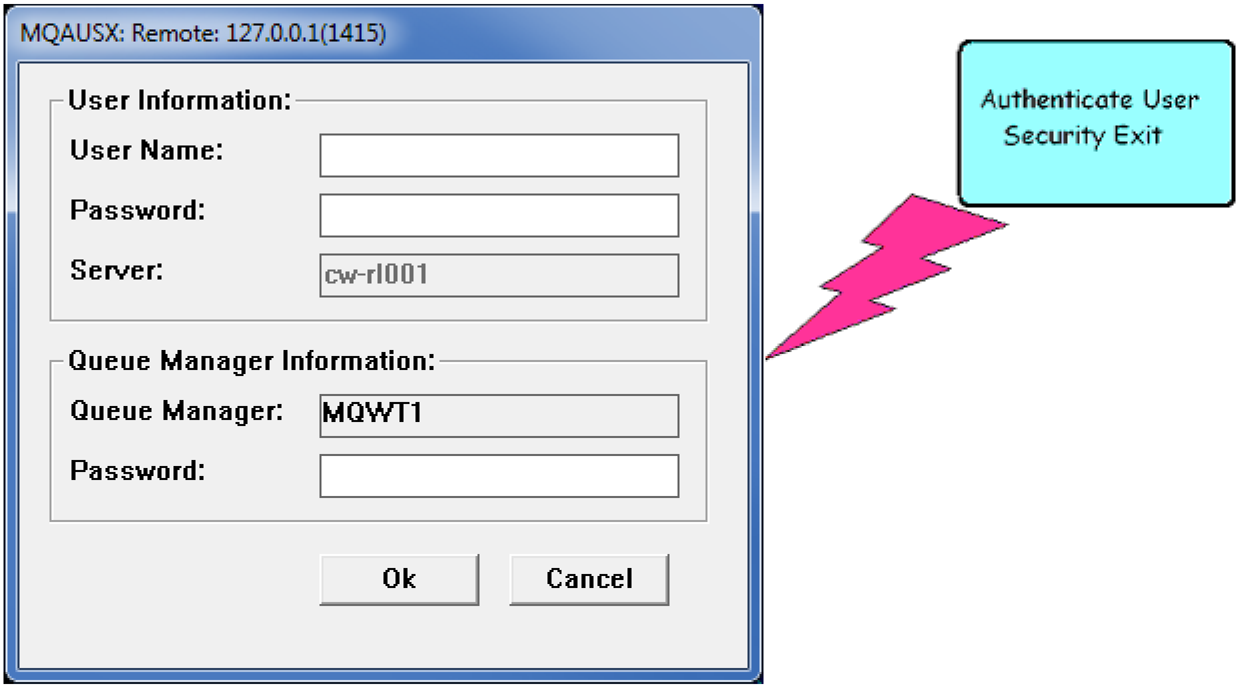
MQAUSX Client-side Configuration Manual



MQAUSX: Remote: 127.0.0.1(1415)

User Information:

User Name:

Password:

Server: cw-rl001

Queue Manager Information:

Queue Manager: MQWT1

Password:

Authenticate User Security Exit

Capitalware Inc.

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

Last Updated: January 2021.

Table of Contents

1 Introduction

1.1 Overview

MQ Authenticate User Security Exit (MQAUSX) is solution that allows a company to fully authenticate a user who is accessing a IBM MQ resource. It authenticates the user's UserId and Password (and possibly Domain Name) against the server's native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file.

The security exit will operate with IBM MQ v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix and Linux environments. It works with Server Connection, Client Connection, Sender, Receiver, Server and Requester channels of IBM MQ queue manager.

The MQ Authenticate User Security Exit solution is comprised of 2 components: client-side security exit and server-side security exit.

1.1.1 Client-Side Security Exit

The ***client-side security exit*** first checks if the server-side exit is defined for the particular channel. The client-side exit will receive a security token to be used in the encryption process of the user's password. It will prompt the user for his / her UserId and Password (and domain name for Windows), encrypt the data and send it to the server-side security exit.

For each connection attempt, the server-side security exit will verify that it is an acceptable client exit attempting the connection. If so, then the server-side will send a unique security token. When the server-side security exit receives the encrypted data, it will decrypt the incoming data and then perform UserId and Password (and domain) authentication against the native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file. If successful, the connection will be allowed.

1.1.2 Server-Side Security Exit

The ***server-side security exit*** supports the concept of 'Proxy IDs'. After a user has been successfully authenticated against the native OS system, LDAP server, Microsoft's Active Directory, Quest Authentication Services, Centrify's DirectControl, Unix/Linux PAM (Pluggable Authentication Module) or an encrypted MQAUSX FBA file and the 'Proxy Mode' flag is set, then the server-side security exit will look up the user's UserID in the Proxy file for their Proxy ID. The Proxy ID will be used for all MQ interactions.

An MQAdmin can define a password for a queue manager. Hence, when enabled, a back-end application and/or end-user would need to not only know their UserID and Password but also the queue manager’s Password to successfully log in. Defining and requiring a queue manager Password in MQAUSX is equivalent to adding perimeter security to your system.

The server-side security exit has the ability to allow or restrict users from logging in with the 'mqm' or 'MUSR_MQADMIN' or 'QMQM' UserIDs. This is controlled by the server-side security exit's property keyword 'Allowmqm'.

The server-side security exit has the capability to allow or limit the incoming channel connections according to the name of the associated Server Connection channel (SVRCONN). Each Server Connection channel can be allocated a maximum number of connections and the server-side security exit will ensure that this maximum is not exceeded.

Client connections to a queue manager are limited by either channel name or the 'DefaultMCC' property keyword in the initialization file. In today's use of J2EE applications, it is a possibility that one J2EE application could overwhelm the queue manager with client connections, thus preventing any connections being made from other applications.

The MQAdmin can enable Excessive Client Connections alerting system that counts the number of connections over a period of time (i.e. Day / Hour / Minute) and writes a message to the log when the count exceeds a particular value. If the keyword WriteToEventQueue is set to ‘Y’ then an event message is also written to an event queue. ECC feature is designed to catch applications that are poorly written, for example, applications that continuously connect and disconnect from the queue manager for every message sent or received.

The server-side security exit has the ability to allow or restrict the incoming IP address, hostname and/or SSL DN. The server-side security exit uses a regular expression parser to parse the incoming client IP address, hostname, and/or SSL DN against a predefined regular expression pattern.

The server-side security exit has the ability to allow or restrict the incoming UserID against a group. A list of groups can be queried for the incoming UserID. The groups can be in the local OS or a group file. If MQAUSX is authenticating against an LDAP server then the group querying can be against the LDAP server.

For those channels where authentication is not required, the server-side security exit can be set to not perform this function. This is controlled by the server-side security exit's property keyword 'NoAuth'.

The server-side security exit, when in non-authentication mode, has the ability to allow or restrict users from connecting with a blank UserID value. This is controlled by the server-side security exit's property keyword 'AllowBlankUserID'.

The server-side security exit, when in non-authentication mode, has the ability to allow or restrict the incoming UserID. The server-side security exit uses a regular expression parser to parse the incoming client UserID against a predefined regular expression pattern.

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.

MQAUSX is 4 products in 1

1. If the client application is configured with the client-side security exit then the user credentials are encrypted and sent to the remote queue manager. This is the best level of security.

2. If the client application is not configured with the client-side security exit and the client-side **AND** server-side are at MQ V8 then MQ V8 will encrypt the user credentials as they flow from the client application to the queue manager.

3. If the client application is not configured with the client-side security exit then the user credentials are sent in plain text to the remote queue manager. This feature is available for Java/JMS, Java and C# DotNet client applications. For native applications (i.e. C/C++), then the application must use and populate the MQCSP structure with the UserID and Password.
    - Using MQAUSX with No Client-side Security Exit - Part 1 (coding examples) http://www.capitalware.com/rl_blog/?p=638
    - Using MQAUSX with No Client-side Security Exit - Part 2 (configuring tools like MQ Explorer, SupportPac MO71, MQ Visual Edit, etc..) http://www.capitalware.com/rl_blog/?p=659

4. If the MQAdmin sets the MQAUSX IniFile parameter NoAuth to Y then it functions just like MQ Standard Security Exit (MQSSX). MQSSX does not authenticate. It filters the incoming connection based on UserID, IP address, hostname and/or SSL DN.

2 Installing MQ Authenticate User Security Exit Client-side

This section describes how to install Capitalware's MQ Authenticate User Security Exit.

2.1 Client-side Security Exit

Currently, there are 6 client-side security exits. One is for Java based applications using any platform; two are for Windows programs; and the other one is for Unix or Linux.

- **mqausxclnt.dll** is for Windows 32-bit based executables and it is the client-side security exit that will prompt the user for the user's UserId and Password (and domain name) that will be invoked by the MQ Client component. This client-side exit can also operate in batch mode.
- **mqausxdn.dll** is for Windows 32-bit managed .NET based executables and it is the client-side security exit that will be invoked by the MQ Client component. The Unix or Linux client-side exit can ONLY operate in batch mode.
- **64\mqausxclnt.dll** is for Windows 64-bit based executables and it is the client-side security exit that will prompt the user for the user's UserId and Password (and domain name) that will be invoked by the MQ Client component. This client-side exit can also operate in batch mode.
- **64\mqausxdn.dll** is for Windows 64-bit managed .NET based executables and it is the client-side security exit that will be invoked by the MQ Client component. The Unix or Linux client-side exit can ONLY operate in batch mode.
- **enc_clnt_gui.exe** is a Windows GUI program and it is used to create a file that contains the UserId, encrypted Password and remote ServerName.
- **enc_clnt.exe** is a Windows program and it is used to create a file that contains the UserId, encrypted Password and remote ServerName.
- **ccdte.exe** is a Windows GUI program and it is used to create / update / delete CLNTCONN channels in a client channel definition table.
- **mqausxclnt** is for Unix or Linux based executables and it is the client-side security exit that will be invoked by the MQ Client component. The Unix or Linux client-side exit can ONLY operate in batch mode.
- **enc_clnt** is a Unix or Linux program and it is used to create a file that contains the UserId, encrypted Password and remote ServerName.
- **MQAUSXJ.jar** is the actual client-side security exit that will prompt the user for the user's UserId and Password (and domain name) that will be invoked by the MQ Client component. It requires Java v1.3 or higher.
- **SetupMQAUSXE6.sh** is a simple Linux shell script to unzip the MQAUSXJ.jar file for use with MQ Explorer v6.0.
- **SetupMQAUSXE6.bat** is a simple Windows batch file to unzip the MQAUSXJ.jar file for use with MQ Explorer v6.0.
- **mqexplorer_v6.sample.mqsc** is a sample MQSC script to define Client Connection channel for MQ Explorer v6.0.
- **sx_def.bat** is a simple Windows batch file that uses SupportPac MO72 to issue a MQSC command against a client channel definition table.
- **sx_dis.bat** is a simple Windows batch file that uses SupportPac MO72 to display CLNTCONN channel entries from a client channel definition table.

2.1.1 Windows Installation

To install the client-side security exit on Windows, first unzip the **mqausx.zip** and then run the ***mqausx-client-setup.exe*** file from the ***Windows-Client*** directory. Follow the on-screen instructions and the security exit will be installed in the **C:\Capitalware\MQAUSX\** directory (default installation).

3 Configuring MQAUSX Client-side Security Exit

3.1 Configuring Security Exit in MQ Explorer MQ v5.2 or v5.3

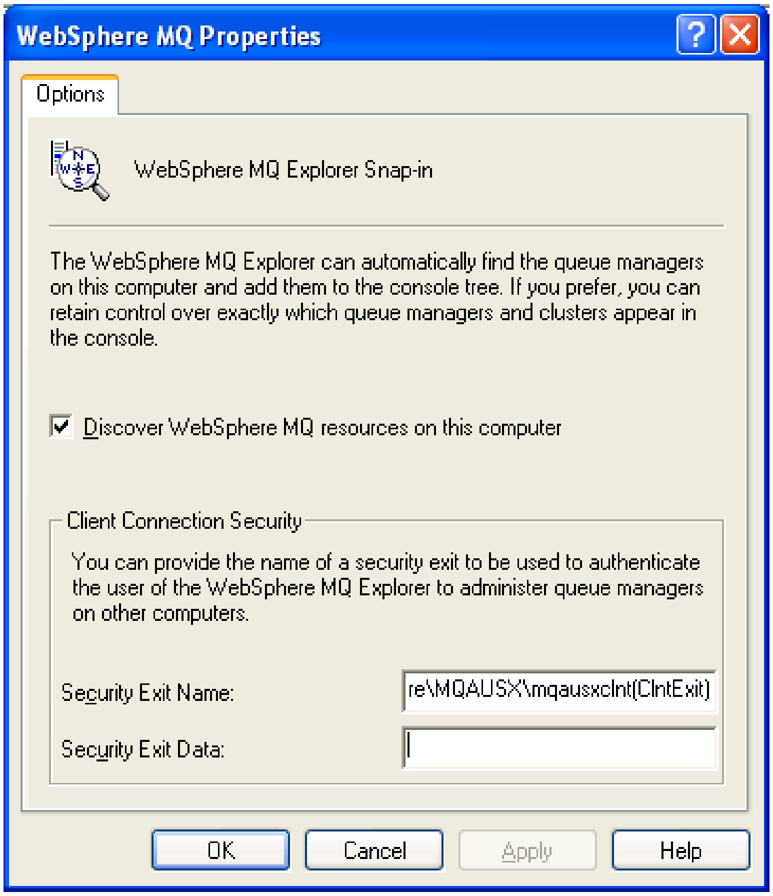
This section describes the necessary steps to enable Security Exits in IBM MQ Explorer v5.2 or v5.3.

3.1.1 GUI popup window for MQ Explorer v5.2 or v5.3

To enable user-defined client-side security exit for authentication do the following steps:

1. Open MQ Explorer v5.2 or v5.3
2. In the left panel, select **IBM MQ** under Console Root or **IBM MQSeries** for v5.2
3. Right-click and select Properties
4. In the **Security Exit Name** field input:
   `C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)`

5. Click Ok on the IBM MQ window



WebSphere MQ Properties

WebSphere MQ Explorer Snap-in

The WebSphere MQ Explorer can automatically find the queue managers on this computer and add them to the console tree. If you prefer, you can retain control over exactly which queue managers and clusters appear in the console.

☑ Discover WebSphere MQ resources on this computer

Client Connection Security

You can provide the name of a security exit to be used to authenticate the user of the WebSphere MQ Explorer to administer queue managers on other computers.

Security Exit Name: re\MQAUSX\mqausxclnt(ClntExit)

Security Exit Data:

3.1.2 Batch or Quiet mode for MQ Explorer MQ v5.2 or v5.3

Each time the user connects to the queue manager, they will be prompted for their UserId and Password (and Server Name). To run in batch or quiet mode, the user can explicitly set the UserId and Password in the channel's SecurityUserData or specify a file in the SecurityUserData that will contain the UserId and Password.

To explicitly set the UserId and Password values do the following for the user-defined client-side security exit for authentication:

- In the left panel, select **IBM MQ** under Console Root or **MQSeries** for v5.2
- Right-click and select Properties
- In the **Security Exit Name** field input:
  ```
  C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
  ```
- In the **Security Exit Data** field input:
  ```
  u=fred;p=abcdef;s=ABC123
  ```
- Click Ok on theIBM MQ window

To specify a file that will contain the UserId and Password values do the following for the user-defined client-side security exit for authentication:

- In the left panel, select **IBM MQ** under Console Root or **MQSeries** for v5.2
- Right-click and select Properties
- In the **Security Exit Name** field, input the following:
  ```
  C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
  ```
- In the **Security Exit Data** field, input the following (read Appendix A for the format of the file):
  ```
  C:\Capitalware\MQAUSX\clnt.ini
  ```

  Or use an encrypted file. (see Appendix B for more information)
  ```
  C:\Capitalware\MQAUSX\clnt.enc
  ```
- Click Ok on theIBM MQ window

Note: Security User Data must NOT exceed 32 characters.

3.2 Configuring Security Exit in MQ Explorer v6.0

This section describes the necessary steps to enable the Client-side Security Exit in MQ Explorer v6.0 for Windows or Linux. MQ Explorer v6.0 only supports Client-side Security Exit with Refresh Pack01 (FP01) or higher. The user can verify the version they are using by typing the command 'dspmqver'. The version should read 6.0.1.0 or higher.

