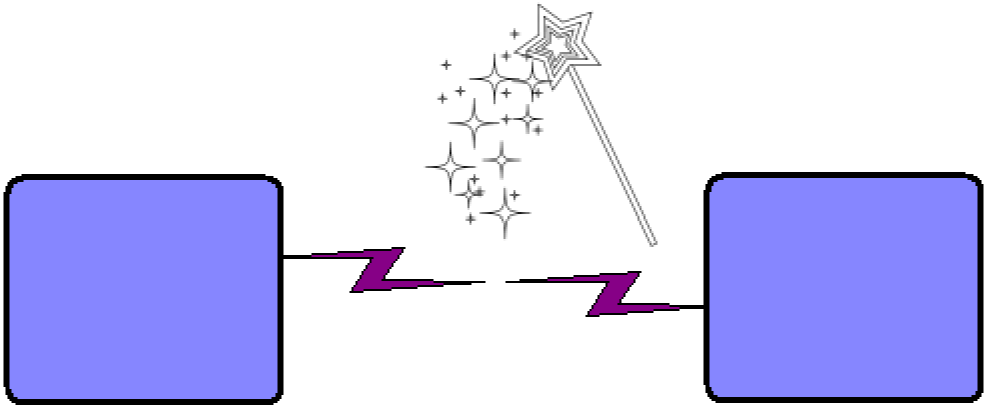
MQ Channel Auto Creation Manager Installation and Operation Manual



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

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

Table of Contents

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'.

The 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 MQCACM then they can purchase a support subscription from Capitalware.

On AIX, HP-UX, Linux, Solaris 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.

1.2 Executive Summary

MQCACM is an MQ CHAD exit.

MQCACM is available in 3 forms:

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

The major features of 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 provides the minimum supported software levels. These prerequisites apply to server-side installations of MQ Channel Auto Creation Manager.

1.3.1 Operating System

MQ Channel Auto Creation Manager can be installed on any of the following supported servers:

1.3.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.3.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.3.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.3.1.4 Linux x86

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

1.3.1.5 Linux x86_64 (64-bit)

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

1.3.1.6 Linux on POWER

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

1.3.1.7 Linux on zSeries (64-bit)

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

1.3.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.3.1.9 Sun Solaris

- Solaris SPARC v8 or higher
- Solaris x86_64 v10 or higher

1.3.1.10 Windows

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

1.3.2 IBM MQ

Sample distributions:

- IBM MQ v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 (32-bit and 64-bit)

Operating System	MQ v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
AIX v6.1 or higher	64-bit
HP-UX IA64 v11.23 or higher	64-bit
IBM i (OS/400)	64-bit
Linux x86	32-bit
Linux x86_64	64-bit
Linux on POWER	64-bit
Linux on zSeries	64-bit
Raspberry Pi ARM	32-bit
Solaris SPARC v10 & v11	64-bit
Solaris x86_64 v10 & v11	64-bit
Windows 2008, 2012, 2016 7, 8, 8.1 & 10	32-bit

1.3.3 Windows 64-bit

The following is the software prerequisite for Windows 64-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x64)
  https://www.microsoft.com/en-ca/download/details.aspx?id=14632

2 Installing MQCACM

This section describes how to install Capitalware's MQ Channel Auto Creation Manager.

2.1 MQCACM

2.1.1 Windows Installation

To install the MQCACM on Windows, first unzip the **mqcacm.zip** and then run the **mqcacm_setup.exe** file. Follow the on-screen instructions and MQCACM will be installed in the **C:\Capitalware\MQCACM\** directory (default installation).

