-create channels

1.3 Prerequisites

This section details the minimum supported software levels. These prerequisites apply to the server-side installations of MQ Channel Auto Creation Manager for z/OS.

1.3.1 Operating System

z/MQCACM 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 v5.3.1, v6.0, v7.0, v7.1, v8.0, v9.0, v9.1 and v9.2

2 Installing z/MQCACM

This section describes how to install Capitalware's MQ Channel Auto Creation Manager for z/OS.

2.1 MQCACM

The following files are the platform specific z/MQCACM details and the required initialization file (IniFile).

2.1.1 z/OS Installation

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

- **MQCACM.LOAD.ZOS** is the XMIT dataset that contains the z/OS load-module.
- **MQCACM.SYSIN.ZOS** is the XMIT dataset that contains a sample initialization file for z/MQCACM and sample MQSC script.

Steps to install z/MQCACM:

1. ftp the z/OS XMIT prepared datasets to the z/OS LPAR.

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

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

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

```
ftp> put MQCACM.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.MQCACM.LOAD.ZOS for STOR command.  
ftp> put MQCACM.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.MQCACM.SYSIN.ZOS for STOR command.
```

To pre-allocating the XMIT datasets go to option 3.2 of ISPF and allocate both datasets: MQCACM.LOAD.ZOS and MQCACM.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(MQCACM.LOAD.ZOS)
TSO RECEIVE INDATASET(MQCACM.SYSIN.ZOS)
```

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

- **+HLQ+.CPTLWARE.MQCACM.LOAD** is the dataset that contains the z/OS load-module.
- **+HLQ+.CPTLWARE.MQCACM.SYSIN** is a dataset that contains a sample initialization file for z/MQCACM and sample MQSC script.

2.1.2 z/MQCACM DataSets

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

2.1.2.1 +HLQ+.CPTLWARE.MQCACM.LOAD

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

2.1.2.2 +HLQ+.CPTLWARE.MQCACM.SYSIN

- **MQCACMIN** is a sample initialization file for the server-side security exit.
- **CACMMQSC** is a sample MQSC script

2.1.3 z/OS CHIN JCL

This section describes the required JCL for z/MQCACM.

2.1.3.1 CSQXLIB DDName

The z/MQCACM 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.MQCACM.LOAD
```

2. Or copy the z/MQCACM load-module to your existing MQ exit / link-edited parameter dataset. Here is a sample JCL to copy the z/MQCACM 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.MQCACM.LOAD  
//*  
//OUT DD DISP=SHR,DSN=+MQHLQ+. +QMGRNAME+. USERAUTH  
//*  
//SYSIN DD *  
COPYMOD OUTDD=OUT,INDD=((IN,R))  
S M=MQCACM  
/*
```

2.1.3.2 MQCACMIN DDName

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

Add the following line to the CHINIT's JCL.

```
//MQCACMIN DD DISP=SHR,DSN=+HLQ+. CPTLWARE.MQCACM.SYSIN(MQCACMIN)  
//CACMCS DD DISP=SHR,DSN=+HLQ+. CPTLWARE.MQCACM.SYSIN(CACMCS)
```

3 Configuring z/MQCACM

This section describes how to configure z/MQCACM.

3.1 IniFile

MQCACM supports 2 ways to specify an IniFile and Channel Auto-Definition IniFile.

3.1.1 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**).

CHIN JCL using Partition DataSet

```
//MQCACMIN DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCACM.SYSIN(MQCACMIN)  
//CACMCS DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCACM.SYSIN(CACMCS)
```

3.1.2 DD Name using Sequential DataSet

The CHIN's DD Name specifies a DSN which will contain the Sequential DataSet.

CHIN JCL using Sequential DataSet

```
//MQCACMIN DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCACM.SYSIN.SEQ  
//CACMCS DD DISP=SHR,DSN=+HLQ+.CPTLWARE.MQCACM.CACMCS.SEQ
```

3.2 Configuring z/MQCACM

This section describes the necessary entries to enable z/MQCACM. The MQ Administrator will need to update the queue managers attribute: CHADEXIT.

3.2.1 z/OS

On z/OS, CHAD will contain the following value assuming a default install.

- CHADEXIT
MQCACM