3.2.1 Local One Time Setup

To use the MQAUSX client-side security with MQ Explorer v6.0, the user must do a one time setup by running the following command:

For Windows:

```
C:\Capitalware\MQAUSX\SetupMQAUSXE6.bat
```

For Linux (32-bit):

```
/var/mqm/exits/SetupMQAUSXE6.sh
```

For Linux (64-bit):

```
/var/mqm/exits64/SetupMQAUSXE6.sh
```

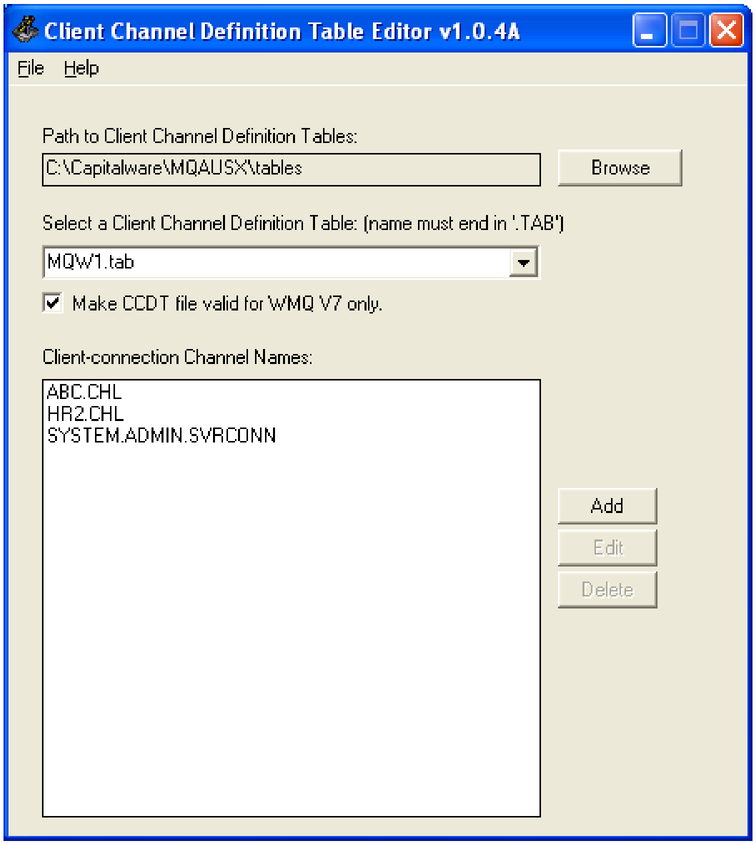
3.2.2 Remote One Time Setup for All Non-version 6 Queue Managers

To use MQ Explorer v6.0 with remote queue managers that are not at v6.0, the MQ Admin must create a Model Queue called 'SYSTEM.MQEXPLORER.REPLY.MODEL'. This must be done to all non-version 6 queue managers.

```
DEFINE QMODEL ('SYSTEM.MQEXPLORER.REPLY.MODEL') +
       PUT(ENABLED) +
       DEFPRTY(0) +
       DEFPSIST(NO) +
       GET(ENABLED) +
       DEFTYPE(TEMPDYN) +
       MAXDEPTH(5000) +
       MAXMSGL(4194304) +
       NOSHARE +
       DEFSOPT(EXCL) +
       MSGDLVSQ(PRIORITY) +
       USAGE(NORMAL) +
       NOTRIGGER +
       PROCESS(' ') +
       INITQ(' ') +
       REPLACE
```

3.2.3 Creating a Client Channel Definition Table Entry

MQ Explorer v6.0 requires the user to create a client channel definition table to use a client-side security exit. To enable user-defined client-side security exit for authentication, do the following steps:



Path to Client Channel Definition Tables: C:\Capitalware\MQAUSX\tables

Select a Client Channel Definition Table: (name must end in '.TAB') MQW1.tab

☑ Make CCDT file valid for WMQ V7 only.

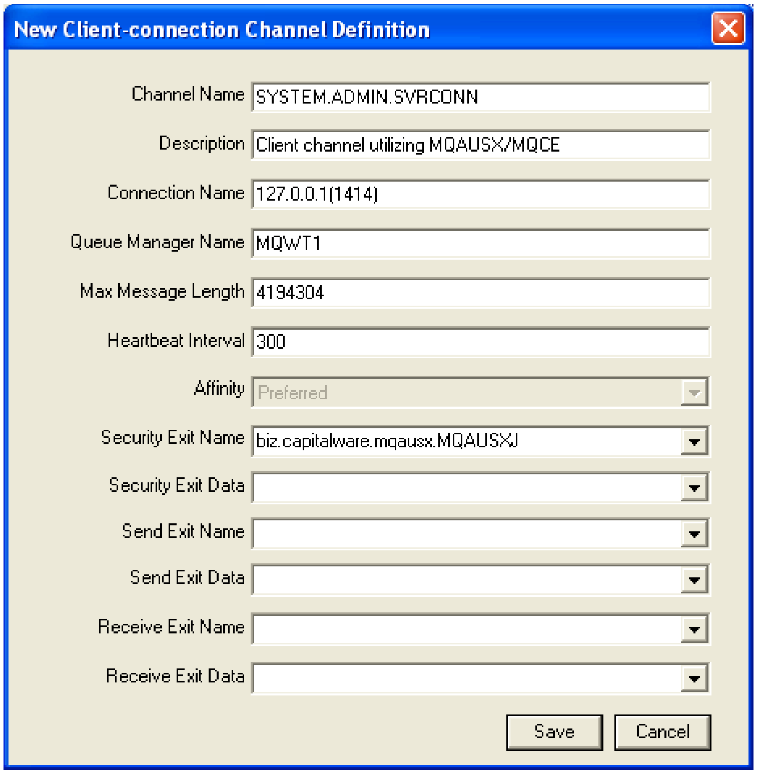
Client-connection Channel Names:

```
ABC.CHL
HR2.CHL
SYSTEM.ADMIN.SVRCONN
```

1 Start the Client Channel Definition Table Editor (From the ***Start*** -> ***All Programs*** menu)

2 Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')

3 Click the Add button to insert a new CLNTCONN channel or click the Edit button to edit an existing CLNTCONN channel.



New Client-connection Channel Definition

Channel Name: SYSTEM.ADMIN.SVRCONN

Description: Client channel utilizing MQAUSX/MQCE

Connection Name: 127.0.0.1(1414)

Queue Manager Name: MQWT1

Max Message Length: 4194304

Heartbeat Interval: 300

Affinity: Preferred

Security Exit Name: biz.capitalware.mqausx.MQAUSXJ

Security Exit Data:

Send Exit Name:

Send Exit Data:

Receive Exit Name:

Receive Exit Data:

4 For ***Security Exit Name***, select ***biz.capitalware.mqausx.MQAUSXJ*** from the drop-down list.

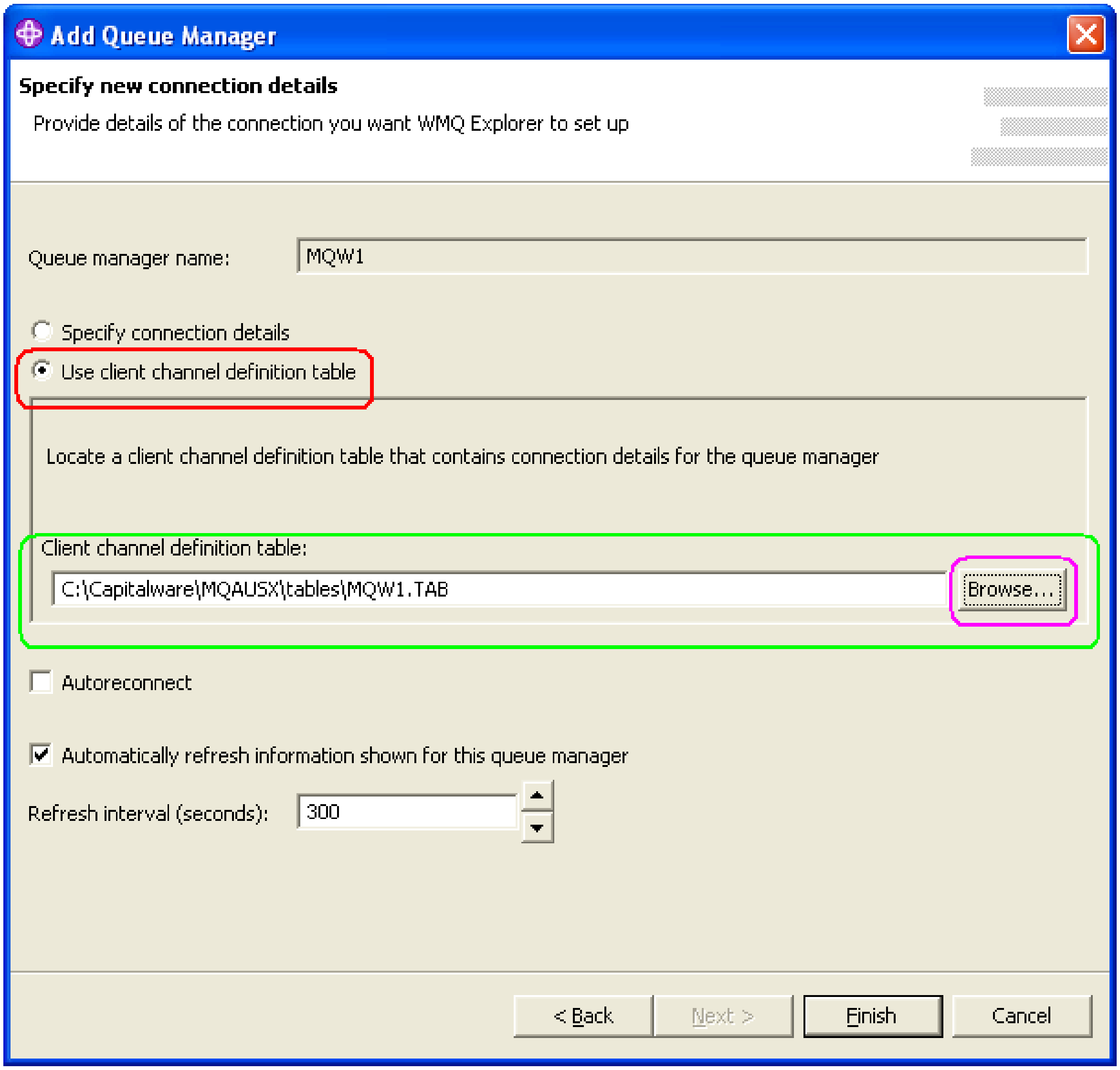
A client channel definition table will be created in the 'tables' directory under the default install directory.

For the example above, a client channel definition table will be found (assuming a default install) at this location:

```
C:\Capitalware\MQAUSX\tables\MQW1.TAB
```

3.2.4 Adding a Queue Manager using a client channel definition table

- Open MQ Explorer v6.0
- In the left panel, right click on ***Queue Managers*** and select ***Show/Hide Queue Manager***
- Click the ***Add*** button
- Fill in the ***Queue manager name*** and click ***Next***
- Click the ***Use client channel definition table*** radio button and then click the ***Browse*** button to select the appropriate client channel definition table from the C:\Capitalware\MQAUSX\tables\ directory.
- Click the ***Finish*** button



Add Queue Manager

Specify new connection details

Provide details of the connection you want WMQ Explorer to set up

Queue manager name: MQW1

○ Specify connection details

● Use client channel definition table

Locate a client channel definition table that contains connection details for the queue manager

Client channel definition table: C:\Capitalware\MQAUSX\tables\MQW1.TAB

☐ Autoreconnect

☑ Automatically refresh information shown for this queue manager

Refresh interval (seconds): 300

3.2.5 IBM APAR IC52821

Recently, an issue was discovered with IBM's MQ Explorer v6.0.2.0 or v6.0.2.1 or v6.0.2.2. This issue affects the use of any client-side security exits including MQAUSX. IBM has fixed the issue. The fix will be included in the MQ Explorer v6.0.2.3 and higher releases.

http://www.ibm.com/support/docview.wss?rs=0&q1=ic52821&uid=swg1IC52821&loc=en_US&cs=utf-8&cc=us&lang=en

Warning: Please exit MQ Explorer before applying the fix.

If you are using MQ Explorer v6.0.2.0 or v6.0.2.1 or v6.0.2.2, you will need to apply APAR IC52821 to fix the program. A copy of the fixed JAR file has been included in the directory, **APAR\MQ_v6\IC52821**, which can be found on the MQAUSX CD and in the MQAUSX download file.

The steps to apply the fix are as follows:

1. Close MQ Explorer v6 if it is currently running.
2. Navigate to ***<MQ_Install>\eclipse\plugins***
   eg: ***C:\Program Files\IBM\IBM MQ\eclipse\plugins***
3. For v6.0.2.0: Open folder **com.ibm.mq.runtime_6.0.2.0\lib**
4. For v6.0.2.1: Open folder **com.ibm.mq.runtime_6.0.2.1\lib**
5. For v6.0.2.2: Open folder **com.ibm.mq.runtime_6.0.2.2\lib**
6. Back up the existing **com.ibm.mq.jar**
7. Copy the patched **com.ibm.mq.jar**
8. Restart MQ Explorer

The above commands are included in a Windows batch script called: ***fix_IC52821.bat***. To execute fix_IC52821.bat script, go to C:\Capitalware\MQAUSX and then run:

```
C:\Capitalware\MQAUSX\APAR\MQ_v6\IC52821\fix_IC52821.bat
```

3.3 Configuring Security Exit in MQ Explorer v7.0, v7.1, v7.5, v8, v9, v9.1 & v9.2

This section describes the necessary steps to enable the Client-side Security Exit in MQ Explorer v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 for Windows or Linux. MQ Explorer v7.0 only supports Client-side Security Exit with PMR IC58936 applied or Fix Pack 7.0.0.2 or higher.

3.3.1 Directly using Class Name and Classpath

3.3.1.1 GUI popup window for MQ Explorer v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2

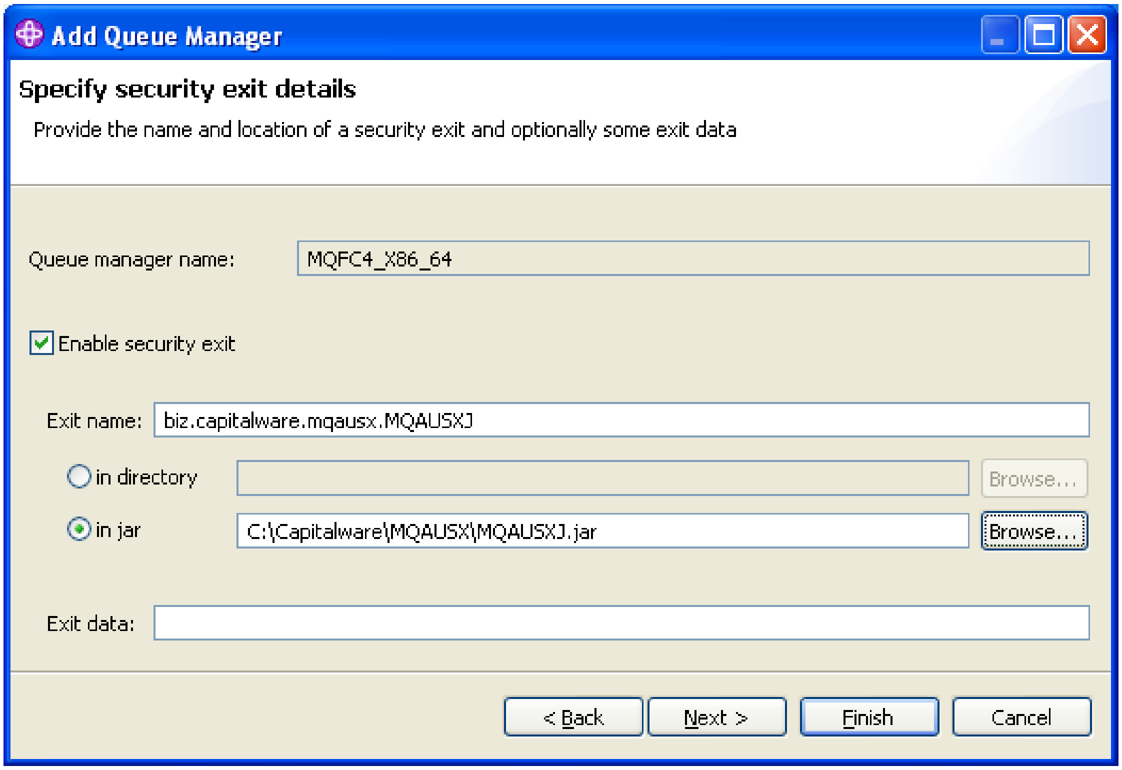
To enable user-defined client-side security exit for authentication, do the following steps:

- Open MQ Explorer v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
- In the left panel, select **Queue Managers** under IBM IBM MQ
- Right-click and select **Add Remote Queue Manager**
- Input the new queue manager name and click **Next**
- Input the hostname, port number and channel name and click **Next**
- Click (enable) the checkbox **Enable security exit**
- In the **Exit Name** field, input the following:

```
biz.capitalware.mqausx.MQAUSXJ
```

- Select the **in jar** radio button and input the following:

```
C:\Capitalware\MQAUSX\MQAUSXJ.jar
```



Add Queue Manager

Specify security exit details

Provide the name and location of a security exit and optionally some exit data

Queue manager name: MQFC4_X86_64

☑ Enable security exit

Exit name: biz.capitalware.mqausx.MQAUSXJ

○ in directory

● in jar C:\Capitalware\MQAUSX\MQAUSXJ.jar

Exit data:

- Click the **Finish** button

3.3.1.2 Batch or Quiet mode for MQ Explorer MQ v7.0, v7.1, v7.5, v8, v9 & v9.1

Each time the user connects to the queue manager, he / she will be prompted for his / her UserId and Password (and Server Name). To run in batch or quiet mode, the user can explicitly set the UserId and Password in the channel's Exit Data or specify a file in the Exit Data that will contain the UserId and Password.