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

2.1.2 Linux 32-bit Installation

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

- **Linux_x86/mqcacm_linux.tar**
- **RaspberryPi_ARM/mqcacm_raspberrypi_arm.tar**

Steps to install the MQCACM:

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

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

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

   ```
   chown mqm:mqm mqcacm
   chmod 550 mqcacm
   ```

2.1.3 Unix and Linux 64-bit Installation

To install the 64-bit version of MQCACM on Unix or Linux, first unzip the **mqcacm.zip** and then select the appropriate TAR file for the target platform. You will find 7 TAR files in the original ZIP file:

- **AIX/64-bit/mqcacm_aix53_64.tar**
- **AIX/64-bit/mqcacm_aix61_64.tar** for AIX v6.1 or higher
- **HPUX_IA64 /mqcacm_hpux64_ia64.tar**
- **Linux_x86_64/mqcacm_linux_x86_64.tar**
- **Linux_POWER/mqcacm_linux_power64.tar**
- **Linux_zSeries/64-bit/mqcacm_linux_zseries64.tar**
- **Solaris_SPARC/64-bit/mqcacm_solaris64.tar** for Solaris SPARC v8 and v9
- **Solaris_SPARC/64-bit/mqcacm_solaris10_64.tar** for Solaris SPARC v10 or higher
- **Solaris_x86_64/mqcacm_solaris_x86_64.tar**

Steps to install the MQCACM:

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

   ```
   cd /var/mqm/exits64/
   tar -xvf  mqcacm_xxx64.tar
   ```

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

   ```
   chown mqm:mqm mqcacm
   chmod 550 mqcacm
   ```

2.1.4 IBM i Installation

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

- **mqcacm.savf** is the IBM i 'Save File' that contains the library with MQCACM.
- **mqcacm_iseries.tar** is the IBM i IFS TAR file that contains a sample initialization file for MQCACM and sample MQSC script to define MQ channels with MQCACM

Steps to install the MQCACM:

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

```
CRTSAVF FILE(QGPL/MQCACM)
```

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

```
ftp -s:mqcacm_iseries.ftp  iseries_hostname
```

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

binary
cd QGPL
put mqcacm.savf

quote SITE NAMEFMT 1

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

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

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

3 Configuring MQCACM

This section describes how to configure MQCACM.

3.1 MQCACM Pathing

IBM MQ does not provide a mechanism for specifying the MQCACM IniFile. Therefore, MQCACM will look fr the IniFile in the default install directory or the MQAdmin can specify an environment variable to explicitly specify the path to the MQCACM IniFile.

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

3.1.1 Default Path

For Windows:

```
C:\Capitalware\MQCACM\
```

For IBM MQ 32-bit on Linux:

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

For IBM MQ 64-bit on Unix and Linux:

```
/var/mqm/exits64/
```

For IBM MQ on IBM i:

```
/QIBM/UserData/mqm/mqcacm/
```

3.1.2 MQCACM_HOME Environment Variable

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

E.g. Unix

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

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

3.2 Configuring MQCACM

This section describes how to configure the MQCACM.