```
ALTER QMGR CHADEXIT('MQCACM')
```

4 MQCACM IniFile Keywords

This section describes IniFile keywords.

4.1 Logging

This section describes the necessary entries to enable z/MQCACM to write log information. To enable and control logging, you need 5 keywords in the IniFile:

1. **LogMode** specifies what type of logging the user wishes to have. LogMode supports 2 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

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

4.2 Allow or Restrict the Incoming IP Address

This section describes the necessary entries to enable the feature that allows or restricts the incoming IP addresses through the use of regular expression patterns. This feature uses the following two keywords:

- **UseAllowIP** controls the use of AllowIP. Set to Y to activate feature.
- **AllowIP** specifies the regular expression patterns that limit the allowable incoming IP addresses

MQCACM will look up the regular expression patterns from the **AllowIP** keyword in order to determine if the entire incoming IP Address matches any of the specified expression patterns. Each regular expression pattern is separated from the next pattern by a semi-colon (;).

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more)
- '?' matches any single character
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z, a-z)
- [SET] matches any character in the specified set,
- [!SET] or [^SET] matches any character except those specified in the set (negation).

A SET can be composed of characters or ranges. A range is in the form: 'character – character' (i.e. 0-9 or A-Z). Although this is the simplest range allowed in the [] pattern, more complex inclusive ranges such as [0-9a-zA-Z] are allowed. [0-9a-zA-Z] specifies that the character can be 0 through 9 **or** a through z **or** A through Z. Other characters are allowed (ie. 8 bit characters) if your system supports them.

In order to suppress the special syntactic significance of any of these characters '[' * ? # @ ! ^ - \, a backslash ('\') must precede the special character.

Note: AllowIP must NOT exceed 1024 characters.

```
UseAllowIP=Y
AllowIP=192.168.*.*;10.15[0-9].2[0-5][0-9];127.0.0.?
```

4.3 Allow or Restrict the Incoming Naming Standard

This section describes the necessary entries to enable the feature that allows or restricts the incoming Naming Standard through the use of regular expression patterns. This feature uses the following two keywords:

- **UseNamingStandard** controls the use of NamingStandard. Set to Y to activate feature.
- **NamingStandard** specifies the regular expression patterns that limit the allowable incoming Naming Standard

MQCACM will look up the regular expression patterns from the **NamingStandard** keyword in order to determine if the entire incoming SSL DN matches any of the specified expression patterns. Each regular expression pattern is separated from the next pattern by a semi-colon (;).

In the regular expression pattern:

- '*' matches any sequence of characters (zero or more)
- '?' matches any single character
- '#' matches any single numeric digit (0-9)
- '@' matches any single alphabetic character (A-Z, a-z)
- [SET] matches any character in the specified set,
- [!SET] or [^SET] matches any character except those specified in the set (negation).

A SET can be composed of characters or ranges. A range is in the form: 'character – character' (i.e. 0-9 or A-Z). Although this is the simplest range allowed in the [] pattern, more complex inclusive ranges such as [0-9a-zA-Z] are allowed. [0-9a-zA-Z] specifies that the character can be 0 through 9 **or** a through z **or** A through Z. Other characters are allowed (ie. 8 bit characters) if your system supports them.

In order to suppress the special syntactic significance of any of these characters '[' * ? # @ ! ^ - \, a backslash ('\') must precede the special character.

Note: NamingStandard must NOT exceed 1024 characters.

```
UseNamingStandard=Y
NamingStandard=TEST_???.CHL;ABC_*.CHL
```

4.4 AllowClusRcvr

This section describes the necessary entries to enable the automatic channel creation for channel type 'CLUSRCVR'. This is controlled by the IniFile's property keyword 'AllowClusRcvr'. Setting 'AllowClusRcvr' to 'Y' (Yes) will activate this feature; otherwise, it will be blocked. The default value is 'N'.

```
AllowClusRcvr=Y
```

4.5 AllowClusSdr

This section describes the necessary entries to enable the automatic channel creation for channel type 'CLUSSDR'. This is controlled by the IniFile's property keyword 'AllowClusSdr'. Setting 'AllowClusSdr' to 'Y' (Yes) will activate this feature; otherwise, it will be blocked. The default value is 'N'.