To explicitly set the UserId and Password values for the user-defined client-side security exit for authentication, do the following:

1. Open MQ Explorer v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
2. In the left panel, select **Queue Managers** under IBM IBM MQ
3. Right-click and select **Add Remote Queue Manager**
4. Input the new queue manager name and click **Next**
5. Input the hostname, port number and channel name and click **Next**
6. Click (enable) the checkbox **Enable security exit**
7. In the **Exit Name** field, input the following:

```
biz.capitalware.mqausx.MQAUSXJ
```

8. Select the **in jar** radio button and input the following:

```
C:\Capitalware\MQAUSX\MQAUSXJ.jar
```

9. In the **Exit Data** field input:

```
u=myUserId;p=myPassword
```

10. Click the **Finish** button

To specify a file that will contain the UserId and Password values for the user-defined client-side security exit for authentication, follow the steps 1-8 above and then do the following:

9. In the **Exit Data** field, input the following for a file:

```
C:\Capitalware\MQAUSX\clnt.ini
```

Input the following for an encrypted file (see Appendix B for more information)

```
C:\Capitalware\MQAUSX\clnt.enc
```

10. Click the **Finish** button

Note: Security User Data must NOT exceed 32 characters.

3.3.2 Indirectly using CCDT

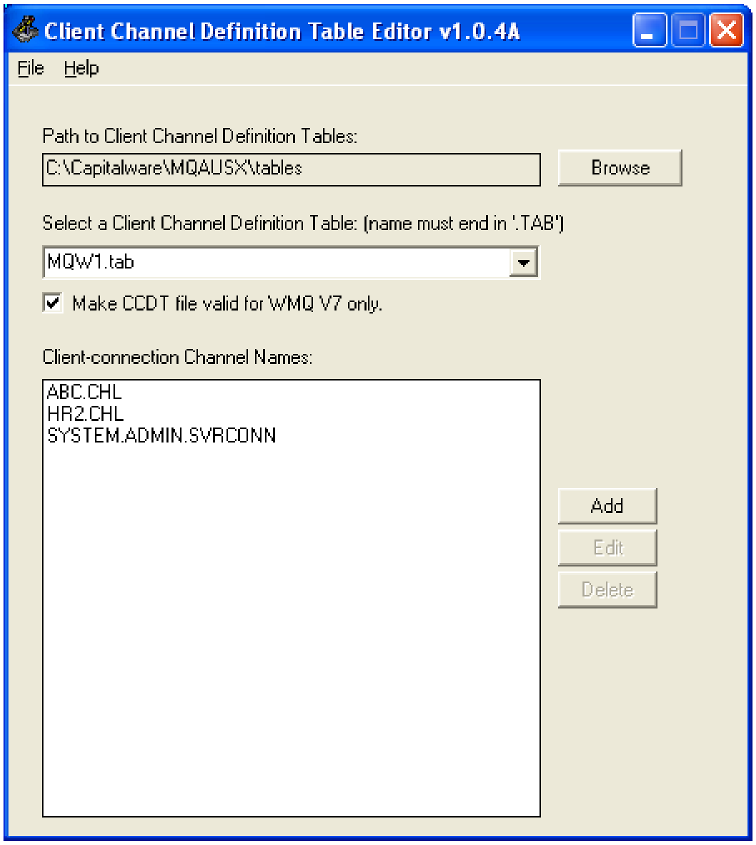
3.3.2.1 Creating a CCDT Entry

MQ Explorer is a Java application and MQ Client library supports both a Channel Security Exit as a Java JAR file and a native Windows DLL in a CCDT entry. The CCDT file will be found (assuming a default install) at this location:

```
C:\Capitalware\MQAUSX\tables\MQW1.TAB
```

3.3.2.1.1.1 Creating a CCDT Entry for a Pure Java Implementation

To enable user-defined client-side security exit for authentication, do the following steps:



Path to Client Channel Definition Tables: C:\Capitalware\MQAUSX\tables

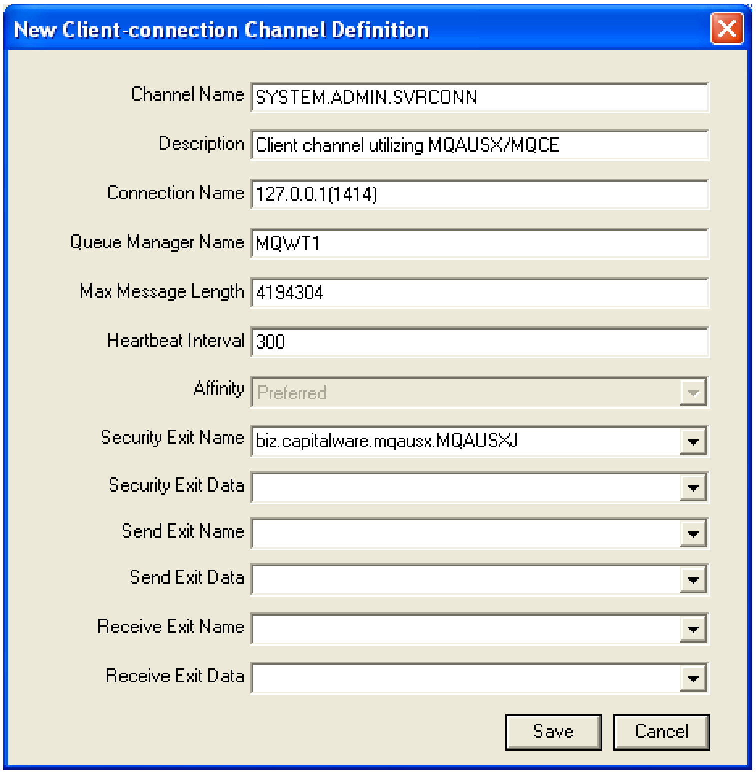
Select a Client Channel Definition Table: (name must end in '.TAB') MQW1.tab

☑ Make CCDT file valid for WMQ V7 only.

Client-connection Channel Names:

ABC.CHL
HR2.CHL
SYSTEM.ADMIN.SVRCONN

1 Start the Client Channel Definition Table Editor (From the ***Start*** -> ***All Programs*** menu)
2 Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')
3 Click the Add button to insert a new CLNTCONN channel or click the Edit button to edit an existing CLNTCONN channel.



Channel Name: SYSTEM.ADMIN.SVRCONN

Description: Client channel utilizing MQAUSX/MQCE

Connection Name: 127.0.0.1(1414)

Queue Manager Name: MQWT1

Max Message Length: 4194304

Heartbeat Interval: 300

Affinity: Preferred

Security Exit Name: biz.capitalware.mqausx.MQAUSXJ

Security Exit Data:

Send Exit Name:

Send Exit Data:

Receive Exit Name:

Receive Exit Data:

4 For ***Security Exit Name***, select ***biz.capitalware.mqausx.MQAUSXJ*** from the drop-down list.

A client channel definition table will be created in the 'tables' directory under the default install directory.

Important: For the pure Java implementation, the CLASSPATH for MQ Explorer needs to be updated. There are 2 ways this can be accomplished:

1) Update the runmqcfg_rcp.cmd file in the MQ_Install_Directory\bin\ directory (i.e. C:\Program Files\IBM\IBM MQ\bin) with the exitClasspath JVM environment variable (add it as the last “set AMQ_EXPLORER” command).

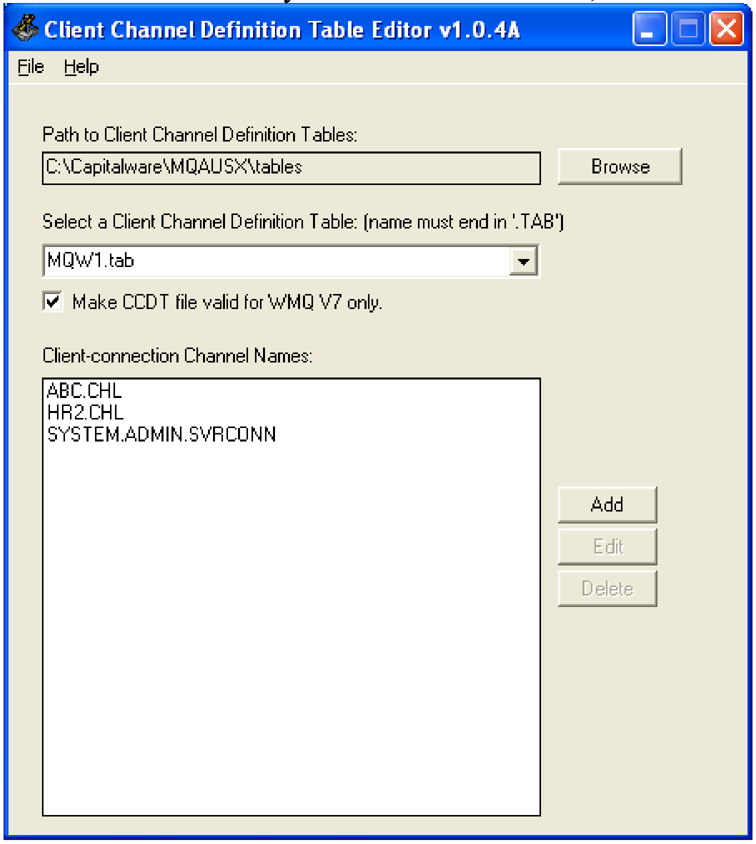
```
set AMQ_EXPLORER=%AMQ_EXPLORER% "-Dcom.ibm.mq.exitClasspath=C:/Capitalware/MQAUSX/MQAUSXJ.jar"
```

2) Edit the mqclient.ini file in the MQ_Install_Directory\bin\ directory (i.e. C:\Program Files\IBM\IBM MQ\bin) and add JavaExitsClassPath keyword after the section name ClientExitPath (make sure to use the "Tab" character to indent JavaExitsClassPath and not spaces):

```
ClientExitPath:
   JavaExitsClassPath=C:\Capitalware\MQAUSX\MQAUSXJ.jar
```

3.3.2.1.1.2 Creating a CCDT Entry using a native Windows DLL

To enable user-defined client-side security exit for authentication, do the following steps:



Path to Client Channel Definition Tables: C:\Capitalware\MQAUSX\tables

Select a Client Channel Definition Table: (name must end in '.TAB') MQW1.tab

☑ Make CCDT file valid for WMQ V7 only.

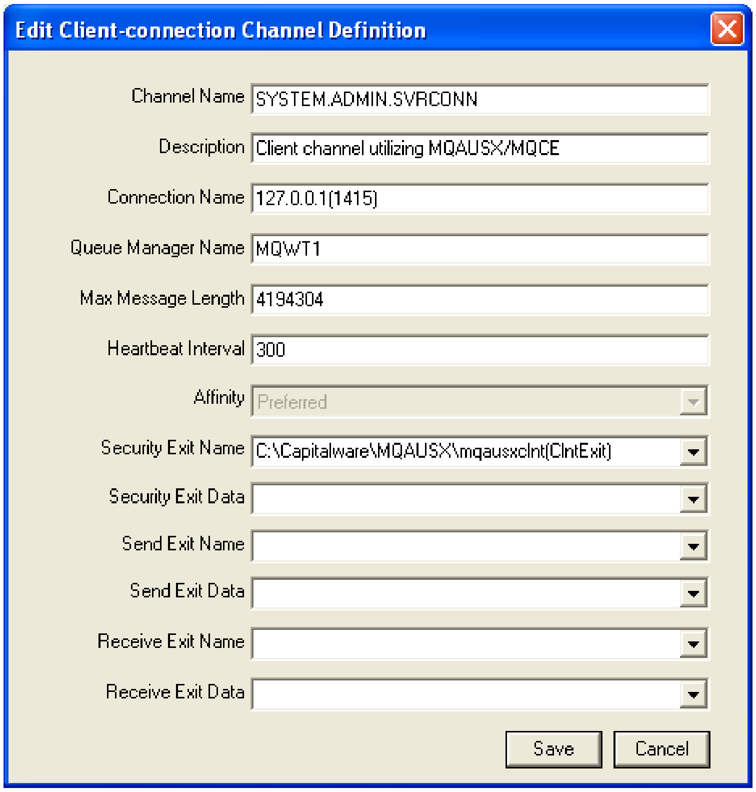
Client-connection Channel Names:

ABC.CHL
HR2.CHL
SYSTEM.ADMIN.SVRCONN

1 Start the Client Channel Definition Table Editor (From the ***Start*** -> ***All Programs*** menu)

2 Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')

3 Click the Add button to insert a new CLNTCONN channel or click the Edit button to edit an existing CLNTCONN channel.



Channel Name SYSTEM.ADMIN.SVRCONN

Description Client channel utilizing MQAUSX/MQCE

Connection Name 127.0.0.1(1415)

Queue Manager Name MQWT1

Max Message Length 4194304

Heartbeat Interval 300

Affinity Preferred

Security Exit Name C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)

Security Exit Data

Send Exit Name

Send Exit Data

Receive Exit Name

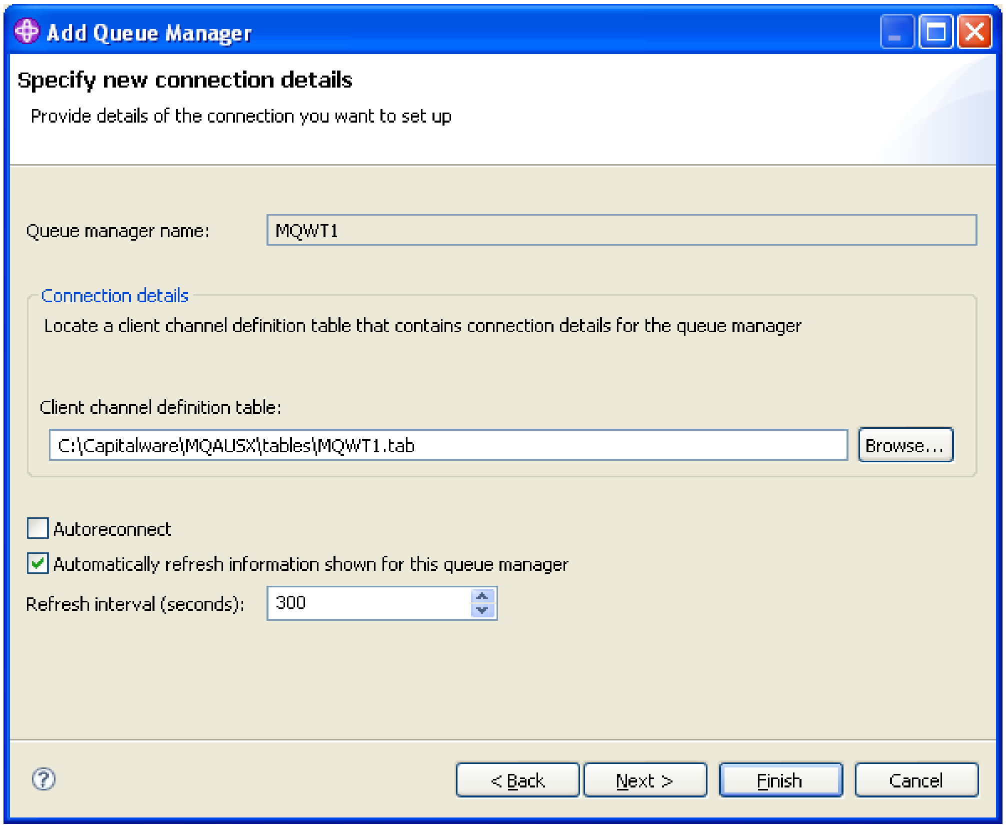
Receive Exit Data

4 For ***Security Exit Name***, select ***C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)*** from the drop-down list.