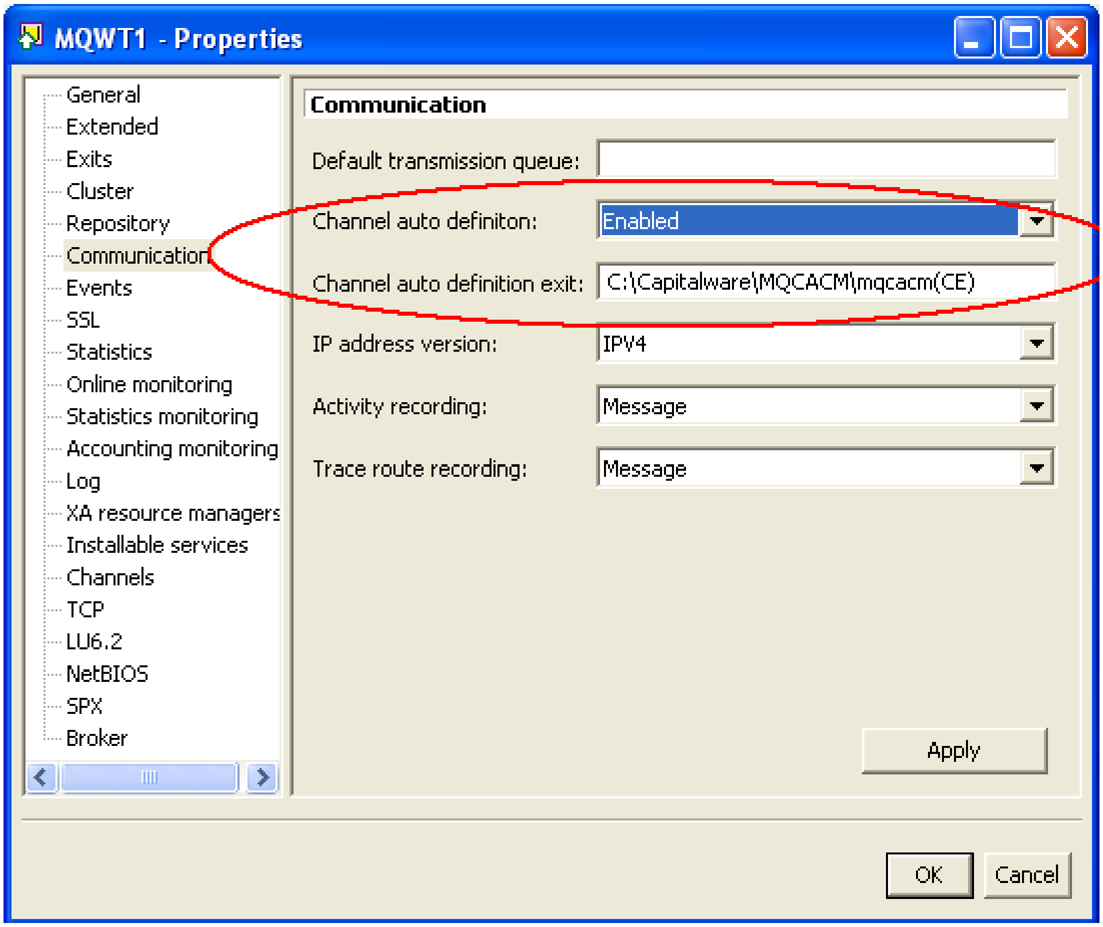
3.2.1 Windows

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



MQWT1 - Properties

Communication

Default transmission queue:

Channel auto definiton: Enabled

Channel auto definition exit: C:\Capitalware\MQCACM\mqcacm(CE)

IP address version: IPV4

Activity recording: Message

Trace route recording: Message

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

3.2.3 Unix and Linux 64-bit

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

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

The following is an example of an MQSC command:

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

3.2.4 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 CHAD(ENABLED)
ALTER QMGR CHADEXIT('MQCACM    MQCACM    ')
```

4 MQCACM IniFile Keywords

This section describes IniFile keywords.

4.1 Logging

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

1. **LogMode** specifies what type of logging the user wishes to have. LogMode supports 4 values [Q / N / V / D] where Q is Quiet, N is Normal, V is Verbose and D is Debug. The default value is N.

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

   For Windows:
   ```
   LogFile=C:\Capitalware\MQCACM\mqcacm.log
   ```

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

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

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

   Token Replacement for LogFile keyword:
   - **%QM%** - Substitutes the name of the queue manager
   - **%CHL%** - Substitutes the name of the channel
   - **%UID%** - Substitutes the UserID
   - **%PID%** - Substitutes the Process ID
   - **%TID%** - Substitutes the Thread ID

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

   The RotateLogDaily feature will keep up to 9 backup log files. The first connection request after midnight (and not at midnight) will cause it to roll / rotate the log files. If there are already 9 backup log files then the ninth backup log file will be delete and 8 becomes 9, 7 becomes 8, etc...

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

```
LogMode=N
LogFile=C:\Capitalware\MQCACM\mqcacm.log
RotateLogDaily=Y
BackupLogFileCount=9
```

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 Channel Auto-Definition IniFile Path

This section describes the necessary steps to allow the MQAdmin to define a different path to the Channel Auto-Definition IniFile. By default, the path will be the MQCACM install directory.

To specify a different Channel Auto-Definition IniFile Path, you need 2 keywords in the IniFile:

- **UseChadIniFilePath** set to Y if you are defining a different path to the Channel Auto-Definition IniFile
- **ChadIniFilePath** specifies the path (location) rather than the default path of the Channel Auto-Definition IniFile