```
AllowClusSdr=Y
```

4.6 SetConnectionName

This section describes the necessary steps to enable the overriding of the channel's CONNECTIONNAME field. SetConnectionName specifies whether or not the z/MQCACM exit will override the CONNECTIONNAME field.

SetConnectionName can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the CONNECTIONNAME field, set the SetConnectionName value to Y as shown below.

```
SetConnectionName=Y
```

4.7 SetMCAUser

This section describes the necessary steps to enable the overriding of the channel's MCAUSER field. SetMCAUser specifies whether or not the z/MQCACM exit will override the MCAUSER field.

SetMCAUser can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the MCAUSER field, set the SetMCAUser value to Y as shown below.

```
SetMCAUser=Y
```

4.8 SetMessageExit

This section describes the necessary steps to enable the overriding of the channel's MSGEXIT and MSGDATA fields. SetMessageExit specifies whether or not the z/MQCACM exit will override the MSGEXIT and MSGDATA fields.

SetMessageExit can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the MSGEXIT and MSGDATA fields, set the SetMessageExit value to Y as shown below.

```
SetMessageExit=Y
```

4.9 SetReceiveExit

This section describes the necessary steps to enable the overriding of the channel's RCVEXIT and RCVDATA fields. SetReceiveExit specifies whether or not the z/MQCACM exit will override the RCVEXIT and RCVDATA fields.

SetReceiveExit can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the RCVEXIT and RCVDATA fields, set the SetReceiveExit value to Y as shown below.

```
SetReceiveExit=Y
```


4.10 SetSecurityExit

This section describes the necessary steps to enable the overriding of the channel's SCYEXIT and SCYDATA fields. SetSecurityExit specifies whether or not the z/MQCACM exit will override the SCYEXIT and SCYDATA fields.

SetSecurityExit can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the SCYEXIT and SCYDATA fields, set the SetSecurityExit value to Y as shown below.

```
SetSecurityExit=Y
```

4.11 SetSendExit

This section describes the necessary steps to enable the overriding of the channel's SENDEXIT and SENDDATA fields. SetSendExit specifies whether or not the z/MQCACM exit will override the SENDEXIT and SENDDATA fields.

SetSendExit can have a value of either Y or N. The default value is 'N'. If you want the z/MQCACM exit to override the SENDEXIT and SENDDATA fields, set the SetSendExit value to Y as shown below.

```
SetSendExit=Y
```

5 z/MQCACM Channel Auto-Definition IniFile

This section describes how to configure the z/MQCACM Channel Auto-Definition IniFile.

5.1 Channel Auto-Definition IniFile Search Order

The z/MQCACM exit supports individual Channel Auto-Definition IniFile for each queue manager. The CHIN JCL needs to have a *CACMCS* DD for Cluster-Sender channels and *CACMCR* DD for Cluster-Receiver channels

5.2 Channel Auto-Definition IniFile Keywords

IniFile keywords are grouped together in sections. A section name is the actual CLUSSDR channel name. A section name is surrounded by square brackets ('[' and ']').

```
[SECTION-NAME]
ScyExit=keyword-value
ScyData= keyword-value
```

To specify default values for any Channel Auto-Definition IniFile keyword, use the default section. The default section is optional.

```
[default]
ScyExit=
ScyData=
```

The IniFile supports the following keywords and their respective values:

5.2.1 ConName

The ConName keyword specifies a value to override the current CONNAME field in the channel auto-definition. ConName is optional. The ConName keyword is only used if SetConName is set to 'Y' in the mqcacm.ini file and its value is not blank.

5.2.2 Partner

The Partner keyword specifies a value to be verified against the incoming connection request's Partner name. Partner is optional. If the keyword is not specified or its value is blank then no check is performed.

5.2.3 MCAUser

The MCAUser keyword specifies a value to override the current MCAUSER field in the channel auto-definition. MCAUSER is optional. If the keyword is not specified or its value is blank then no override is performed.

5.2.4 MsgExit and MsgData

The MsgExit and MsgData keywords are only used if SetMessageExit is set to 'Y' in the mqcacm.ini file.

5.2.4.1 MsgExit

The MsgExit keyword specifies a value in order to override the current MSGEXIT field in the channel auto-definition. MsgExit is optional. If the keyword is not specified or its value is blank then no override is performed.

5.2.4.2 MsgData

The MsgData keyword specifies a value in order to override the current MSGDATA field in the channel auto-definition. MsgData is optional. If the keyword is not specified or its value is blank then no override is performed.

Below is a sample Channel Auto-Definition IniFiles using MsgExit and MsgData entries:

```
[TO.QMGRNAME]
MsgExit=C:\temp\MyMsgExit(ME)
MsgData= C:\temp\MyData
```

5.2.5 RcvExit and RcvData

The RcvExit and RcvData keywords are only used if SetReceiveExit is set to 'Y' in the mqcacm.ini file.

5.2.5.1 RcvExit Keyword

The RcvExit keyword specifies a value in order to override the current RCVEXIT field in the channel auto-definition. RcvExit is optional. If the keyword is not specified or its value is blank then no override is performed.

5.2.5.2 RcvData Keyword

The RcvData keyword specifies a value in order to override the current RCVDATA field in the channel auto-definition. RcvData is optional. If the keyword is not specified or its value is blank then no override is performed.

Below is a sample Channel Auto-Definition IniFiles using RcvExit and RcvData entries:

```
[TO.QMGRNAME]
RcvExit=C:\temp\MyRcvExit(RE)
RcvData= C:\temp\MyData
```

5.2.6 ScyExit and ScyData

The ScyExit and ScyData keywords are only used if SetSecurityExit is set to 'Y' in the mqcacm.ini file.

5.2.6.1 ScyExit

The ScyExit keyword specifies a value in order to override the current SCYEXIT field in the channel auto-definition. ScyExit is optional. If the keyword is not specified or its value is blank then no override is performed.

5.2.6.2 ScyData

ScyData The ScyData keyword specifies a value in order to override the current SCYDATA field in the channel auto-definition. ScyData is optional. If the keyword is not specified or its value is blank then no override is performed.

Below is a sample Channel Auto-Definition IniFiles using ScyExit and ScyData entries:

```
[TO.QMGRNAME]
ScyExit=
ScyData=
```

5.2.7 SendExit and SendData

The SendExit and SendData keywords are only used if SetSendExit is set to 'Y' in the mqcacm.ini file.

5.2.7.1 SendExit

The SendExit keyword specifies a value in order to override the current SENDEXIT field in the channel auto-definition. SendExit is optional. If the keyword is not specified or its value is blank then no override is performed.

5.2.7.2 SendData

The SendData keyword specifies a value in order to override the current SENDDATA field in the channel auto-definition. SendData is optional. If the keyword is not specified or its value is blank then no override is performed.

Below is a sample Channel Auto-Definition IniFiles using SendExit and SendData entries:

```
[TO.QMGRNAME]
SendExit=C:\temp\MySendExit(SE)
SendData= C:\temp\MyData
```

6 MQCACM Log File

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

6.1 z/OS

The log file is located at the following (assuming a default install of SYSPRINT):

CHIN Started-task JES-log

All log entries will be marked with either **INFO** or **ERROR** in columns 21 to 26.

```
2010/06/04 12:39:23.384 MQCACM #00336 {04860} I: Allowing automatic channel creation: QMgr='MQWT2'  
ChlName='TEST.IT' ConName='127.0.0.1' RemoteUserID=''
```

7 Appendix A – z/MQCACM IniFile

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