A client channel definition table will be created in the 'tables' directory under the default install directory.

3.3.2.2 Adding a Queue Manager using a client channel definition table

- Open MQ Explorer v7.0, v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
- In the left panel, right click on ***Queue Managers*** and select ***Add Remote Queue Manager***
- Fill in the ***Queue manager name***, select the radio button "Connect using a client channel definition table" and click *Next*
- Cick the ***Browse*** button to select the appropriate client channel definition table from the C:\Capitalware\MQAUSX\tables\ directory.
- Click the ***Finish*** button


Add Queue Manager
Specify new connection details
Provide details of the connection you want to set up
Connection details
Locate a client channel definition table that contains connection details for the queue manager
Browse...
< Back
Next >
Finish
Cancel

Queue manager name: MQWT1

Client channel definition table: C:\Capitalware\MQAUSX\tables\MQWT1.tab

□ Autoreconnect

☑ Automatically refresh information shown for this queue manager

Refresh interval (seconds): 300

3.3.3 IBM APAR IC58936 and IZ69820

Issues were discovered with IBM's MQ Explorer v7.0.0.0 or v7.0.0.1 or v7.0.1.1. These issues affect the use of any client-side security exits including MQAUSX. IBM has fixed both issues. The fix for IC58936 is included in MQ v7.0.1.0 and the fix for IZ69820 will be included in the MQ v7.0.1.2 and higher releases.

http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg1IC58936

Warning: Please exit all Java applications and MQ Explorer before applying the fix.

If you are using MQ Explorer v7.0.0.0 or v7.0.0.1 or v7.0.1.1, you will need to apply APAR IC58936 or IZ69820. A copy of the fixed JAR file has been included in the directory, **APAR\MQ_v7\IC52936** and **APAR\MQ_v7\IZ69820**, which can be found on the MQAUSX CD and in the MQAUSX download file.

To apply the fix, execute fix_MQ_v7.bat script, go to C:\Capitalware\MQAUSX and then run:

```
C:\Capitalware\MQAUSX\APAR\MQ_v7\fix_MQ_v7.bat
```

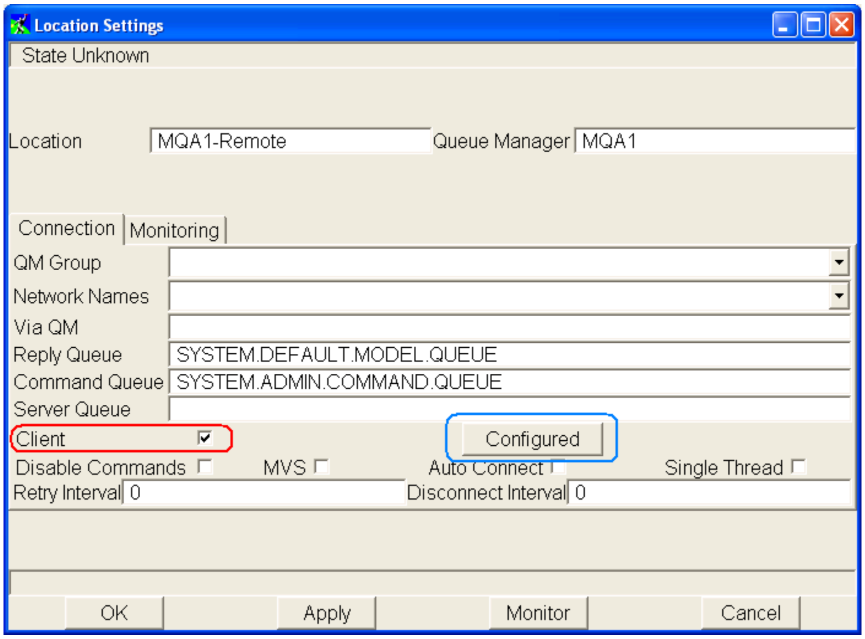
3.4 Configuring Security Exit in SupportPac MO71

This section describes the necessary steps to enable Security Exits in SupportPac MO71.

3.4.1 GUI popup window for SupportPac MO71

To enable user-defined client-side security exit for authentication:

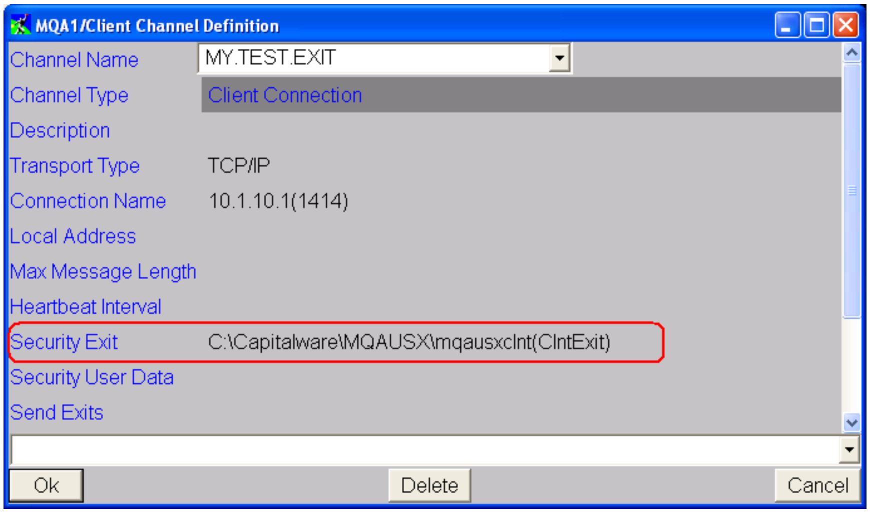
1. Select the queue manager in the tree display.
2. Right-click and select Open Location
3. Click the **Configured** button to the right of the Client label
4. Click on the field to the right of label **Security Exit** and input:
   `C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)`
5. Click Ok on the Channel Definition window
6. Click Ok on the Location Setting window


Location Settings

State Unknown

Location: MQA1-Remote
Queue Manager: MQA1

QM Group:
Network Names:
Via QM:
Reply Queue: SYSTEM.DEFAULT.MODEL.QUEUE
Command Queue: SYSTEM.ADMIN.COMMAND.QUEUE
Server Queue:
Client ☑
Disable Commands □ MVS □ Auto Connect □ Single Thread □
Retry Interval: 0
Disconnect Interval: 0


MQA1/Client Channel Definition

Channel Name: MY.TEST.EXIT
Channel Type: Client Connection
Description:
Transport Type: TCP/IP
Connection Name: 10.1.10.1(1414)
Local Address:
Max Message Length:
Heartbeat Interval:
Security Exit: C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
Security User Data:
Send Exits:

3.4.2 Batch or Quiet mode for SupportPac MO71

Each time the user connects to the queue manager, they will be prompted for their UserId and Password (and Server Name). To run in batch or quiet mode, the user can explicitly set the UserId and Password in the channel's SecurityUserData or specify a file in the SecurityUserData that will contain the UserId and Password.

To explicitly set the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Right-click and select Open Location
3. Click the **Configured** button to the right of the Client label
4. Click on the field to the right of label **Security Exit** and input:
   ```
   C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
   ```
5. Click on the field to the right of label **Security User Data** and input:
   ```
   u=fred;p=abcdef;s=ABC123
   ```
6. Click Ok on the Channel Definition window
7. Click Ok on the Location Setting window

To specify a file that will contain the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Right-click and select Open Location
3. Click the **Configured** button to the right of the Client label
4. Click on the field to the right of label **Security Exit** and input:
   ```
   C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
   ```
5. Click on the field to the right of label **Security User Data** and input the following (read Appendix A for the format of the file):
   ```
   C:\Capitalware\MQAUSX\clnt.ini
   ```
   Or use an encrypted file. (see Appendix B for more information)
   ```
   C:\Capitalware\MQAUSX\clnt.enc
   ```
6. Click Ok on the Channel Definition window
7. Click Ok on the Location Setting window

Note: Security User Data must NOT exceed 32 characters.

3.5 Configuring Security Exit in IBM's WBIMB Eclipse Tool Kit

This section describes the necessary steps to enable Security Exits in the WBIMB Eclipse Tool Kit. The steps are as follows:

- Click File -> New -> Domain. The Domain view appears
- Enter the queue manager name, host, and port that you wish to use
- Enter the security exit name:

```
biz.capitalware.mqausx.MQAUSXJ
```

- Enter the location of the JAR:

```
C:\Capitalware\MQAUSX\MQAUSXJ.jar
```

3.5.1 WBIMB Server-side Channel Configuration

The WBIMB Eclipse Tool Kit communicates with a queue manager using the 'SYSTEM.BKR.CONFIG' channel. Therefore, the server-side security exit MUST be configured for the 'SYSTEM.BKR.CONFIG' channel.

3.6 WebSphere Message Broker Explorer V8.0 or higher

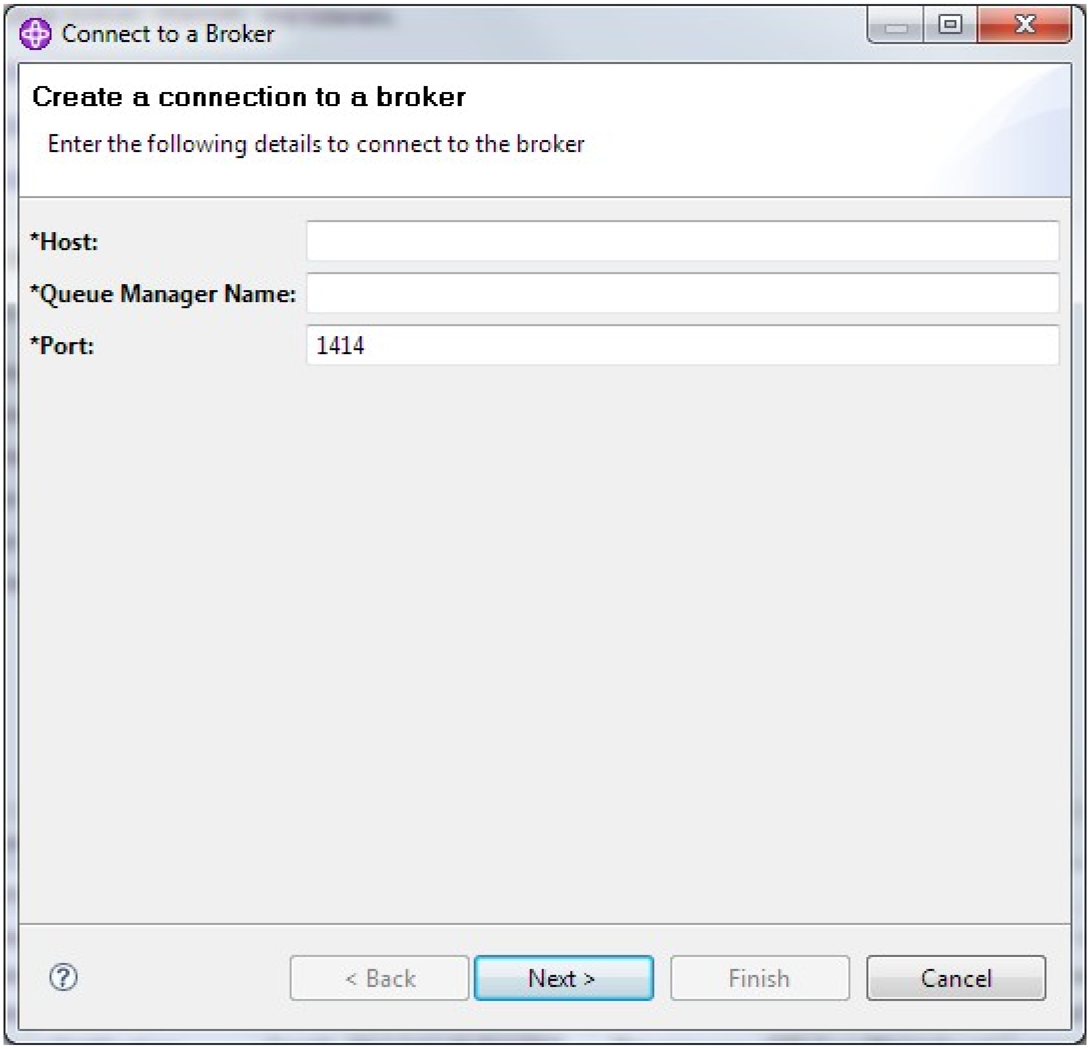
This section describes the necessary steps to enable the Client-side Security Exit in WebSphere Message Broker Explorer V8.0.

3.6.1 Directly using Class Name and Classpath

3.6.1.1 GUI popup window for Message Broker Explorer

To enable user-defined client-side security exit for authentication, do the following steps:

- Open Message Broker Explorer
- Right-click the Broker icon then select '**Connect to a remote broker...**'



Create a connection to a broker

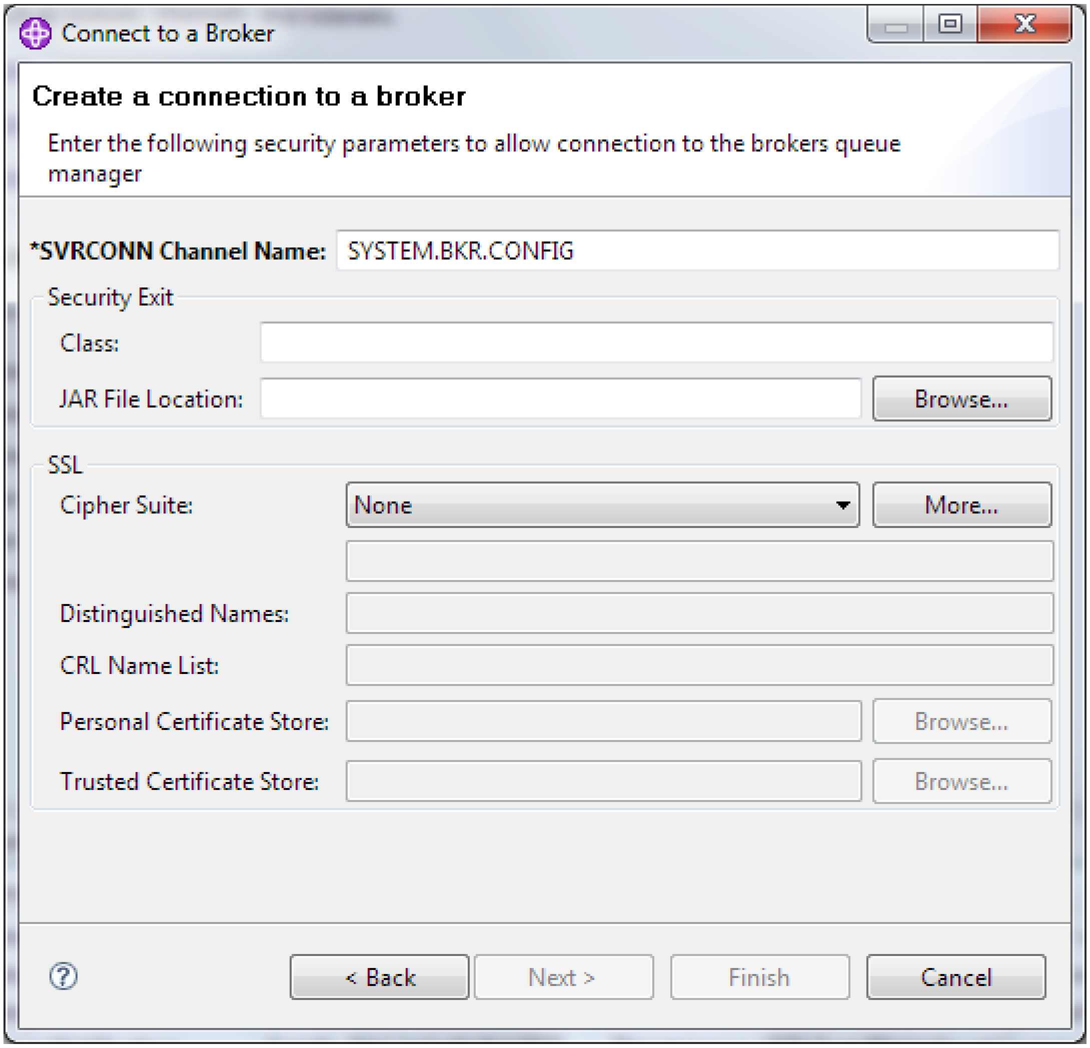
Enter the following details to connect to the broker

*Host:

*Queue Manager Name:

*Port: 1414

- Input the new queue manager name, hostname and port number then click **Next**



Connect to a Broker

Create a connection to a broker

Enter the following security parameters to allow connection to the brokers queue manager

*SVRCONN Channel Name: SYSTEM.BKR.CONFIG

Security Exit

Class:

JAR File Location:

SSL

Cipher Suite: None

Distinguished Names:

CRL Name List:

Personal Certificate Store:

Trusted Certificate Store:

- In the **Class** field, input the following:

```
biz.capitalware.mqausx.MQAUSXJ
```

- Select the **Browse** button and input the following:

```
C:\Capitalware\MQAUSX\MQAUSXJ.jar
```

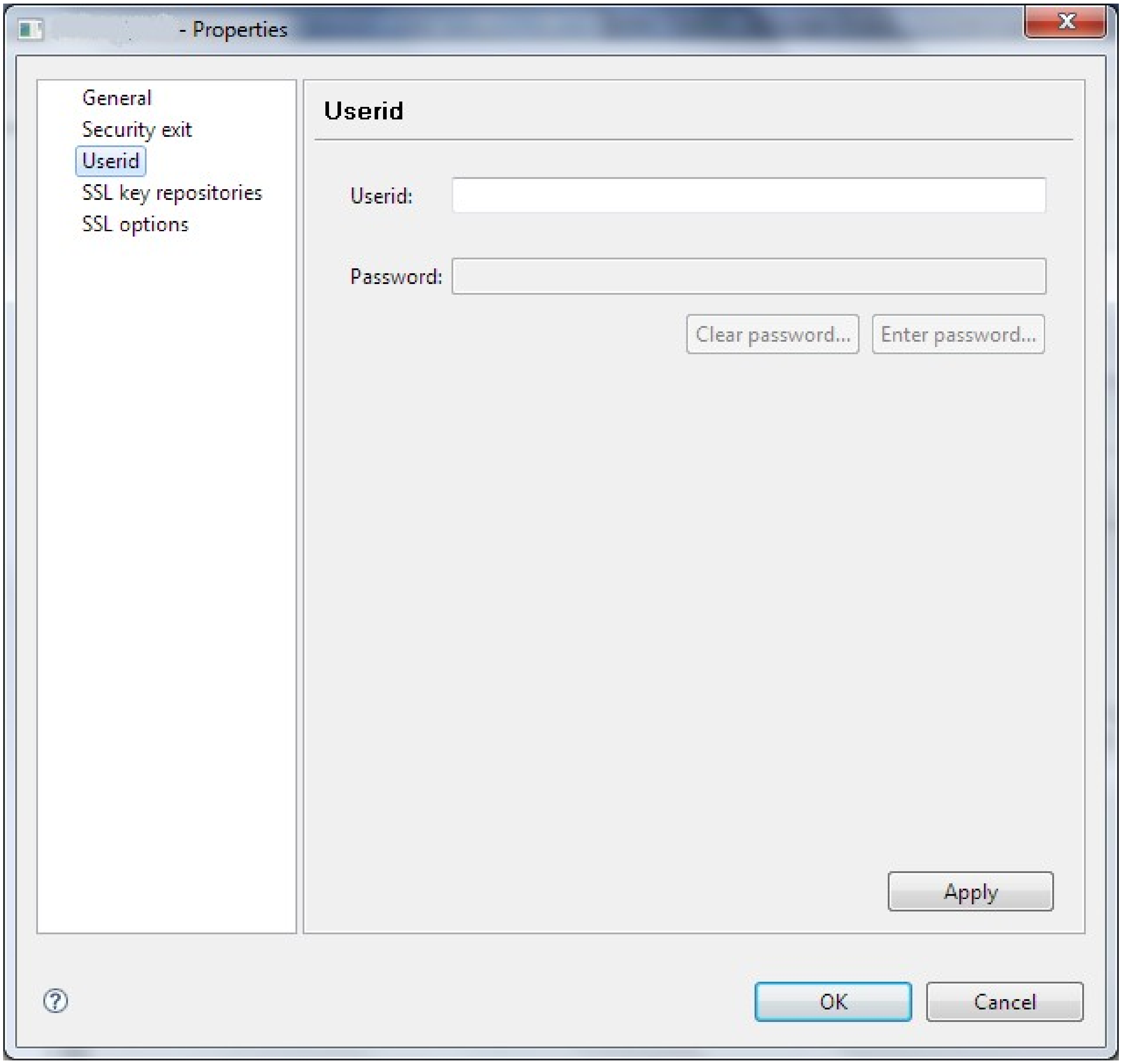
- Click the **Finish** button

3.6.1.2 Batch or Quiet mode for WebSphere Message Broker Explorer V8.0

Each time the user connects to the queue manager, he / she will be prompted for his / her UserId and Password (and Server Name). To run in batch or quiet mode, the user can explicitly set the UserId and Password in WebSphere Message Broker Explorer V8.0.

To explicitly set the UserId and Password values for the user-defined client-side security exit for authentication, do the following:

1. Open Message Broker Explorer
2. In the left panel, select the previously defined broker queue manager
3. Right-click on it and select **Connection Details -> Properties**
4. Input the new queue manager name and click **Next**
5. Input your UserID and Password then click **Apply** then **OK**


- Properties

Userid

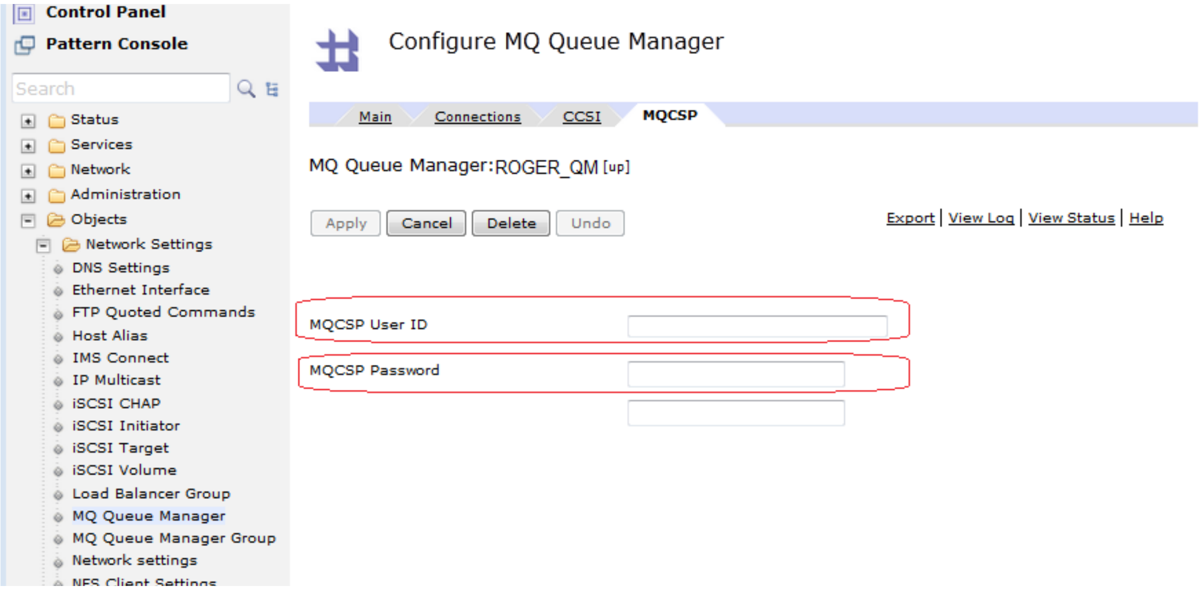
Userid:

Password:

3.7 Configuring IBM DataPower

This section describes the necessary steps to allow IBM's DataPower to send the UserID and Password to MQAUSX server-side component. Note: DataPower is a closed architecture, so the MQAUSX client-side security exit cannot be installed on the DataPower appliance. The steps are as follows:

- Open the DataPower Administration Panel
- Go to Objects → Network Settings → MQ Queue Manager
- Fill in the information on the **Main** tab
- Click the **Connections** tab and fill in the channel, IP address and port number
- Click the **MQCSP** tab and input the application's UserID and Password in the **MQCSP User ID** and **MQCSP Password** fields
- Click the Apply button to save the information



MQ Queue Manager:ROGER_QM [up]

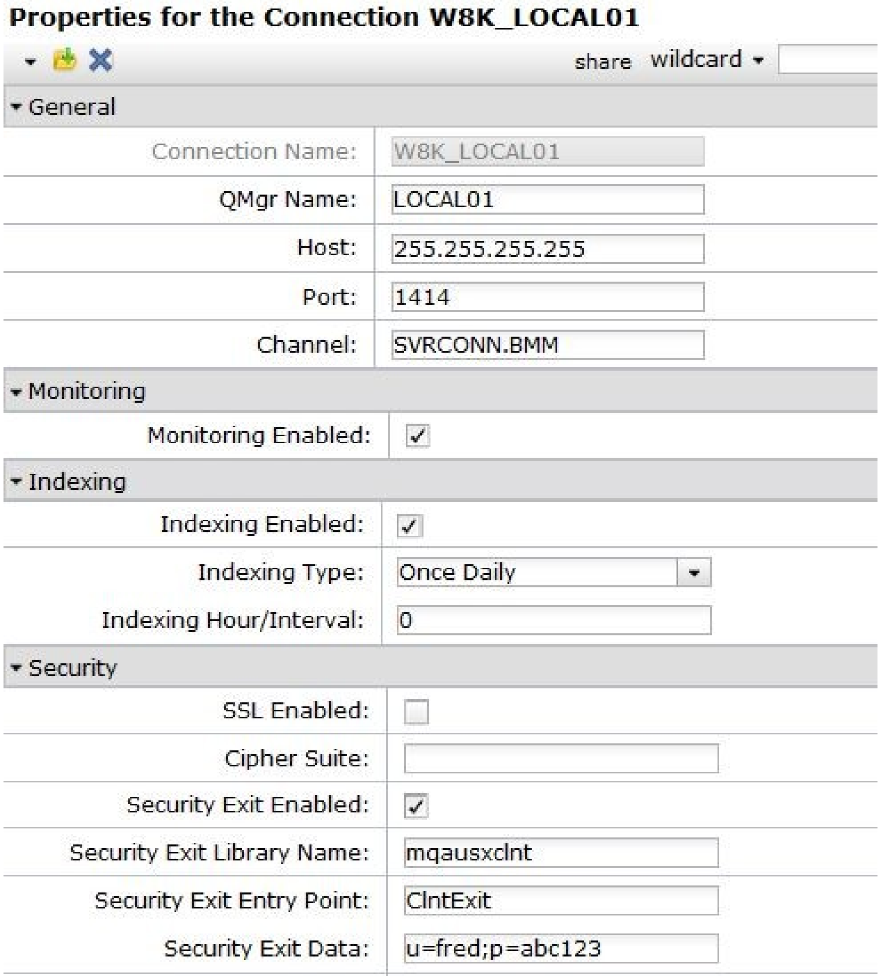
3.8 Configuring Security Exit in BMC Middleware Management - Administration (BMM Admin)

This section describes the necessary steps to enable the MQAUSX client-side security exit in the BMC Middleware Management - Administration (BMM Admin). The steps are as follows:

1. Even though BMM Admin is a Java app, it uses a non-Java security exit.

   - BMM Admin is written in the Java programming language, but is not an MQ Java Client. Thus, the Java exit point is not available. Instead, BMM Admin uses a Native MQ Client interface.
   - BMM Admin invokes the Native MQ Client exit point, requiring a dll (Windows) or shared library (Unix/Linux).
   - For BMM Admin, it is the connection binding rather than the programming language that matters for exits.

2. BMC Middleware Management - Administration 7.2.XX is a 64-bit application.
3. The MQ environment must be set for BMM Admin to be able to find the exit (setmqenv).



Properties for the Connection W8K_LOCAL01

General

Connection Name:	W8K_LOCAL01
QMgr Name:	LOCAL01
Host:	255.255.255.255
Port:	1414
Channel:	SVRCONN.BMM

Monitoring

Monitoring Enabled:	☑

Indexing

Indexing Enabled:	☑
Indexing Type:	Once Daily
Indexing Hour/Interval:	0

Security

SSL Enabled:	□
Cipher Suite:	
Security Exit Enabled:	☑
Security Exit Library Name:	mqausxclnt
Security Exit Entry Point:	ClntExit
Security Exit Data:	u=fred;p=abc123

To explicitly set the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the properties for the Queue Manager Connection.
2. Click the checkbox to the right of the label Security Exit Enabled to enable it.
3. Click on the field to the right of the label **Security Exit Client Name** and input: `mqausxclnt`
4. Click on the field to the right of the label **Security Exit Entry Point** and input: `ClntExit`
5. Click on the field to the right of label **Security Exit Data** and input: `u=fred;p=abcdef`
6. Save your changes.

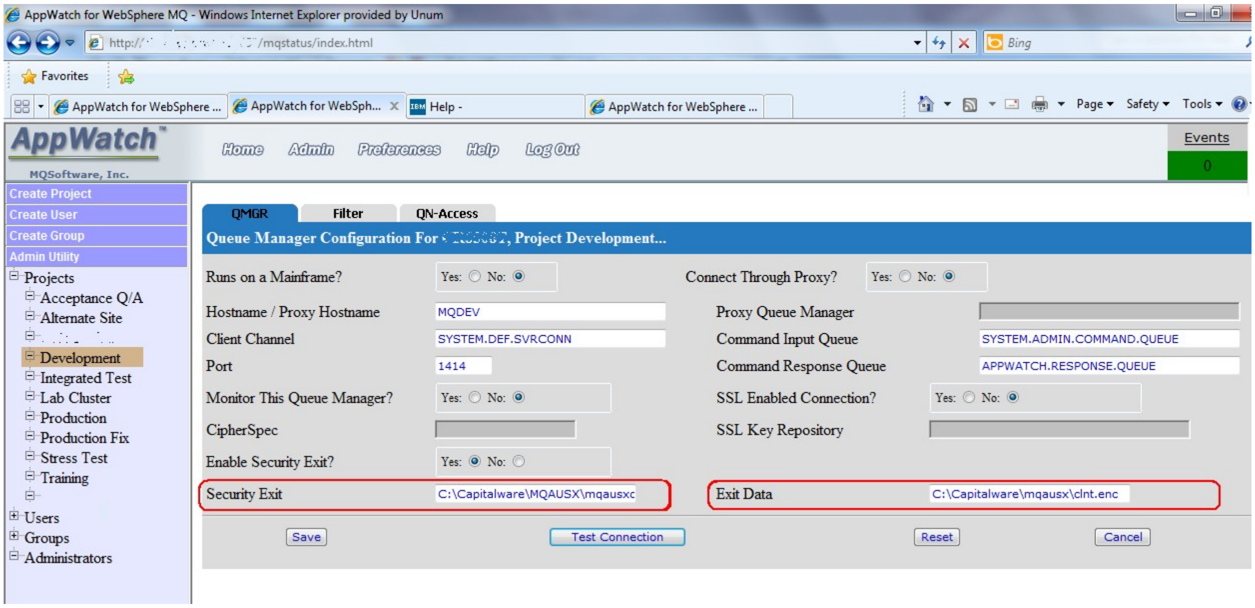
To specify a file that will contain the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the properties for the Queue Manager Connection.
2. Click the checkbox to the right of the label Security Exit Enabled to enable it.
3. Click on the field to the right of the label **Security Exit Client Name** and input: `mqausxclnt`
4. Click on the field to the right of the label **Security Exit Entry Point** and input: `ClntExit`
5. Click on the field to the right of label **Security Exit Data** and input the following (read Appendix A for the format of the file): `C:\Capitalware\MQAUSX\clnt.ini`
   Or use an encrypted file. (see Appendix B for more information) `C:\Capitalware\MQAUSX\clnt.enc`
6. Save your changes.

Note: Security User Data must NOT exceed 32 characters.

3.9 Configuring Security Exit in BMC's Administration for IBM MQ (AppWatch)

This section describes the necessary steps to enable Security Exits in the BMC's Administration for IBM MQ (AppWatch). The steps are as follows:



Queue Manager Configuration For [redacted], Project Development...