```
UseChadIniFilePath=Y
ChadIniFilePath=c:\temp\chad\
```

4.5 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
```

4.6 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
```

4.7 AllowRcvr

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

```
AllowRcvr=Y
```

4.8 AllowSvrConn

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

```
AllowSvrConn=Y
```

4.9 SetConnectionName

This section describes the necessary steps to enable the overriding of the channel's CONNECTIONNAME field. SetConnectionName specifies whether or not the 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 MQCACM exit to override the CONNECTIONNAME field, set the SetConnectionName value to Y as shown below.

```
SetConnectionName=Y
```

4.10 SetMCAUser

This section describes the necessary steps to enable the overriding of the channel's MCAUSER field. SetMCAUser specifies whether or not the 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 MQCACM exit to override the MCAUSER field, set the SetMCAUser value to Y as shown below.

```
SetMCAUser=Y
```

4.11 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 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 MQCACM exit to override the MSGEXIT and MSGDATA fields, set the SetMessageExit value to Y as shown below.

```
SetMessageExit=Y
```

4.12 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 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 MQCACM exit to override the RCVEXIT and RCVDATA fields, set the SetReceiveExit value to Y as shown below.

```
SetReceiveExit=Y
```

4.13 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 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 MQCACM exit to override the SCYEXIT and SCYDATA fields, set the SetSecurityExit value to Y as shown below.

```
SetSecurityExit=Y
```

4.14 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 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 MQCACM exit to override the SENDEXIT and SENDDATA fields, set the SetSendExit value to Y as shown below.

