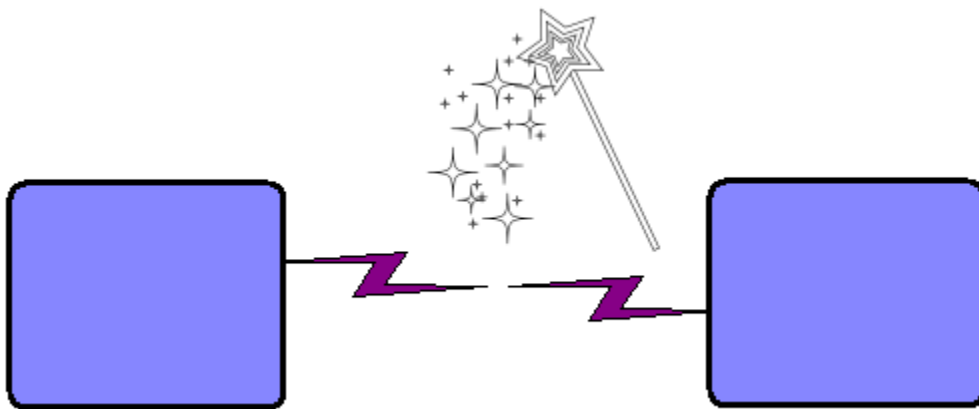


MQ Channel Auto Creation Manager Cluster Configuration Manual



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

Last Updated: January 2021.
© Copyright Capitalware Inc. 2010, 2021.

Table of Contents

1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
2 CLUSTERING OVERVIEW.....	2
2.1 CLUSTER-SENDER AND CLUSTER-RECEIVER CHANNEL PAIR.....	2
2.2 CHANNEL AUTO-DEFINITION.....	3
3 MQCACM CLUSTERING PREREQUISITES.....	4
3.1 PREREQUISITE # 1.....	4
3.2 PREREQUISITE # 2.....	4
3.3 PREREQUISITE # 3.....	4
4 IMPLEMENTING MQCACM WITH CLUSTER CHANNELS.....	5
4.1 STOP THE CLUSTER CHANNELS.....	5
4.2 EDITING THE CHANNEL AUTO-DEFINITION INIFILE.....	5
4.2.1 Windows.....	5
4.2.2 Linux 32-bit.....	5
4.2.3 Unix and Linux 64-bit.....	6
4.2.4 IBM i.....	6
4.2.5 z/OS.....	6
4.3 CONFIGURING THE MQCACM EXIT.....	7
4.3.1 Windows.....	7
4.3.2 Linux 32-bit.....	8
4.3.3 Unix and Linux 64-bit.....	8
4.3.4 IBM i.....	8
4.3.5 z/OS.....	9
4.4 CONFIGURING A CLUSTER-RECEIVER CHANNEL.....	10
4.4.1 Windows.....	10
4.4.2 Linux 32-bit.....	11
4.4.3 Unix and Linux 64-bit.....	11
4.4.4 IBM i.....	12
4.4.5 z/OS.....	12
4.5 CONFIGURING A CLUSTER-SENDER CHANNEL.....	13
5 APPENDIX A – MQCACM INIFILE SUMMARY.....	14
6 APPENDIX B – MQCACM CHANNEL AUTO-DEFINITION INIFILE SUMMARY...18	18
7 APPENDIX C – LICENSE AGREEMENT.....	21
8 APPENDIX D – NOTICES.....	25

1 Introduction

1.1 Overview

MQ Channel Auto Creation Manager (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. MQCACM is invoked when a request is received to start an undefined Receiver, Server-Connection, Cluster-Receiver or Cluster-Sender channel. 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).

MQCACM will operate with IBM MQ v7.0, 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, Cluster-Receiver and Cluster-Sender channels of IBM MQ queue manager.

MQCACM has the ability to allow or restrict the incoming IP address from auto-creating a channel. 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.

MQCACM has the ability to allow or restrict the incoming auto-create channel request based on the company's naming standard. 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.

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, MQCACM can be used with 3rd party security exits to explicitly set or clear the supplied default channel definition values.

MQCACM can be used to prevent a 'Denial-Of-Service' (DOS) attack against a queue manager by setting all 4 'Allow' keywords (AllowClusRcvr, AllowClusSdr, AllowRcvr and AllowSvrConn) to 'N'.