```
LogMode=N
LogFile=SYSPRINT
AllowClusSdr=Y
AllowClusRcvr=Y
UseAllowIP=N
AllowIP=*
UseNamingStandard=N
NamingStandard=*
SetSecurityExit=Y
SequenceNumberFlag=N
```

Note: Keywords are case sensitive.

Keyword	Description of Server-side keywords
AllowClusRcvr	<p>AllowClusRcvr specifies whether or not to enable the automatic channel creation for channel type 'CLUSRCVR'. AllowClusRcvr supports 2 values [Y / N]. The default value is N.</p> <p>e.g. AllowClusRcvr=Y</p>
AllowClusSdr	<p>AllowClusSdr specifies whether or not to enable the automatic channel creation for channel type 'CLUSDDR'. AllowClusSdr supports 2 values [Y / N]. The default value is N.</p> <p>e.g. AllowClusSdr=Y</p>
AllowIP	<p>AllowIP specifies a set of regular expression patterns that the incoming channel's IP address will be compared against. The default is '*'. You must separate the IP regular expression patterns with a ';' semi-colon.</p> <p>e.g. AllowIP=192.168.*.1[0-5][0-9];127.0.0.?.*.*.[0-9]</p> <p>Note: Only used if UseAllowIP is set to 'Y'.</p>
LogFile	<p>LogFile specifies the location of the log file. The default is as follows:</p> <p>For z/OS: LogFile=SYSPRINT</p>

Keyword	Description of Server-side keywords
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 N.</p> <p>e.g. LogMode=N</p>
NamingStandard	<p>NamingStandard specifies to specify a channel naming standard that must be followed by the incoming channel. You must separate the NamingStandard regular expression patterns with a ';' semi-colon.</p> <p>e.g. NamingStandard=TEST_???.CHL;ABC_*.CHL</p> <p>Note: Only used if UseNamingStandard is set to 'Y'.</p>
SetConnectionName	<p>SetConnectionName specifies whether or not the z/MQCACM exit will override the CONNECTIONNAME field. SetConnectionName supports 2 values [Y / N] where the default value is N.</p> <p>e.g. SetConnectionName=Y</p>
SetMCAUser	<p>SetMCAUser specifies whether or not the z/MQCACM exit will override the MCAUSER field. SetMCAUser supports 2 values [Y / N] where the default value is N.</p> <p>e.g. SetMCAUser=Y</p>
SetMessageExit	<p>SetMessageExit specifies whether or not the z/MQCACM exit will override the MSGEXIT and MSGDATA fields. SetMessageExit supports 2 values [Y / N] where the default value is N.</p> <p>e.g. SetMessageExit=Y</p>
SetReceiveExit	<p>SetReceiveExit specifies whether or not the z/MQCACM exit will override the RCVEXIT and RCVDATA fields. SetMessageExit supports 2 values [Y / N] where the default value is N.</p> <p>e.g. SetReceiveExit=Y</p>

Keyword	Description of Server-side keywords
SetSecurityExit	<p>SetSecurityExit specifies whether or not the z/MQCACM exit will override the SCYEXIT and SCYDATA fields. SetMessageExit supports 2 values [Y / N] where the default value is Y.</p> <p>e.g. SetSecurityExit=Y</p>
SetSendExit	<p>SetSendExit specifies whether or not the z/MQCACM exit will override the SENDEXIT and SENDDATA fields. SetMessageExit supports 2 values [Y / N] where the default value is N.</p> <p>e.g. SetSendExit=Y</p>
UseAllowIP	<p>UseAllowIP allows MQ Admin to allow or restrict incoming channel IP address by comparing it against a regular expression pattern. UseAllowIP supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UseAllowIP=Y</p>
UseNamingStandard	<p>UseNamingStandard allows MQ Admin to specify a channel naming standard that must be followed by the incoming channel. UseNamingStandard supports 2 values [Y / N]. The default value is N.</p> <p>e.g. UseNamingStandard=Y</p>

8 Appendix B – MQCACM Channel Auto-Definition IniFile Summary

A sample Channel Auto-Definition IniFile below is the Channel Auto-Definition IniFile supplied:

```
[default]
ScyExit=
ScyData=
MCAUser=MQTEST
```

IniFile keywords are grouped together in sections. A section name is the actual channel name. A section name is surrounded by square brackets ('[' and ']'). To specify default values for any Channel Auto-Definition IniFile keyword, use the default section. The default section is optional. The IniFile supports the following keywords and their respective values:

Note: Keywords are case sensitive.

Keyword	Description of keywords
ConName	<p>ConName specifies a value in order to override the current CONNAME field in the channel auto-definition. ConName is optional. If the keyword is not specified or its value is blank then no override is performed.</p> <p>e.g. ConName=127.0.0.1(1415)</p> <p>Note: Only used if SetConnectionName is set to 'Y' in the MQCACM IniFile.</p>
MCAUser	<p>MCAUser specifies a value in order to override the current MCAUSER field in the channel auto-definition. MCAUser is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. MCAUser=MQTEST</p> <p>Note: Only used if SetMCAUser is set to 'Y' in the MQCACM IniFile</p>
Partner	<p>Partner specifies a value to be verified against the incoming connection request's Partner name. Partner is optional. If the keyword is not specified or its value is blank then no check is performed.</p> <p>e.g. Partner=QM5</p>

Keyword	Description of keywords
ScyData	<p>ScyData specifies a value in order to override the current SCYDATA field in the channel auto-definition. ScyData is optional. If the keyword is not specified then no override is performed.</p> <p>Example: ScyData=</p> <p>Note: Only used if SetSecurityExit is set to 'Y' in the MQCACM IniFile.</p>
ScyExit	<p>ScyExit specifies a value in order to override the current SCYEXIT field in the channel auto-definition. ScyExit is optional. If the keyword is not specified then no override is performed.</p> <p>Example: ScyExit=</p> <p>Note: Only used if SetSecurityExit is set to 'Y' in the MQCACM IniFile.</p>
MsgData	<p>MsgData specifies a value in order to override the current MSGDATA field in the channel auto-definition. MsgData is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. MsgData =SampleMessageData</p> <p>Note: Only used if SetMessageExit is set to 'Y' in the MQCACM IniFile.</p>
MsgExit	<p>MsgExit specifies a value in order to override the current MSGEXIT field in the channel auto-definition. MsgExit is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. MsgExit=SampleExit</p> <p>Note: Only used if SetMessageExit is set to 'Y' in the MQCACM IniFile.</p>
RcvData	<p>RcvData specifies a value in order to override the current RCVDATA field in the channel auto-definition. RcvData is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. RcvData=SampleMessageData</p> <p>Note: Only used if SetReceiveExit is set to 'Y' in the MQCACM IniFile.</p>

Keyword	Description of keywords
RcvExit	<p>RcvExit specifies a value in order to override the current RCVEXIT field in the channel auto-definition. RcvExit is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. RcvExit =SampleExit</p> <p>Note: Only used if SetReceiveExit is set to 'Y' in the MQCACM IniFile.</p>
SendData	<p>SendData specifies a value in order to override the current SENDDATA field in the channel auto-definition. SendData is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. SendData=SampleMessageData</p> <p>Note: Only used if SetSendExit is set to 'Y' in the MQCACM IniFile.</p>
SendExit	<p>SendExit specifies a value in order to override the current SENDEXIT field in the channel auto-definition. SendExit is optional. If the keyword is not specified then no override is performed.</p> <p>e.g. SendExit =SampleExit</p> <p>Note: Only used if SetSendExit is set to 'Y' in the MQCACM IniFile.</p>

9 Appendix C – z/MQCACM Upgrade Procedures

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

1. Stop the queue manager's CHIN (channel initiator).
2. ftp the z/OS XMIT prepared datasets to the z/OS LPAR.

ftp -s:mqcacm.ftp z/OS_hostname

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

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

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

```
ftp> put MQCACM.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.MQCACM.LOAD.ZOS for STOR command.
ftp> put MQCACM.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.MQCACM.SYSIN.ZOS for STOR command.
```

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

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

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

- Restart the queue manager's CHIN.

10 Appendix D – Capitalware Product Display Version

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

10.1 Examples

10.1.1 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.MQCACM.LOAD
```

11 Appendix E – Support

The support for MQ Channel Auto Creation Manager for z/OS can be found at the following location:

Online Help Desk Ticketing System at
www.capitalware.com/phpst/

By email at:
support@capitalware.com

By regular mail at:

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

12 Appendix F – Summary of Changes

- MQ Channel Auto Creation Manager for z/OS v1.0.6
 - Enhanced the code for dumping the pointers passed into exit.
 - Fixed an issue in the subroutine that removes trailing blanks
- MQ Channel Auto Creation Manager for z/OS v1.0.5
 - Tuned the code that is called on entry
 - Tuned the logging code
- MQ Channel Auto Creation Manager for z/OS v1.0.4
 - Fixed an issue in the logging framework where a constant was being modified.
- MQ Channel Auto Creation Manager for z/OS v1.0.3
 - Enhanced logging - the LogFile keyword now supports the following tokens: %QM% and %CHL%
- MQ Channel Auto Creation Manager for z/OS v1.0.2
 - Fixed an issue with the Ini Processor not finding next section
 - Fixed an issue with using "size_t" variable type when it should have been "int"
- MQ Channel Auto Creation Manager for z/OS v1.0.1
 - Improved the IniFile processing speed.
- MQ Channel Auto Creation Manager for z/OS v1.0.0
 - Initial release.

13 Appendix G – License Agreement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this

definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for

informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2012 Capitalware Inc.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

14 Appendix H – 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.