```
SetSendExit=Y
```

5 MQCACM Channel Auto-Definition IniFile

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

5.1 Channel Auto-Definition IniFile Search Order

The MQCACM exit supports individual or shared configuration of the ***.clussdr.ini*** file for multiple queue managers on the same server. The MQCACM exit first searches for a Channel Auto-Definition IniFile with the prefix of the queue manager name. If successful, it will use that IniFile. Otherwise, the MQCACM exit uses the default Channel Auto-Definition IniFile with the name: ***mqcacm.clussdr.ini***.

e.g. If the queue manager name is QM1, the MQCACM exit searches for the ***qm1.clusdr.ini*** file. If that IniFile is not found, the MQCACM exit uses the Channel Auto-Definition IniFile with the name: ***mqcacm.clussdr.ini***.

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 Miscellaneous

This section describes the extra files that were included to help the user get MQCACM up and running in a very quick manner.

6.1 Windows

Sample IniFile

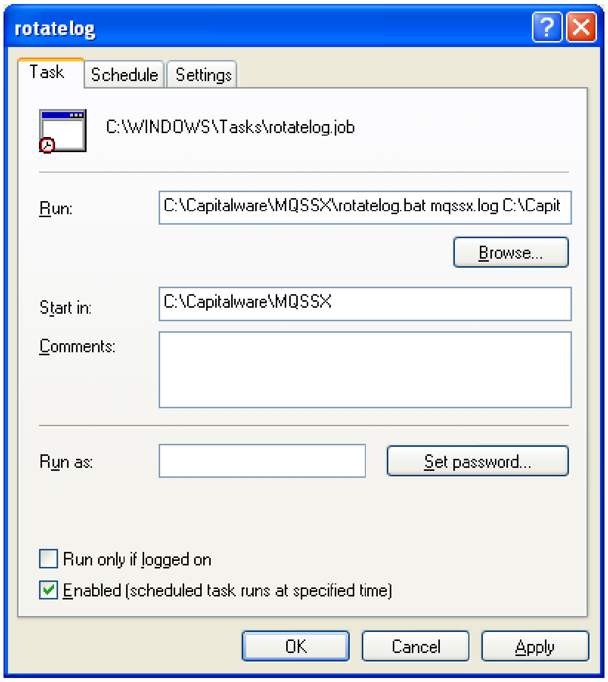
The '***mqcacm.ini***' file is a basic MQCACM IniFile. It has the standard IniFile parameters that the user may need to use or update. The '***mqcacm.ini.readme***' file is a plain text help file with a description of the parameters.

Sample MQSC scripts

The '***mqcacm.sample.mqsc***' file is a sample MQSC script to update the 2 system defined channels with the MQCACM information.

Rotate log script

The '***rotatelog.bat***' file is a Windows batch script to rotate (backup) the mqcacm.log file. Actually, it is generic in implementation; hence, it can be used to rotate any log file that the user wishes to be rotated. The batch script requires 2 parameters: log file name and the directory of log file.



C:\WINDOWS\Tasks\rotatelog.job

Run: C:\Capitalware\MQSSX\rotatelog.bat mqssx.log C:\Capit

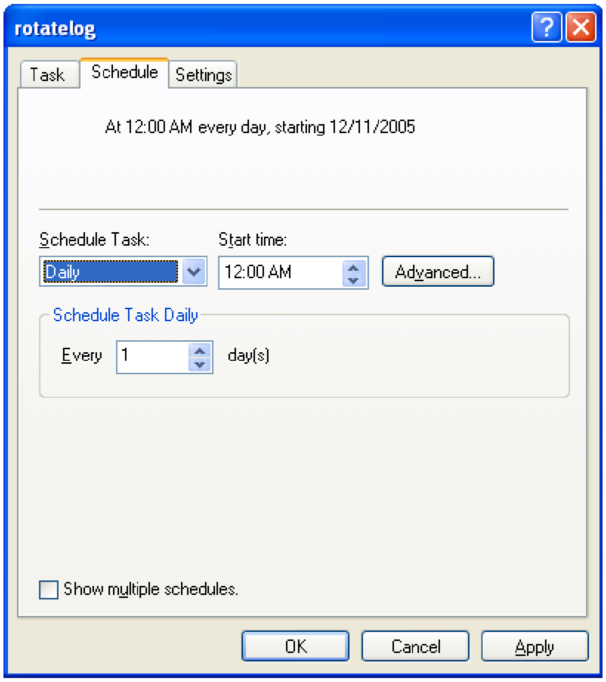
Start in: C:\Capitalware\MQSSX

Comments:

Run as:

□ Run only if logged on

☑ Enabled (scheduled task runs at specified time)



At 12:00 AM every day, starting 12/11/2005

Schedule Task: Daily

Start time: 12:00 AM

Schedule Task Daily

Every 1 day(s)

□ Show multiple schedules.

6.2 Unix and Linux

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

Sample MQSC scripts
The '***mqcacm.sample.mqsc***' file is a sample MQSC script to update the 2 system defined channels with the MQCACM information.

Rotate log script
The '***rotatelog.sh***' file is a Unix / Linux shell script to rotate (backup) the mqcacm.log file. Actually, it is generic in implementation; hence, it can be used to rotate any log file that the user wishes to be rotated. The shell script requires 2 parameters: log file name and the directory of log file.

Sample daily CRON entry for IBM MQ 32-bit on Unix and Linux:

0 0 * * * /var/mqm/exits/rotatelog.sh mqaux.log /var/mqm/exits/ > /tmp/mqcacm.log.run 2 > &1

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

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

6.3 IBM i

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

Sample MQSC scripts
The '***mqcacm.sample.mqsc***' file is a sample MQSC script to update the 2 system defined channels with the MQCACM information.

6.4 MQCACM Log File

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

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

7 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	**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. e.g. AllowClusRcvr=Y
AllowClusSdr	**AllowClusSdr** specifies whether or not to enable the automatic channel creation for channel type 'CLUSSDR'. AllowClusSdr supports 2 values [Y / N]. The default value is N. e.g. AllowClusSdr=Y
AllowIP	**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. e.g. AllowIP=192.168.*.1[0-5][0-9];127.0.0.?;10.*.*.[0-9] Note: Only used if UseAllowIP is set to 'Y'.
AllowRcvr	**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. e.g. AllowRcvr=Y
AllowSvrConn	**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. e.g. AllowSvrConn=Y

Keyword	Description of Server-side keywords
ChadIniFilePath	**ChadIniFilePath** specifies the location of the Channel Auto-Definition IniFile. The default is as follows: For Windows: ChadIniFilePath=C:\Capitalware\MQCACM\ For IBM MQ 32-bit on Linux: ChadIniFilePath=/var/mqm/exits/ For IBM MQ 64-bit on Unix and Linux: ChadIniFilePath=/var/mqm/exits64/ For IBM MQ on IBM i: ChadIniFilePath=/QIBM/UserData/mqm/mqcacm/ Note: Only used if UseChadIniFilePath is set to 'Y'.