On IBM i, Linux, Unix and Windows, MQCACM 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.

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

2 Clustering Overview

This section provides an overview of MQ clustering

2.1 Cluster-Sender and Cluster-Receiver Channel Pair

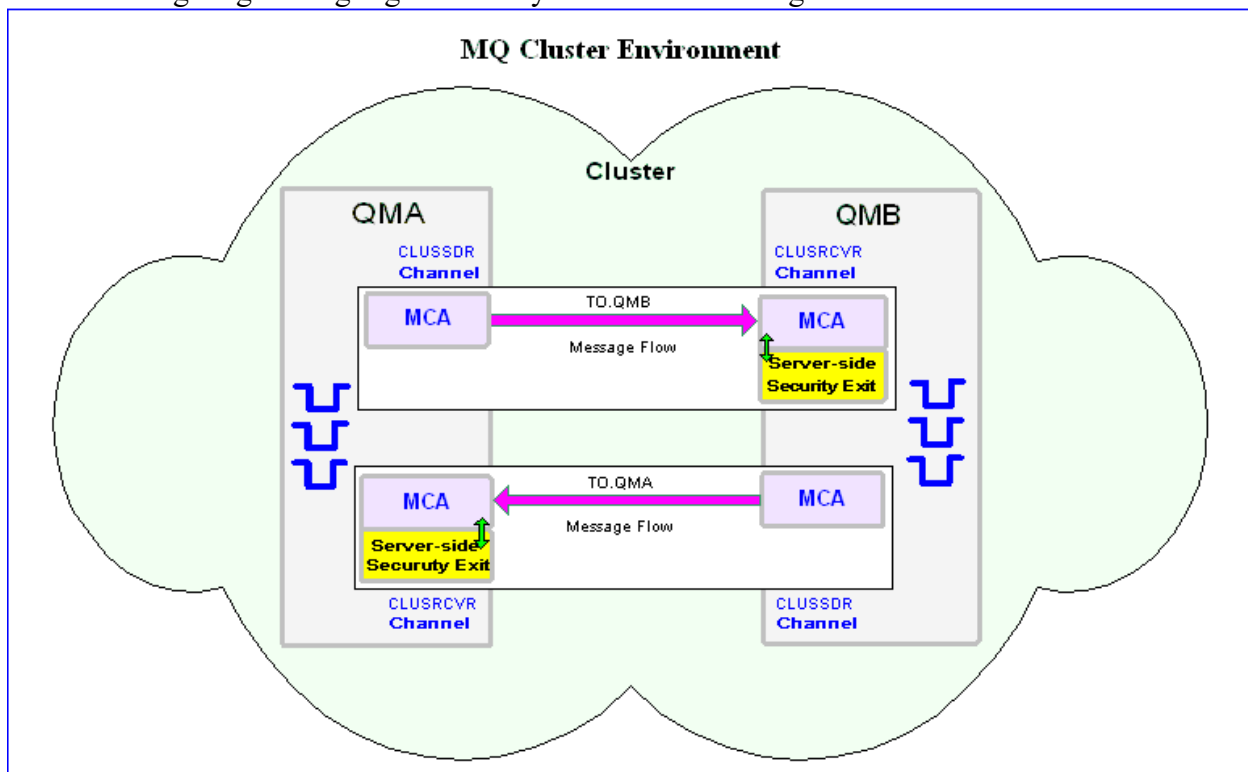
As noted below (in yellow) in the diagram, 3rd party server-side security exits work with the Cluster-Receiver (CLUSRCVR) channel.

There is a Message Channel Agent (MCA) at each end of the channel. The MCA is a component that handles the sending and receiving of messages between queue managers. Before the MCA can send and receive messages, the server-side security exit must be invoked as detailed below:

- The MCA that is running the Cluster-Receiver channel will call server-side security exit to perform its security related functions.

After the server-side security exit has completed its processing, the channel will go to a 'Running' state and the messages will flow along the channel.

The following diagram highlights security exits in a clustering environment:



2.2 Channel Auto-Definition

When a queue manager has joined a cluster, MCA creates, by auto-definition, the Cluster-Sender channel from the attributes of the Cluster-Receiver channel. If an MQAdmin manually defines a Cluster-Sender channel, the queue manager will automatically modify those values to match the Cluster-Receiver channel's attributes. In other words, the MQAdmin cannot modify an auto-created Cluster-Sender channel via MQSC commands.