Runs on a Mainframe? Yes: ○ No: ●

Connect Through Proxy? Yes: ○ No: ●

Hostname / Proxy Hostname: MQDEV

Proxy Queue Manager:

Client Channel: SYSTEM.DEF.SVRCONN

Command Input Queue: SYSTEM.ADMIN.COMMAND.QUEUE

Port: 1414

Command Response Queue: APPWATCH.RESPONSE.QUEUE

Monitor This Queue Manager? Yes: ○ No: ●

SSL Enabled Connection? Yes: ○ No: ●

CipherSpec:

SSL Key Repository:

Enable Security Exit? Yes: ● No: ○

Security Exit: C:\Capitalware\MQAUSX\mqausxc

Exit Data: C:\Capitalware\mqausx\clnt.enc

To explicitly set the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Click on the field to the right of label **Security Exit** and input:
   `C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)`
3. Click on the field to the right of label **Exit Data** and input:
   `u=fred;p=abcdef`
4. Click **Save** on the 'Queue Manager Configuration' window

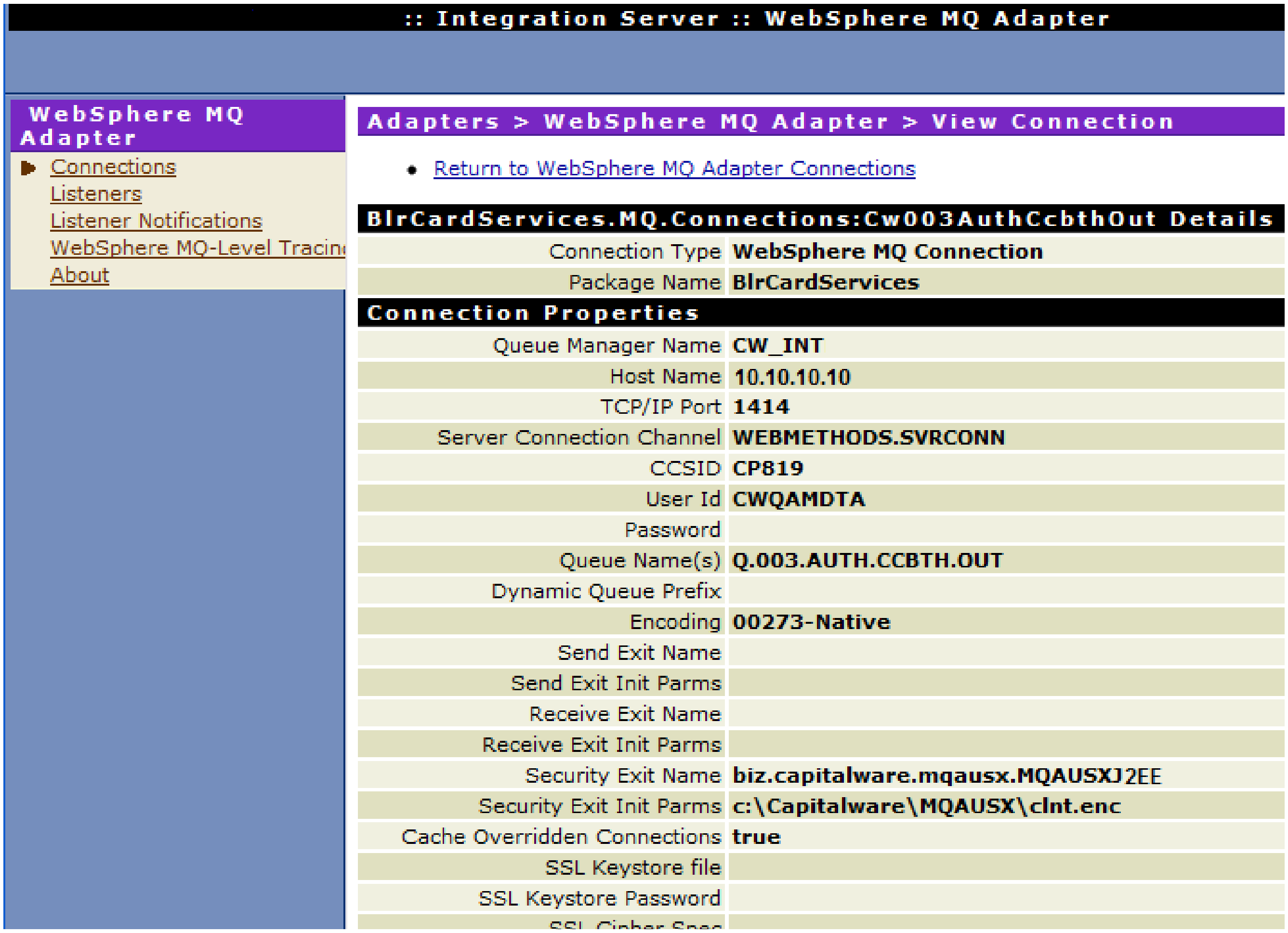
To specify a file that will contain the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Click on the field to the right of label **Security Exit** and input:
   `C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)`
3. Click on the field to the right of label **Exit Data** and input the following (read Appendix A for the format of the file):
   `C:\Capitalware\MQAUSX\clnt.ini`
   Or use an encrypted file. (see Appendix B for more information)
   `C:\Capitalware\MQAUSX\clnt.enc`
4. Click Save on the 'Queue Manager Configuration' window

Note: Security User Data must NOT exceed 32 characters.

3.10 Configuring Security Exit in webMethods MQ Adapter

This section describes the necessary steps to enable Security Exits in webMethods MQ Adapter for IBM MQ. The steps are as follows:


:: Integration Server :: WebSphere MQ Adapter
WebSphere MQ Adapter
Connections
Listeners
Listener Notifications
WebSphere MQ-Level Tracin
About
Adapters > WebSphere MQ Adapter > View Connection
Return to WebSphere MQ Adapter Connections

BlrCardServices.MQ.Connections:Cw003AuthCcbthOut Details

Connection Type	WebSphere MQ Connection
Package Name	BlrCardServices

Connection Properties

Queue Manager Name	CW_INT
Host Name	10.10.10.10
TCP/IP Port	1414
Server Connection Channel	WEBMETHODS.SVRCONN
CCSID	CP819
User Id	CWQAMDTA
Password	
Queue Name(s)	Q.003.AUTH.CCBTH.OUT
Dynamic Queue Prefix	
Encoding	00273-Native
Send Exit Name	
Send Exit Init Parms	
Receive Exit Name	
Receive Exit Init Parms	
Security Exit Name	biz.capitalware.mqausx.MQAUSXJ2EE
Security Exit Init Parms	c:\Capitalware\MQAUSX\clnt.enc
Cache Overridden Connections	true
SSL Keystore file	
SSL Keystore Password	
SSL Cipher Spec	

To explicitly set the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Click on the field to the right of label **Security Exit Name** and input: `biz.capitalware.mqausx.MQAUSXJ2EE`
3. Click on the field to the right of label **Security Exit Init Parms** and input: `u=fred;p=abcdef`
4. Click **Save** on the 'Queue Manager Configuration' window
5. Set the CLASSPATH to the location of the MQAUSXJ.jar file

To specify a file that will contain the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

1. Select the queue manager in the tree display.
2. Click on the field to the right of label **Security Exit Name** and input:

    `biz.capitalware.mqausx.MQAUSXJ2EE`

3. Click on the field to the right of label **Security Exit Init Parms** and input the following (read Appendix A for the format of the file):
   `C:\Capitalware\MQAUSX\clnt.ini`
   Or use an encrypted file. (see Appendix B for more information)
   `C:\Capitalware\MQAUSX\clnt.enc`
4. Click Save on the 'Queue Manager Configuration' window
5. Set the CLASSPATH to the location of the MQAUSXJ.jar file

Note: Security User Data must NOT exceed 32 characters.

3.11 Configuring Security Exit in Mercury's SiteScope

This section describes the necessary steps to enable Security Exits in Mercury's SiteScope.

To enable user-defined client-side security exit for authentication, the following steps should be followed:

- Using a text editor, open the SiteScope configuration file:

  ```
  SiteScope\groups\master.config
  ```

- Add a single line entry to the master.config file using the following syntax:

  ```
  _mqMonitorSecurityExit= biz.capitalware.mqausx.MQAUSXJ
  ```

- Copy the custom security class file to a location in the classpath of the Java Virtual Machine (JVM) running on the SiteScope server.

For example, copy the security exit class into the
***<SiteScope install path>*\SiteScope\java\lib\ext**
directory. Also, the user can add the client-side security exit to their CLASSPATH as per this example:

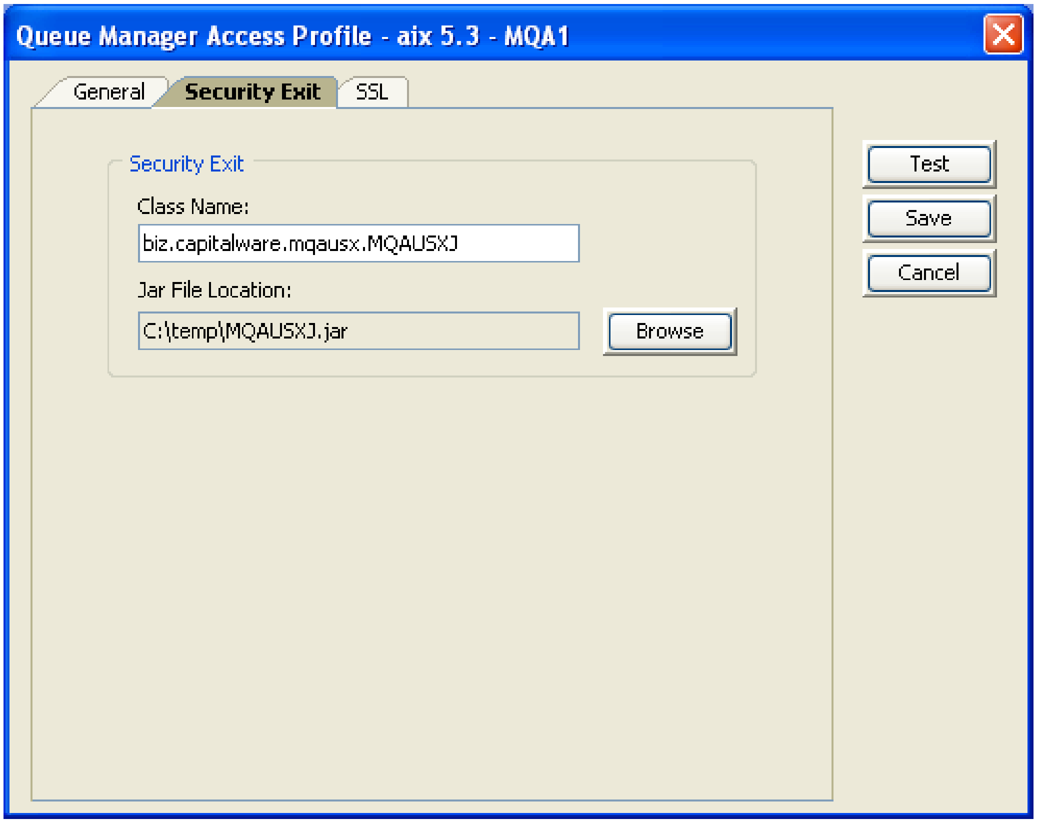
```
SET CLASSPATH=C:\Capitalware\MQAUSX\MQAUSXJ.jar;%CLASSPATH%
```

3.12 Configuring Security Exit in Capitalware's MQ Visual Edit

This section describes the necessary steps to enable Security Exits in the MQ Visual Edit.

The steps are as follows:

- Click File -> Open Queue
- Select the Queue Manager Access Profile and then click the Edit button
- Enter the queue manager name, host, and port that you wish to use
- Enter the Security Exit Class Name: `biz.capitalware.mqausx.MQAUSXJ`
- Enter the Security Exit Jar File Location: `C:\Capitalware\MQAUSX\MQAUSXJ.jar`



Security Exit

Class Name: biz.capitalware.mqausx.MQAUSXJ

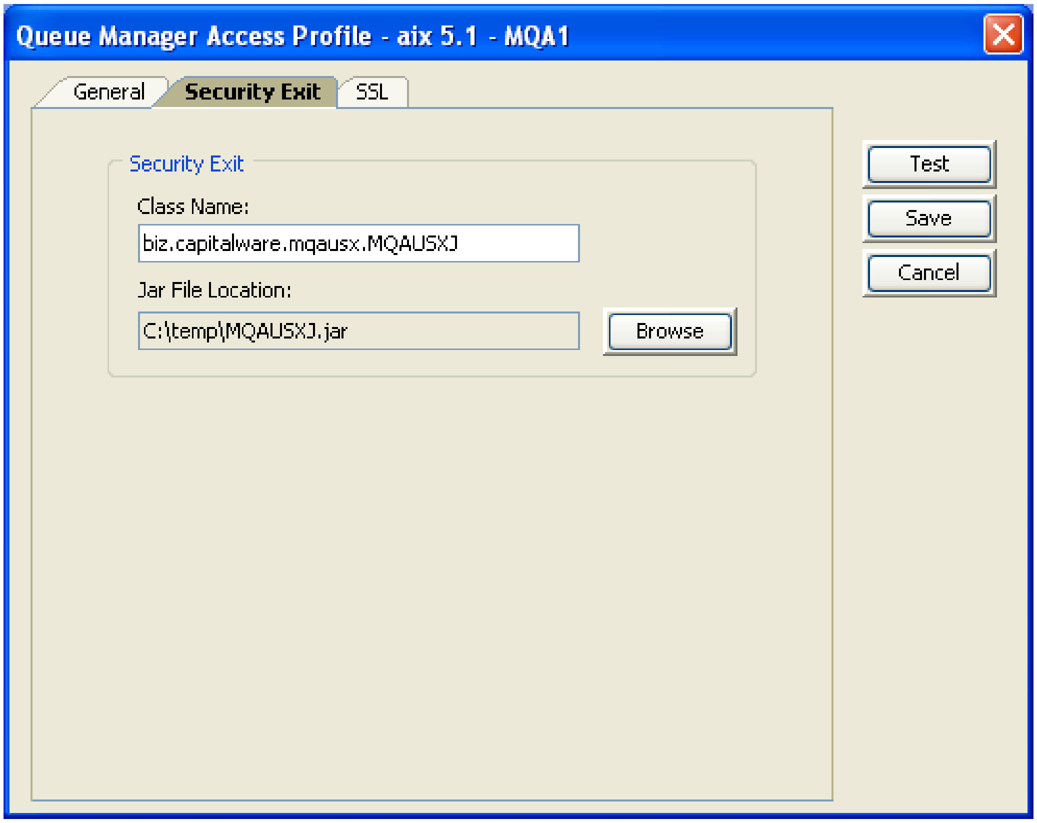
Jar File Location: C:\temp\MQAUSXJ.jar

3.13 Configuring Security Exit in Capitalware's MQ Visual Browse

This section describes the necessary steps to enable Security Exits in the MQ Visual Browse.

The steps are as follows:

- Click File -> Open Queue
- Select the Queue Manager Access Profile and then click the Edit button
- Enter the queue manager name, host, and port that you wish to use
- Enter the Security Exit Class Name:
  `biz.capitalware.mqausx.MQAUSXJ`
- Enter the Security Exit Jar File Location:
  `C:\Capitalware\MQAUSX\MQAUSXJ.jar`


Queue Manager Access Profile - aix 5.1 - MQA1

Security Exit

Class Name: biz.capitalware.mqausx.MQAUSXJ

Jar File Location: C:\temp\MQAUSXJ.jar

3.14 Configuring Security Exit in Capitalware's MQ Batch Toolkit

This section describes the necessary steps to enable Security Exits in the MQ Batch Toolkit.

3.14.1 AddProfile Command

For more information on the AddProfile command, see chapter 3.1 on the ***MQ Batch Toolkit Installation and Operation*** manual.

3.14.1.1 Windows

On Windows issue the following command:

```
mqbt AddProfile -p MQA1 -m MQA1 -h 10.10.10.10 -n 1414 -c TEST.CHL -x
     biz.capitalware.mqausx.MQAUSXJ2EE -f C:\Capitalware\MQAUSX\
     MQAUSXJ.jar -u myuserID -w mypwd
```

where

- MQA1 is the queue manager name
- 10.10.10.10 is the IP address of the server
- 1414 is the listener's port number
- TEST.CHL is the channel name
- biz.capitalware.mqausx.MQAUSXJ2EE is the security exit class name
- C:\Capitalware\MQAUSX\MQAUSXJ.jar is the full path and file name of the JAR file
- myuserID is your UserID
- mypwd is your password

3.14.1.2 Unix/Linux

On Unix or Linux issue the following command:

```
mqbt AddProfile -p MQA1 -m MQA1 -h 10.10.10.10 -n 1414 -c TEST.CHL -x
     biz.capitalware.mqausx.MQAUSXJ2EE -f /var/mqm/exits64/MQAUSXJ.jar  -
     u myuserID -w mypwd
```

where

- MQA1 is the queue manager name
- 10.10.10.10 is the IP address of the server
- 1414 is the listener's port number
- TEST.CHL is the channel name
- biz.capitalware.mqausx.MQAUSXJ2EE is the security exit class name
- /var/mqm/exits64/MQAUSXJ.jar is the full path and file name of the JAR file
- myuserID is your UserID
- mypwd is your password

3.14.2 AlterProfile Command

For more information on the AlterProfile command, see chapter 3.2 on the ***MQ Batch Toolkit Installation and Operation*** manual.