BackupLogFileCount	**BackupLogFileCount** specifies the number of backup logfiles that MQCACM will be keeping. The default value is 9. e.g. BackupLogFileCount=9
LogFile	**LogFile** specifies the location of the log file. The default is as follows: For Windows: LogFile=C:\Capitalware\MQCACM\mqcacm.log For IBM MQ 32-bit on Linux: LogFile=/var/mqm/exits/mqcacm.log For IBM MQ 64-bit on Unix and Linux: LogFile=/var/mqm/exits64/mqcacm.log For IBM MQ on IBM i: LogFile=/QIBM/UserData/mqm/mqcacm/mqcacm.log
LogMode	**LogMode** specifies what type of logging the user wishes to have. LogMode supports 4 values [Q / N / V / D] where Q is Quiet, N is Normal, V is Verbose and D is Debug. The default value is N. e.g. LogMode=N

Keyword	Description of Server-side keywords
NamingStandard	**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. e.g. NamingStandard=TEST_???.CHL;ABC_*.CHL Note: Only used if UseNamingStandard is set to 'Y'.
RotateLogDaily	**RotateLogDaily** specifies whether or not daily log file rotation should take place. RotateLogDaily supports 2 values [Y / N]. The default value is Y. e.g. RotateLogDaily=Y
SetConnectionName	**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. e.g. SetConnectionName=Y
SetMCAUser	**SetMCAUser** specifies whether or not the MQCACM exit will override the MCAUSER field. SetMCAUser supports 2 values [Y / N] where the default value is N. e.g. SetMCAUser=Y
SetMessageExit	**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. e.g. SetMessageExit=Y
SetReceiveExit	**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. e.g. SetReceiveExit=Y

Keyword	Description of Server-side keywords
SetSecurityExit	**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. e.g. SetSecurityExit=Y
SetSendExit	**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. e.g. SetSendExit=Y
UseAllowIP	**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. e.g. UseAllowIP=Y
UseChadIniFilePath	**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. e.g. UseChadIniFilePath=Y
UseNamingStandard	**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. e.g. UseNamingStandard=Y

8 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	**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. e.g. ConName=127.0.0.1(1415) Note: Only used if SetConnectionName is set to 'Y' in the ***mqcacm.ini*** file.
MCAUser	**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. e.g. MCAUser=mqtest Note: Only used if SetMCAUser is set to 'Y' in the ***mqcacm.ini*** file.
Partner	**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. e.g. Partner=QM5

Keyword	Description of keywords
ScyData	**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. Example: ScyData= Note: Only used if SetSecurityExit is set to 'Y' in the ***mqcacm.ini*** file.
ScyExit	**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. Example: ScyExit= Note: Only used if SetSecurityExit is set to 'Y' in the ***mqcacm.ini*** file.
MsgData	**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. e.g. MsgData =SampleMessageData Note: Only used if SetMessageExit is set to 'Y' in the ***mqcacm.ini*** file.
MsgExit	**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. e.g. MsgExit=SampleExit Note: Only used if SetMessageExit is set to 'Y' in the ***mqcacm.ini*** file.
RcvData	**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. e.g. RcvData=SampleMessageData Note: Only used if SetReceiveExit is set to 'Y' in the ***mqcacm.ini*** file.

Keyword	Description of keywords
RcvExit	**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. e.g. RcvExit =SampleExit Note: Only used if SetReceiveExit is set to 'Y' in the ***mqcacm.ini*** file.
SendData	**SendData** specifies a value in order to override the current SENDDATA field in the channel auto-definition. SendDatais optional. If the keyword is not specified then no override is performed. e.g. SendData=SampleMessageData Note: Only used if SetSendExit is set to 'Y' in the ***mqcacm.ini*** file.
SendExit	**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. e.g. SendExit =SampleExit Note: Only used if SetSendExit is set to 'Y' in the ***mqcacm.ini*** file.

9 Appendix C – MQCACM Upgrade Procedures

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

9.1.1 Windows Upgrade

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

9.1.2 Linux 32-bit Upgrade

- Login under the mqm account
- Stop the queue manager.
- Backup all MQCACM IniFiles in the MQCACM install directory
- Copy the appropriate tar file to the ***/var/mqm/exits/*** directory
- Un-tar the contents of the tar file.

  i.e. For AIX, do the following command:

  ```
  tar -xvf  mqcacm_linux.tar
  ```
- Run the script as follows:

  ```
  ./setssx.sh
  ```
- Restore the MQCACM IniFiles if they were altered / deleted.
- Delete the MQCACM tar file
- Restart the queue manager

9.1.3 Unix and Linux 64-bit Upgrade

- Stop the queue manager.
- Backup all MQCACM IniFiles in the MQCACM install directory
- Copy the appropriate tar file to the ***/var/mqm/exits64/*** directory
- Un-tar the contents of the tar file.
  i.e. For AIX, do the following command:
  ```
     tar -xvf  mqcacm_aix.tar
  ```
- Run the script as follows:
  ```
    ./setssx.sh
  ```
- Restore the MQCACM IniFiles if they were altered / deleted.
- Delete the MQCACM tar file
- Restart the queue manager

9.1.4 IBM i Upgrade

- Stop the queue manager.
- Backup all MQCACM IniFiles in the MQCACM install directory
- ftp the IBM i files to the IBM i server as follows:

```
ftp -s:mqcacm_iseries.ftp  iseries_hostname
```

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

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

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

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

- Restore the MQCACM IniFiles if they were altered / deleted.
- Restart the queue manager

10 Appendix D – Support

The support for MQ Channel Auto Creation Manager 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: MQCACM Support
Unit 11, 1673 Richmond Street, PMB524
London, Ontario N6G2N3
Canada

11 Appendix E – Summary of Changes

- MQ Channel Auto Creation Manager v1.0.6
  - Enhanced the code for dumping the pointers passed into exit.
  - Fixed an issue in the subroutine that removes trailing blanks
  - Fixed an issue with default exit path

- MQ Channel Auto Creation Manager v1.0.5
  - Added code to append trailing slash for ExitPath & ChadIniFilePath if it is missing.
  - Tuned the code that is called on entry
  - Tuned the logging code

- MQ Channel Auto Creation Manager v1.0.4
  - Fixed an issue in the logging framework where a constant was being modified.

- MQ Channel Auto Creation Manager v1.0.3
  - Enhanced logging - the LogFile keyword now supports the following tokens: %QM%, %CHL%, %UID%, %PID% & %TID%

- MQ Channel Auto Creation Manager v1.0.2
  - Fixed an issue with the Ini Processor not finding next section
  - Fixed an issue on Windows with freeing environment variable memory (error with FreeEnvironmentStrings Windows API call)
  - Fixed an issue with using "size_t" variable type when it should have been "int"

- MQ Channel Auto Creation Manager v1.0.1
  - Improved the IniFile processing speed.

- MQ Channel Auto Creation Manager v1.0.0
  - Initial release.

12 Appendix F – 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.

13 Appendix G – 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.