To explicitly set the Cluster-Sender channel attributes, a Channel Auto-Definition exit must be used. The MQCACM installation package includes an MQ Channel Auto-Definition (MQ CHAD) exit called *MQCACM*.

MQCACM has been designed to override 10 attributes of a Cluster-Sender channel as listed below:

- CONNAME - The connection information for the channel
- MCAUSER – The UserID to be used by the channel
- MSGEXIT - The message exit name for the channel
- MSGDATA - The message exit data for the channel
- RCVEXIT - The receive exit name for the channel
- RCVDATA - The receive exit data for the channel
- SCYEXIT - The security exit name for the channel
- SCYDATA - The security exit data for the channel
- SENDEXIT - The send exit name for the channel
- SENDDATA - The send exit data for the channel

Note: MQCACM also works with these other channels: Cluster-Receiver, Receiver and Server-Connection channels.

3 MQCACM Clustering Prerequisites

This section details the necessary steps that must be completed **BEFORE** implementing MQCACM and any 3rd party security exit in a MQ clustering environment. The implementation of security exits in an MQ clustering environment is extremely complex and must be completed very carefully.

3.1 Prerequisite # 1

The MQAdmin must read and understand Chapters 1 through 4 of IBM's *IBM MQ Queue Manager Clusters* manual. The IBM MQ manuals can be found at the following link: <http://www.ibm.com/software/integration/wmq/library/>

3.2 Prerequisite # 2

The MQAdmin must make sure that all of the cluster channels have been successfully started. i.e. all cluster channels must have a channel status of 'Running'.

3.3 Prerequisite # 3

MQAdmin must stop and restart all Cluster-Sender channels (since the Channel Auto-Definition does not kick-in until they have been restarted) at least twice (yes, twice). They must have a channel status of 'Running'. ***This step is absolutely crucial.***

4 Implementing MQCACM with Cluster Channels

This section describes the necessary steps to enable 3rd party server-side security exits and MQCACM with cluster channels. All steps must be followed exactly. i.e. no skipping of steps.

4.1 Stop The Cluster Channels

The MQAdmin must stop *ALL* CLUSSDR and CLUSRCVR channels (to which you are applying the MQCACM security exits to). *The channels must be in the 'Stopped' state before continuing any further.*

4.2 Editing the Channel Auto-Definition IniFile

This section describes the necessary entries to enable the MQCACM exit. Please review Chapter 5 for more information on editing the Channel Auto-Definition IniFile.

The following examples below for Windows, Unix, Linux, IBM i and z/OS are for server-side only security exit implementations.

4.2.1 Windows

For Windows, a sample Channel Auto-Definition IniFile with the name: *mqcacm.clussdr.ini* has been included with the MQCACM installation package.

The following is the contents of the *mqcacm.clussdr.ini* file:

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

4.2.2 Linux 32-bit

For Linux, a sample Channel Auto-Definition IniFile with the name: *mqcacm.clussdr.ini* has been included with the MQCACM installation package.

The following is the contents of the *mqcacm.clussdr.ini* file:

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

4.2.3 Unix and Linux 64-bit

For Unix and Linux (excluding Linux x86), a sample Channel Auto-Definition IniFile with the name: *mqcacm.clussdr.ini* has been included with the MQCACM installation package.

The following is the contents of the *mqcacm.clussdr.ini* file:

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

4.2.4 IBM i

For IBM i, a sample Channel Auto-Definition IniFile with the name: *mqcacm.clussdr.ini* has been included with the MQCACM installation package.

The following is the contents of the *mqcacm.clussdr.ini* file:

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

4.2.5 z/OS

For z/OS, a sample Channel Auto-Definition IniFile with the name: DD *CACMCS* has been included with the z/MQCACM installation package.

The following is the contents of the DD *CACMCS*:

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

4.3 Configuring the MQCACM Exit

This section describes how to configure the MQCACM exit for a queue manager.

Note: Do NOT alter the QMGR CHAD parameter, as it is not required for clustering.