3.14.2.1 Windows

On Windows issue the following command:

```
mqbt AlterProfile -p MQA1 -x biz.capitalware.mqausx.MQAUSXJ2EE -f C:\
     Capitalware\MQAUSX\MQAUSXJ.jar -u myuserID -w mypwd
```

where

- MQA1 is the profile name
- biz.capitalware.mqausx.MQAUSXJ2EE is the security exit class name
- C:\Capitalware\MQAUSX\MQAUSXJ.jar is the full path and file name of the JAR file
- myuserID is your UserID
- mypwd is your password

3.14.2.2 Unix/Linux

On Unix or Linux, issue the following command:

```
mqbt AddProfile -p MQA1 -x biz.capitalware.mqausx.MQAUSXJ2EE -f
     /var/mqm/exits64/MQAUSXJ.jar -u myuserID -w mypwd
```

where

- MQA1 is the profile name
- biz.capitalware.mqausx.MQAUSXJ2EE is the security exit class name
- /var/mqm/exits64/MQAUSXJ.jar is the full path and file name of the JAR file
- myuserID is your UserID
- mypwd is your password

3.15 Configuring Security Exit in Capitalware's Universal File Mover

This section describes the necessary steps to enable Security Exits in the Universal File Mover. For more information on the editing the UFM_MQ XML file, see chapter 7 on the ***Universal File Mover Installation and Operation*** manual.

3.15.1.1 Windows

The following is an example of a UFM_MQ XML file for connecting to a remote queue manager using a security exit:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE UFM_MQ SYSTEM "UFM_MQ.dtd">
<UFM_MQ>
    <QMgrName>MQA1</QMgrName>
    <QueueName>TEST.Q1</QueueName>
    <Hostname>10.10.10.10</Hostname>
    <ChannelName>TEST.CHL</ChannelName>
    <Port>1414</Port>
    <SecurityExit>biz.capitalware.mqausx.MQAUSXJ2EE</SecurityExit>
    <SecurityExitPath>C:\Capitalware\MQAUSX\MQAUSXJ.jar</SecurityExitPath>
    <UserID>myuserID</UserID>
    <Password>mypwd</Password>
</UFM_MQ>
```

Note: The UFM_MQ XML files must be stored in the **<UFM_Install_PATH>\mq\** directory.
i.e. C:\Capitalware\UFM\mq\

3.15.1.2 Unix/Linux

The following is an example of a UFM_MQ XML file for connecting to a remote queue manager using a security exit:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE UFM_MQ SYSTEM "UFM_MQ.dtd">
<UFM_MQ>
    <QMgrName>MQA1</QMgrName>
    <QueueName>TEST.Q1</QueueName>
    <Hostname>10.10.10.10</Hostname>
    <ChannelName>TEST.CHL</ChannelName>
    <Port>1414</Port>
    <SecurityExit>biz.capitalware.mqausx.MQAUSXJ2EE</SecurityExit>
    <SecurityExitPath>/var/mqm/exits64/MQAUSXJ.jar</SecurityExitPath>
    <UserID>myuserID</UserID>
    <Password>mypwd</Password>
</UFM_MQ>
```

Note: The UFM_MQ XML files must be stored in the **<UFM_Install_PATH>/mq/** directory.
i.e. /opt/Capitalware/UFM/mq/

3.16 Configuring Security Exit for WebSphere Application Server

This section describes the necessary steps to enable Security Exits in IBM's WebSphere Application Server (WAS):

3.16.1 Updating WAS's JVM Classpath

Since the MQAUSXJ.jar file is intended for use by all applications deployed on WAS. Copy the MQAUSXJ.jar file to the ***ws.ext.dirs*** directory. As a result, the jar file will be loaded by the WAS extensions class loader.

On Windows, the ***ws.ext.dirs*** may be configured as

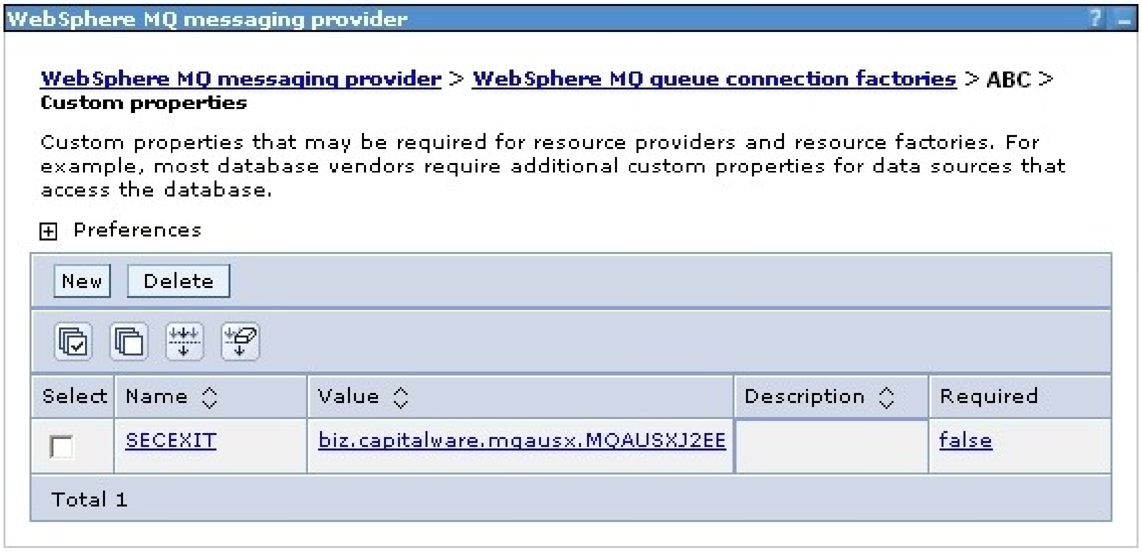
```
{WAS_Install_Path}\WebSphere6\AppServer\lib\ext
```

On Unix / Linux, the ***ws.ext.dirs*** may be configured as

```
{WAS_Install_Path}/WebSphere6/AppServer/lib/ext
```

3.16.2 Configuring WAS Admin Console

To add the MQAUSXJ2EE class to your deployed WAS application, add the **SECEXIT** custom property to your IBM MQ connection factory as show below:



WebSphere MQ messaging provider > WebSphere MQ queue connection factories > ABC >
Custom properties

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

Select	Name	Value	Description	Required
□	SECEXIT	biz.capitalware.mqausx.MQAUSXJ2EE		false
Total 1				

This setup assumes that the WAS application will be passing the UserID and Password using the ***createConnection*** method.

```
ConnectionFactory cf = (ConnectionFactory)ctx.lookup("MyQCF");
Connection conn = cf.createConnection("myUserId","myPswd");

Or
```

```
MQQueueConnectionFactory qcf = new MQQueueConnectionFactory();
QueueConnection qc =  qcf.createQueueConnection("myUserId","myPswd");
```

If the WAS application is not capable or cannot pass the UserID and Password using the ***createConnection*** method, the **SECEXITINIT** custom property needs to be added via the WAS Admin console.

The value for the **SECEXITINIT** custom property can be in 1 of 3 forms:

1. Set the UserID and Password explicitly as follows

   ```
   u=myUserId;p=myPswd
   ```

2. Set the custom property to a client-side IniFile

   ```
   C:\Capitalware\MQAUSX\clnt.ini
   ```

3. Set the custom property to a client-side encrypted IniFile

   ```
   C:\Capitalware\MQAUSX\clnt.enc
   ```

   Or use an encrypted file (see Appendix B for more information).

3.17 Configuring Security Exit for JBoss V7 or higher

This section describes the necessary steps to enable Security Exits in JBoss V7 or higher.

3.17.1 Updating standalone.xml (or standalone-full.xml)

3.17.1.1 MQAUSXJ Information

In the ***standalone.xml*** (or ***standalone-full.xml***), add the following property to the ***<system-properties>*** section (This will tell the resource adapter’s classloader where the exit classes are located.):

On Windows:

```
<property name="com.ibm.mq.cfg.ClientExitPath.JavaExitsClasspath" value="C:/Capitalware/MQAUSX/MQAUSXJ.jar"/>
```

On Unix / Linux:

```
<property name="com.ibm.mq.cfg.ClientExitPath.JavaExitsClasspath" value="/var/mqm/exits64/MQAUSXJ.jar"/>
```

3.17.1.2 MQ Connectivity Information

In the ***standalone.xml*** (or ***standalone-full.xml***), add the following property to the ***<connection-definition>*** section.

```
<config-property name="channel">SYSTEM.DEF.SVRCONN</config-property>
<config-property name="hostName">localhost</config-property>
<config-property name="transportType">CLIENT</config-property>
<config-property name="queueManager">ExampleQM</config-property>
<config-property name="port">1414</config-property>
```

3.17.2 Define the activation-config properties

Next, the user needs to define the activation-config properties. It can be done either using jboss-ejb3.xml file or annotation.

3.17.2.1 Using the jboss-ejb3.xml file

On Windows:

```
<jee:activation-config-property>
  <jee:activation-config-property-name>securityExit</jee:activation-config-property-name>
  <jee:activation-config-property-value>biz.capitalware.mqausx.MQAUSXJ2EE</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config-property>
  <jee:activation-config-property-name>securityExitInit</jee:activation-config-property-name>
  <jee:activation-config-property-value>C:/Capitalware/myclnt.enc</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config>
  <jee:activation-config-property>
    <jee:activation-config-property-name>channel</jee:activation-config-property-name>
    <jee:activation-config-property-value>CHANNEL.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destination</jee:activation-config-property-name>
    <jee:activation-config-property-value>QUEUE.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destinationType</jee:activation-config-property-name>
    <jee:activation-config-property-value>javax.jms.Queue</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>hostName</jee:activation-config-property-name>
    <jee:activation-config-property-value>HOST.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
</jee:activation-config>
```

On Unix / Linux:

```
<jee:activation-config-property>
  <jee:activation-config-property-name>securityExit</jee:activation-config-property-name>
  <jee:activation-config-property-value>biz.capitalware.mqausx.MQAUSXJ2EE</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config-property>
  <jee:activation-config-property-name>securityExitInit</jee:activation-config-property-name>
  <jee:activation-config-property-value>/apps/data/myclnt.enc</jee:activation-config-property-value>
</jee:activation-config-property>

<jee:activation-config>
  <jee:activation-config-property>
    <jee:activation-config-property-name>channel</jee:activation-config-property-name>
    <jee:activation-config-property-value>CHANNEL.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destination</jee:activation-config-property-name>
    <jee:activation-config-property-value>QUEUE.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>destinationType</jee:activation-config-property-name>
    <jee:activation-config-property-value>javax.jms.Queue</jee:activation-config-property-value>
  </jee:activation-config-property>
  <jee:activation-config-property>
    <jee:activation-config-property-name>hostName</jee:activation-config-property-name>
    <jee:activation-config-property-value>HOST.NAME</jee:activation-config-property-value>
  </jee:activation-config-property>
</jee:activation-config>
```

3.17.2.2 Using Annotation

On Windows:

```
@MessageDriven(

    activationConfig = {

@ActivationConfigProperty(propertyName = "securityExit", propertyValue = "biz.capitalware.mqausx.MQAUSXJ2EE"),
@ActivationConfigProperty(propertyName = "securityExitInit", propertyValue = "c:\\CAPITALWARE\\myclnt.enc")


@ActivationConfigProperty(propertyName = "channel", propertyValue="CHANNEL.NAME"),
@ActivationConfigProperty(propertyName = "destination", propertyValue = "QUEUE.NAME"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue"),
@ActivationConfigProperty(propertyName = "hostName", propertyValue = "HOST.NAME"),
@ActivationConfigProperty(propertyName = "port", propertyValue = "1414"),
@ActivationConfigProperty(propertyName = "transportType", propertyValue = "CLIENT"),

})

@ResourceAdapter(value=”wmq.jmsra.rar”)
```

On Unix / Linux:

```
@MessageDriven(

    activationConfig = {

@ActivationConfigProperty(propertyName = "securityExit", propertyValue = "biz.capitalware.mqausx.MQAUSXJ2EE"),
@ActivationConfigProperty(propertyName = "securityExitInit", propertyValue = "/apps/data/myclnt.enc")


@ActivationConfigProperty(propertyName = "channel", propertyValue="CHANNEL.NAME"),
@ActivationConfigProperty(propertyName = "destination", propertyValue = "QUEUE.NAME"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue"),
@ActivationConfigProperty(propertyName = "hostName", propertyValue = "HOST.NAME"),
@ActivationConfigProperty(propertyName = "port", propertyValue = "1414"),
@ActivationConfigProperty(propertyName = "transportType", propertyValue = "CLIENT"),

})

@ResourceAdapter(value=”wmq.jmsra.rar”)
```

3.18 Configuring Security Exit for Spring Framework

This section describes the necessary steps to enable Security Exits in Spring Framework. There are 3 ways the MQAUSX client-side channel security exit can be used with Spring Framework.

3.18.1 CCDT File

Use a CCDT to define the channel, and then use the ***ibm.mq.ccdtUrl*** property to point at it.

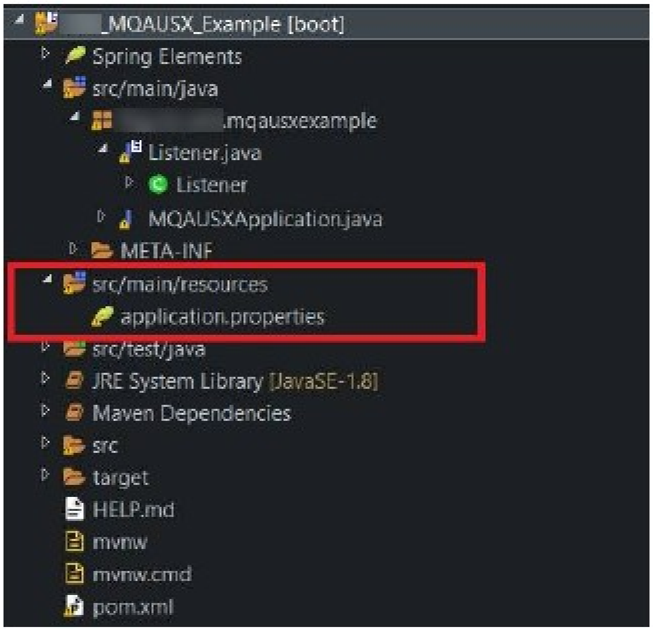
3.18.2 MQConnectionFactoryCustomizer Method

Use an MQConnectionFactoryCustomizer method in the application code to set the values of the security exit and security exit data.

```
@Bean
public MQConnectionFactoryCustomizer myCustomizer()
{
  MQConnectionFactoryCustomizer c = new MQConnectionFactoryCustomizer()
  {
    @Override
    public void customize(MQConnectionFactory factory)
    {
      factory.setStringProperty(MQConstants.WMQ_SECURITY_EXIT,"biz.capitalware.mqausx.MQAUSXJ2EE");
      factory.setStringProperty(MQConstants.XMSC_WMQ_SECURITY_EXIT_INIT,"/some/path/to/clnt.enc");
    }
  }
  return c;
}
```

3.18.3 MQ Spring Boot Module

Starting with MQ Spring Boot module v2.2.6 (or higher), you can use ***ibm.mq.additionalProperties.<propertyname>*** in the ***/resources/application.properties*** file to set channel security exit values.



```
ibm.mq.queueManager=MQA1
ibm.mq.channel=TEST.DEV.CHL
ibm.mq.connName=10.10.10.10(1414)
ibm.mq.additionalProperties.XMSC_WMQ_SECURITY_EXIT=biz.capitalware.mqausx.MQAUSXJ2EE
ibm.mq.additionalProperties.XMSC_WMQ_SECURITY_EXIT_INIT=/some/path/to/clnt.enc
```

3.19 Configuring Security Exit for use in J2EE Application Server

This section describes the necessary steps to enable Security Exits in a J2EE Application Server like WebLogic Server.