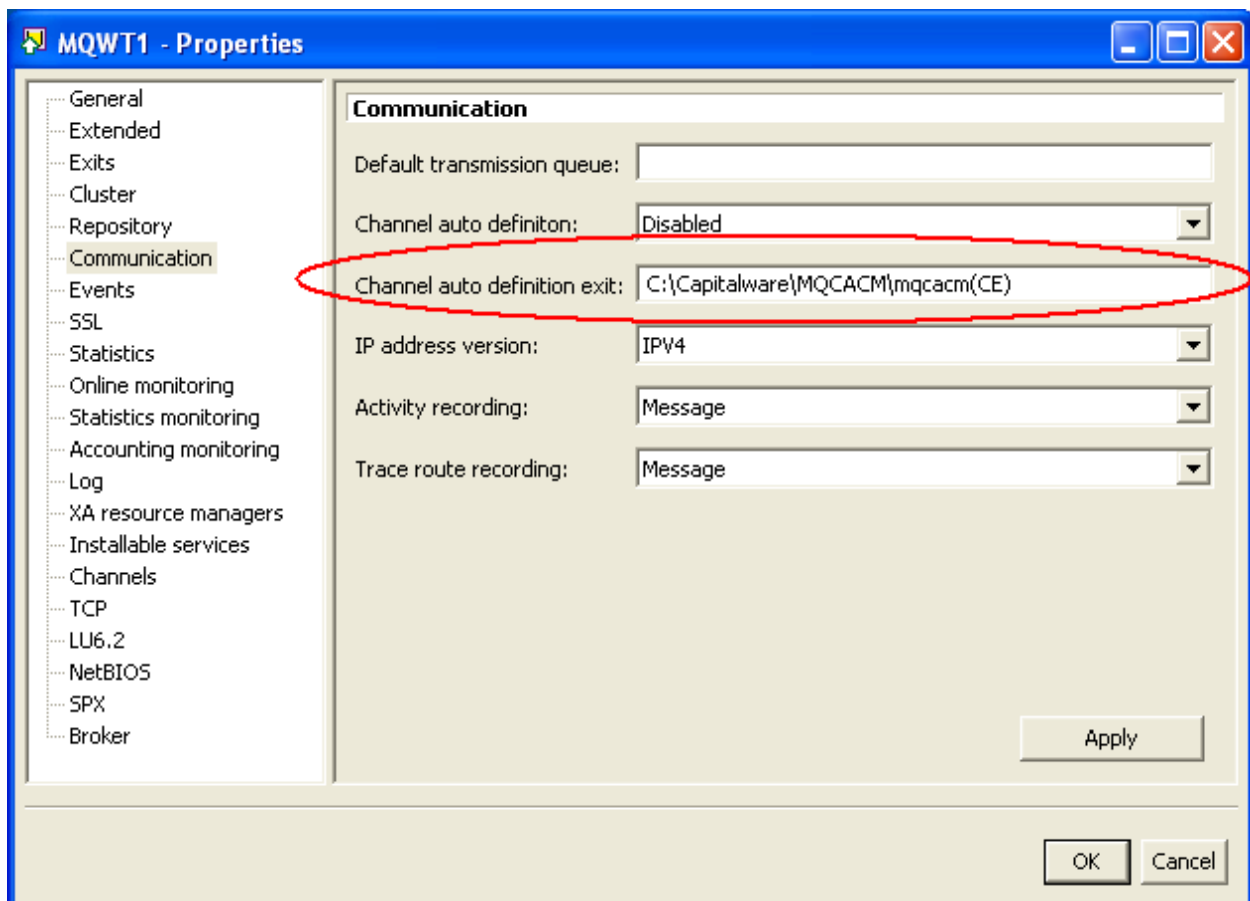
4.3.1 Windows

For Windows, CHADEXIT will contain the following values, assuming a default install:

- CHADEXIT
C:\Capitalware\MQCACM\mqcacm(CE)

The following is an example of an MQSC command:

```
ALTER QMGR CHADEXIT('C:\Capitalware\MQCACM\mqcacm(CE)')
```



4.3.2 Linux 32-bit

For Linux, CHADEXIT will contain the following values, assuming a default install:

- CHADEXIT
`/var/mqm/exits/mqcacm(CE)`

The following is an example of an MQSC command:

```
ALTER QMGR CHADEXIT('/var/mqm/exits/mqcacm(CE)')
```

4.3.3 Unix and Linux 64-bit

For Unix and Linux (excluding Linux x86), CHADEXIT will contain the following values, assuming a default install:

- CHADEXIT
`/var/mqm/exits64/mqcacm(CE)`

The following is an example of an MQSC command:

```
ALTER QMGR CHADEXIT('/var/mqm/exits64/mqcacm(CE)')
```

4.3.4 IBM i

For IBM i, CHADEXIT will contain the following values, assuming a default install:

- CHADEXIT
`MQCACM MQCACM`

The following is an example of an MQSC command:

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

4.3.5 z/OS

For z/OS, CHADEXIT will contain the following values, assuming a default install:

- CHADEXIT
MQCACM

The following is an example of an MQSC command:

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

4.4 Configuring a Cluster-Receiver Channel

This section describes the necessary entries to enable the server-side security exit on a Cluster-Receiver Channel. The server-side security exit and its data will be applied to 2 fields of the Cluster-Receiver Channel. The MQ Administrator will need to update these 2 fields of the Cluster-Receiver Channel.

For more information on server-side IniFile parameters, please review *Appendix A* of the *MQCACM Installation and Operation* manual.

The following examples below for Windows, Unix, Linux, IBM i and z/OS are for server-side only security exit implementations. We will use “BlockIP2” for the specific examples below but any 3rd party server-side security exits can be used. *Please read the BlockIP2 manual for more detailed information regarding the configuration and setup of BlockIP2.*

4.4.1 Windows

For Windows, SCYEXIT and SCYDATA will contain the following values, assuming a default install:

- SCYEXIT
BlockIP2(BlockExit)
- SCYDATA
FN=c:\path\Blockspec.txt;

The following is an example of an MQSC command for creating a Cluster-Receiver Channel with the server-side security exit and its data:

```
DEFINE CHANNEL ('TO.QMA') CHLTYPE(CLUSRCVR) +  
TRPTYPE(TCP) +  
CONNAME(127.0.0.1(1414) +  
SCYEXIT('BlockIP2(BlockExit)') +  
SCYDATA('FN=c:\path\Blockspec.txt;') +  
REPLACE
```

4.4.2 Linux 32-bit

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

- SCYEXIT
BlockIP2(BlockExit)
- SCYDATA
/var/mqm/exits/Blockspec.txt;

The following is an example of an MQSC command for creating a Cluster-Receiver Channel with the server-side security exit and its data:

```
DEFINE CHANNEL ('TO.QMA') CHLTYPE(RECEIVER) +  
  TRPTYPE(TCP) +  
  CONNAME(127.0.0.1(1414) +  
  SCYEXIT('BlockIP2(BlockExit)') +  
  SCYDATA('/var/mqm/exits/Blockspec.txt;') +  
  REPLACE
```

4.4.3 Unix and Linux 64-bit

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

- SCYEXIT
BlockIP2(BlockExit)
- SCYDATA
/var/mqm/exits64/Blockspec.txt;

The following is an example of an MQSC command for creating a Cluster-Receiver Channel with the server-side security exit and its data:

```
DEFINE CHANNEL ('TO.QMA') CHLTYPE(RECEIVER) +  
  TRPTYPE(TCP) +  
  CONNAME(127.0.0.1(1414) +  
  SCYEXIT('BlockIP2(BlockExit)') +  
  SCYDATA('/var/mqm/exits64/Blockspec.txt;') +  
  REPLACE
```

4.4.4 IBM i

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

- SCYEXIT
BLOCKIP2 BLOCKIP2
- SCYDATA
/QIBM/UserData/mqm/Blockspec.txt;

The following is an example of an MQSC command for creating a Cluster-Receiver Channel with the server-side security exit and its data:

```
DEFINE CHANNEL ('TO.QMA') CHLTYPE(RECEIVER) +  
  TRPTYPE(TCP) +  
  CONNAME(127.0.0.1(1414) +  
  SCYEXIT('BLOCKIP2 BLOCKIP2 ') +  
  SCYDATA('/QIBM/UserData/mqm/Blockspec.txt;') +  
  REPLACE
```

4.4.5 z/OS

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

- SCYEXIT
BLOCKIP2
- SCYDATA
FN=//DD:BLOCKDD;

The following is an example of an MQSC command for creating a Cluster-Receiver Channel with the server-side security exit and its data:

```
DEFINE CHANNEL ('TO.QMA') CHLTYPE(CLUSRCVR) +  
  TRPTYPE(TCP) +  
  CONNAME(127.0.0.1(1414) +  
  SCYEXIT('BLOCKIP2') +  
  SCYDATA('FN=//DD:BLOCKDD;') +  
  REPLACE
```


4.5 Configuring a Cluster-Sender Channel

There is no MQCACM configuration required for the Cluster-Sender channel, as the MQCACM exit will handle the setting of the values of the SCYEXIT and SCYDATA fields.

5 Appendix A – MQCACM IniFile Summary

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

```
LogMode=N
LogFile=C:\Capitalware\MQCACM\mqcacm.log
AllowSvrConn = Y
```

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 'CLUSDR'. 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].*.*[0-9]</p> <p>Note: Only used if UseAllowIP is set to 'Y'.</p>
AllowRcvr	<p>AllowRcvr specifies whether or not to enable the automatic channel creation for channel type 'RCVR'. AllowRcvr supports 2 values [Y / N]. The default value is N.</p> <p>e.g. AllowRcvr=Y</p>
AllowSvrConn	<p>AllowSvrConn specifies whether or not to enable the automatic channel creation for channel type 'SVRCONN'. AllowSvrConn supports 2 values [Y / N]. The default value is N.</p> <p>e.g. AllowSvrConn=Y</p>

Keyword	Description of Server-side keywords
ChadIniFilePath	<p>ChadIniFilePath specifies the location of the Channel Auto-Definition IniFile. The default is as follows:</p> <p>For Windows: ChadIniFilePath=C:\Capitalware\MQCACM\</p> <p>For IBM MQ 32-bit on Linux: ChadIniFilePath=/var/mqm/exits/</p> <p>For IBM MQ 64-bit on Unix and Linux: ChadIniFilePath=/var/mqm/exits64/</p> <p>For IBM MQ on IBM i: ChadIniFilePath=/QIBM/UserData/mqm/mqcacm/</p> <p>Note: Only used if UseChadIniFilePath is set to 'Y'.</p>
BackupLogFileCount	<p>BackupLogFileCount specifies the number of backup logfiles that MQCACM will be keeping. The default value is 9.</p> <p>e.g. BackupLogFileCount=9</p>
LogFile	<p>LogFile specifies the location of the log file. The default is as follows:</p> <p>For Windows: LogFile=C:\Capitalware\MQCACM\mqcacm.log</p> <p>For IBM MQ 32-bit on Linux: LogFile=/var/mqm/exits/mqcacm.log</p> <p>For IBM MQ 64-bit on Unix and Linux: LogFile=/var/mqm/exits64/mqcacm.log</p> <p>For IBM MQ on IBM i: LogFile=/QIBM/UserData/mqm/mqcacm/mqcacm.log</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 N.</p> <p>e.g. LogMode=N</p>

Keyword	Description of Server-side keywords
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>
RotateLogDaily	<p>RotateLogDaily specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y.</p> <p>e.g. RotateLogDaily=Y</p>
SetConnectionName	<p>SetConnectionName specifies whether or not the 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>SetConnectionName specifies whether or not the 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 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 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 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 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>
UseChadIniFilePath	<p>UseChadIniFilePath specifies a user supplied Path to Channel Auto-Definition IniFile will be set via the ChadIniFilePath keyword. UseChadIniFilePath supports 2 values [Y / N] where the default value is N.</p> <p>e.g. UseChadIniFilePath=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>

6 Appendix B – MQCACM Channel Auto-Definition IniFile Summary

A sample Channel Auto-Definition IniFile below is the *mqcacm.clussdr.ini* file supplied for Windows.

```
[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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</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 <i>mqcacm.ini</i> file.</p>

7 Appendix C – 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.

8 Appendix D – Notices

Trademarks:

AIX, IBM, MQSeries, OS/2 Warp, OS/400, iSeries, MVS, OS/390, WebSphere, IBM MQ and z/OS are trademarks of International Business Machines Corporation.

HP-UX is a trademark of Hewlett-Packard Company.

Intel is a registered trademark of Intel Corporation.

Java, J2SE, J2EE, Sun and Solaris are trademarks of Sun Microsystems Inc.

Linux is a trademark of Linus Torvalds.

Mac OS X is a trademark of Apple Computer Inc.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation.

UNIX is a registered trademark of the Open Group.

WebLogic is a trademark of BEA Systems Inc.