3.19.1 Dynamic Interaction via a Connection Factory

3.19.1.1 Updating Application Server's JVM Classpath

Windows:
The JAR file is located at (assuming a default install of **C:\Capitalware\MQAUSX**):

```
SET CLASSPATH=C:\Capitalware\MQAUSX\MQAUSXJ.jar;%CLASSPATH%
```

Unix and Linux (32-bit):
The JAR file is located at (assuming a default install of **/var/mqm/exits/**):

```
export CLASSPATH=/var/mqm/exits/MQAUSXJ.jar;%CLASSPATH%
```

Unix and Linux (64-bit):
The JAR file is located at (assuming a default install of **/var/mqm/exits64/**):

```
export CLASSPATH=/var/mqm/exits64/MQAUSXJ.jar:$CLASSPATH
```

3.19.1.2 Updating Application's JMS binding file

Use IBM MQ's JMSAdmin command to define or alter a QCF (QueueConnectionFactory) or TCF (TopicConnectionFactory). The client-side security exit also works with the XA versions of QCF and TCF (i.e. XAQCF and XATCF).

```
define tcf(tcfClient) qmgr(MY.QMGR) channel(SYSTEM.DEF.SVRCONN)
hostname(MYHOSTNAME) port(1414) transport(CLIENT)
SECEXIT(biz.capitalware.mqausx.MQAUSXJ2EE)

or

define qcf(qcfClient) qmgr(MY.QMGR) channel(SYSTEM.DEF.SVRCONN)
hostname(MYHOSTNAME) port(1414) transport(CLIENT)
SECEXIT(biz.capitalware.mqausx.MQAUSXJ2EE)
```

3.19.1.3 Application Execution

To pass the UserId and Password on the instantiation of the class, the Java J2EE code should look something like the following:

```
ConnectionFactory cf = (ConnectionFactory)ctx.lookup("MyQCF");
Connection conn = cf.createConnection("myUserId","myPswd");

Or

MQQueueConnectionFactory qcf = new MQQueueConnectionFactory();
QueueConnection qc =  qcf.createQueueConnection("myUserId","myPswd");
```

3.19.2 Batch or Quiet mode for J2EE based applications

To run in batch or quiet mode, the user can explicitly set the value of the UserId and Password in the channel's SecurityExitInit field or specify a file in the SecurityExitInit field.

To explicitly set the UserId and Password values, do the following for the user-defined client-side security exit for authentication:

3.19.2.1 Updating Application Server's JVM Classpath

Windows:
The JAR file is located at (assuming a default install of **C:\Capitalware\MQAUSX**):

```
SET CLASSPATH=C:\Capitalware\MQAUSX\MQAUSXJ.jar;%CLASSPATH%
```

Unix and Linux (32-bit):
The JAR file is located at (assuming a default install of **/var/mqm/exits/**):

```
export CLASSPATH=/var/mqm/exits/MQAUSXJ.jar;%CLASSPATH%
```

Unix and Linux (64-bit):
The JAR file is located at (assuming a default install of **/var/mqm/exits64/**):

```
export CLASSPATH=/var/mqm/exits64/MQAUSXJ.jar:$CLASSPATH
```

3.19.2.2 Updating Application's JMS binding file

Use IBM MQ's JMSAdmin command to define or alter a QCF (QueueConnectionFactory) or TCF (TopicConnectionFactory). The client-side security exit also works with the XA versions of QCF and TCF (i.e. XAQCF and XATCF). In the SecurityExitInit field, include the UserId and Password information as follows:

```
define tcf(tcfClient) qmgr(MY.QMGR) channel(SYSTEM.DEF.SVRCONN)
hostname(MYHOSTNAME) port(1414) transport(CLIENT)
SECEXIT(biz.capitalware.mqausx.MQAUSXJ2EE)
SECEXITINIT(u=fred;p=abcdef;s=ABC123)

or

define qcf(qcfClient) qmgr(MY.QMGR) channel(SYSTEM.DEF.SVRCONN)
hostname(MYHOSTNAME) port(1414) transport(CLIENT)
SECEXIT(biz.capitalware.mqausx.MQAUSXJ2EE)
SECEXITINIT(u=fred;p=abcdef;s=ABC123)
```

3.20 Configuring a Security Exit for use in Client Channel Definition Table

This section describes the necessary steps to enable Security Exits in third party applications that use the client channel definition table to connect to a queue manager.

3.20.1 CLNTCONN Channel

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

3.20.1.1 Windows

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

- SCYEXIT

```
C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
```

Leave 'SCYDATA' blank for GUI popup window mode.

- SCYDATA: For example: ''

To explicitly set the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows:

- SCYDATA

```
u=youruserid;p=yourpassword;s=remoteservername;q=qmgrpwd
```

where :

1. youruserid is the user's UserID
2. yourpassword is the user's Password
3. remoteservername is optional and is the Remote Server Name or Domain Controller (Windows only)
4. qmgrpwd is optional and is the queue manager's password

To use a file to hold the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows (please read Appendix A on how to format the file):

- SCYDATA

```
C:\Capitalware\MQAUSX\clnt.ini
```

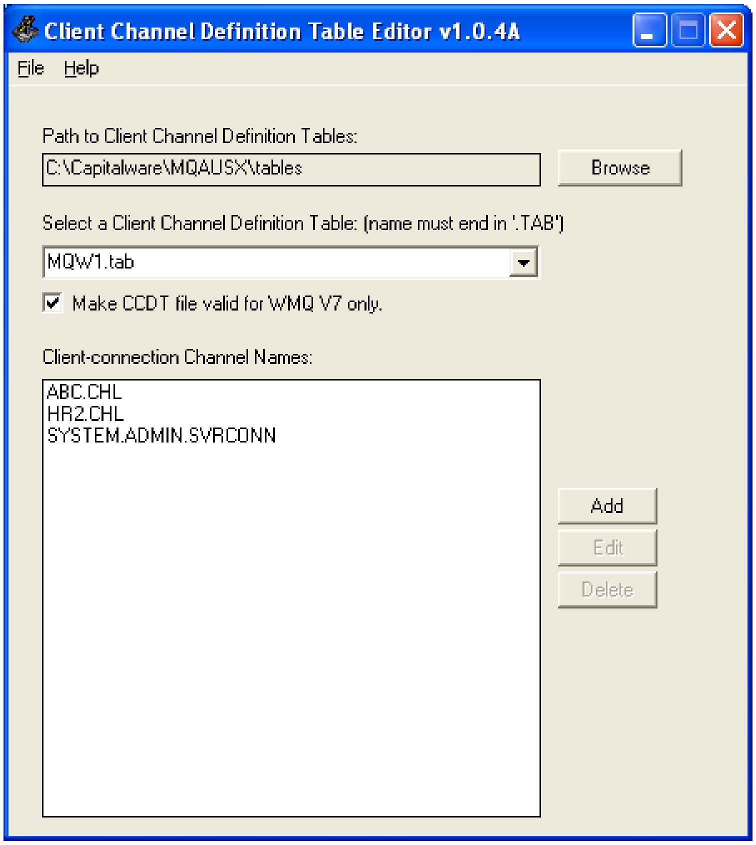
Or use an encrypted file. (see Appendix B for more information)

```
C:\Capitalware\MQAUSX\clnt.enc
```

Note: SCYDATA must NOT exceed 32 characters.

3.20.1.1.1.1Capitalware's Client Channel Definition Table Editor

Sample CLNTCONN for setting the client-side security exit to GUI popup window mode:



Path to Client Channel Definition Tables: C:\Capitalware\MQAUSX\tables

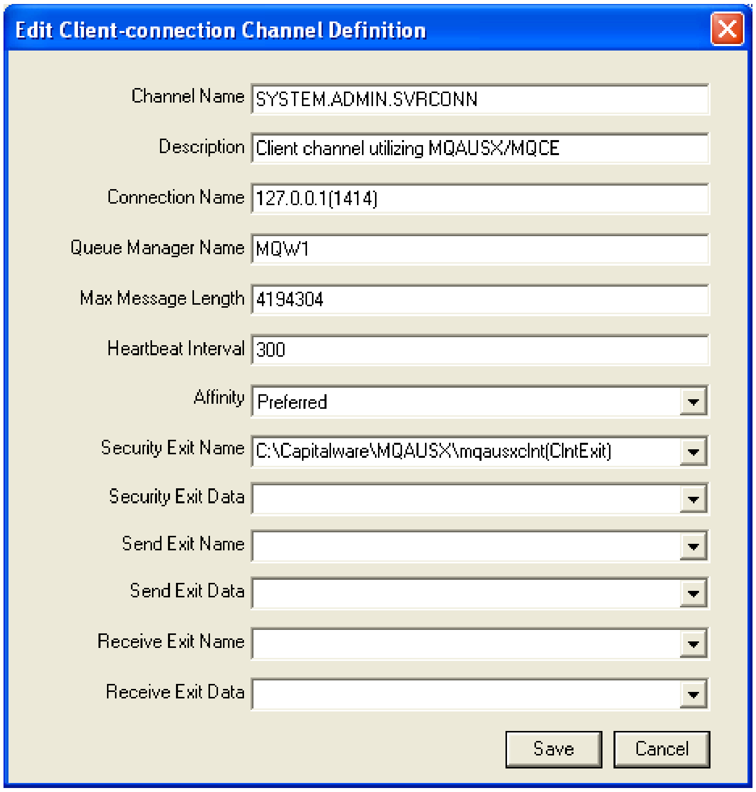
Select a Client Channel Definition Table: (name must end in '.TAB') MQW1.tab

☑ Make CCDT file valid for WMQ V7 only.

Client-connection Channel Names:

ABC.CHL
HR2.CHL
SYSTEM.ADMIN.SVRCONN

- Start the Client Channel Definition Table Editor (From the ***Start*** -> ***All Programs*** menu)
- Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')
- Click the Add button to insert a new CLNTCONN channel or click the Edit button to edit an existing CLNTCONN channel.



Channel Name: SYSTEM.ADMIN.SVRCONN

Description: Client channel utilizing MQAUSX/MQCE

Connection Name: 127.0.0.1(1414)

Queue Manager Name: MQW1

Max Message Length: 4194304

Heartbeat Interval: 300

Affinity: Preferred

Security Exit Name: C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)

Security Exit Data:

Send Exit Name:

Send Exit Data:

Receive Exit Name:

Receive Exit Data:

- For Security Exit Name, select ***C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)*** from the drop-down list.

A client channel definition table will be created in the 'tables' directory under the default install directory.

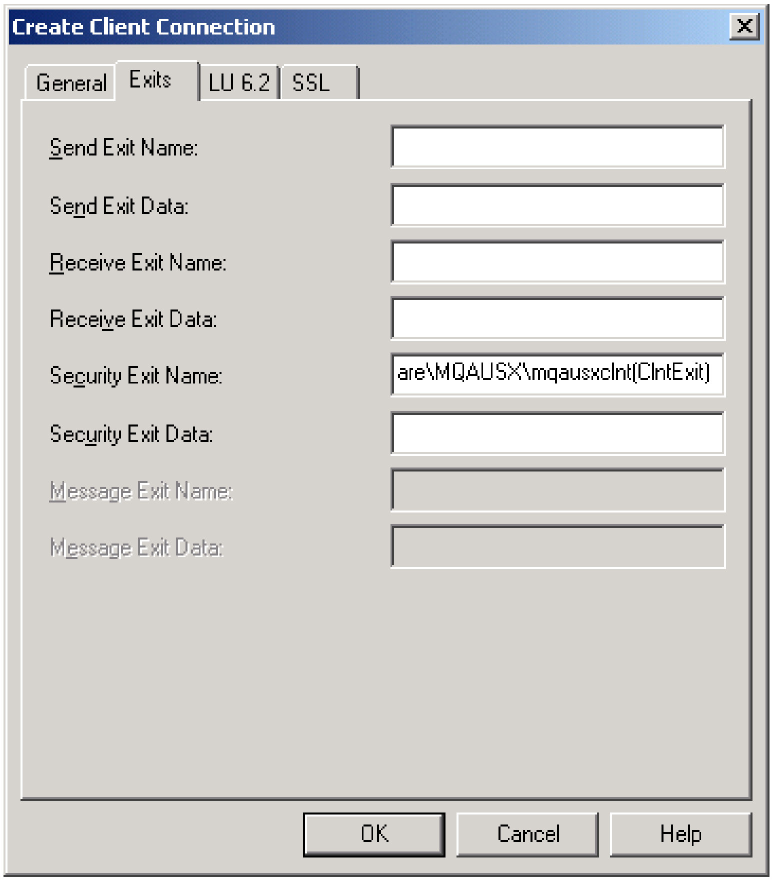
For the example above, a client channel definition table will be found (assuming a default install) at this location:

```
C:\Capitalware\MQAUSX\tables\MQW1.TAB
```

3.20.1.1.1.2MQ Explorer

Sample CLNTCONN for setting the client-side security exit to GUI popup window mode:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.1(1414)') TRPTYPE(TCP) +
       SCYEXIT('C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)') +
       SCYDATA(' ') +
       REPLACE
```



Security Exit Name: are\MQAUSX\mqausxclnt(ClntExit)

3.20.1.2 Unix and Linux for WebSphere v6.0, v7.1, v7.5 & v8.0 (32-bit)

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

- SCYEXIT
  `/var/mqm/exits/mqausxclnt(ClntExit)`

To explicitly set the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows:

- SCYDATA
  `u=youruserid;p=yourpassword;s=remoteservername;q=qmgrpwd`

Where :

1. youruserid is the user's UserID
2. yourpassword is the user's Password
3. remoteservername is optional and is the Remote Server Name or Domain Controller (Windows only)
4. qmgrpwd is optional and is the queue manager's password

To use a file to hold the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows (please read Appendix A for how to format the file):

- SCYEXIT
  `/var/mqm/exits/clnt.ini`

Or use an encrypted file. (see Appendix B for more information)
`/var/mqm/exits/clnt.enc`

Note: SCYDATA must NOT exceed 32 characters.

Note: The client-side security exit for z/OS, Unix and Linux does not support GUI popup mode.

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits/mqausxclnt(ClntExit)') +
       SCYDATA('/var/mqm/exits/clnt.ini') +
       REPLACE
```

3.20.1.3 Unix and Linux for WebSphere v6.0 (64-bit)

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

- SCYEXIT

`/var/mqm/exits64/mqausxclnt(ClntExit)`

To explicitly set the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows:

- SCYDATA

  `u=youruserid;p=yourpassword;s=remoteservername;q=qmgrpwd`

Where :

1. youruserid is the user's UserID
2. yourpassword is the user's Password
3. remoteservername is optional and is the Remote Server Name or Domain Controller (Windows only)
4. qmgrpwd is optional and is the queue manager's password

To use a file to hold the UserID, Password and Server (non-GUI mode) then set SCYDATA as follows (please read Appendix A for how to format the file):

- SCYEXIT

`/var/mqm/exits64/clnt.ini`

Or use an encrypted file. (see Appendix B for more information)
`/var/mqm/exits64/clnt.enc`

Note: SCYDATA must NOT exceed 32 characters.

Note: The client-side security exit for z/OS, Unix and Linux does not support GUI popup mode.

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqausxclnt(ClntExit)') +
       SCYDATA('/var/mqm/exits64/clnt.ini') +
       REPLACE
```

4 Configuring Security Exit in non popup mode

4.1 Client-side Security Exit using Environment Variables

This section describes the necessary steps to enable the client-side security exit to use environment variables or JVM environments to specify UserId and Password or a file that contains the UserId and Password.

The following describes the environment variables / JVM environment variables:

1. MQAUSX_UID - specifies the UserId to be used.
2. MQAUSX_PWD - specifies the password to be used.
3. MQAUSX_SERVER - specifies the server name to be used. (Optional).
4. MQAUSX_QMGR_PWD - specifies the queue manager password to be used. (Optional).
5. MQAUSX_FILE - specifies the file that will contain the UserId and Password values
6. MQAUSX_ENCFILE - specifies an encrypted file (created by the enc_clnt program) that will contain the UserId and encrypted Password values. The file name must end with 'enc'.

The client-side security exit handles internal processing of environment variables, SCYDATA and a popup security window as follows:

1. Checks for the existence of MQAUSX_UID and MQAUSX_PWD. If found, it is used; otherwise go to step # 2
2. Checks for the existence of MQAUSX_ENCFILE. If found, it is used; otherwise go to step #3
3. Checks for the existence of MQAUSX_FILE. If found, it is used; otherwise go to step #4
4. Checks for the existence of SCYDATA. If found, it is used; otherwise go to step # 5
5. Displays a popup security window to the end-user.

4.1.1 Native Applications

To use environment variables to specify the UserId and Pasword or a file, do the following for the user-defined client-side security exit for authentication:

4.1.1.1 Windows

Set MQAUSX_UID, MQAUSX_PWD and MQAUSX_SERVER **OR** set MQAUSX_ENCFILE **OR** set MQAUSX_FILE environment variables but do not set both groups of environment variables.

1. Set the following environment variables to specify UserId and Password as follows:

   ```
   set MQAUSX_UID=fred
   set MQAUSX_PWD=abcdef
   ```

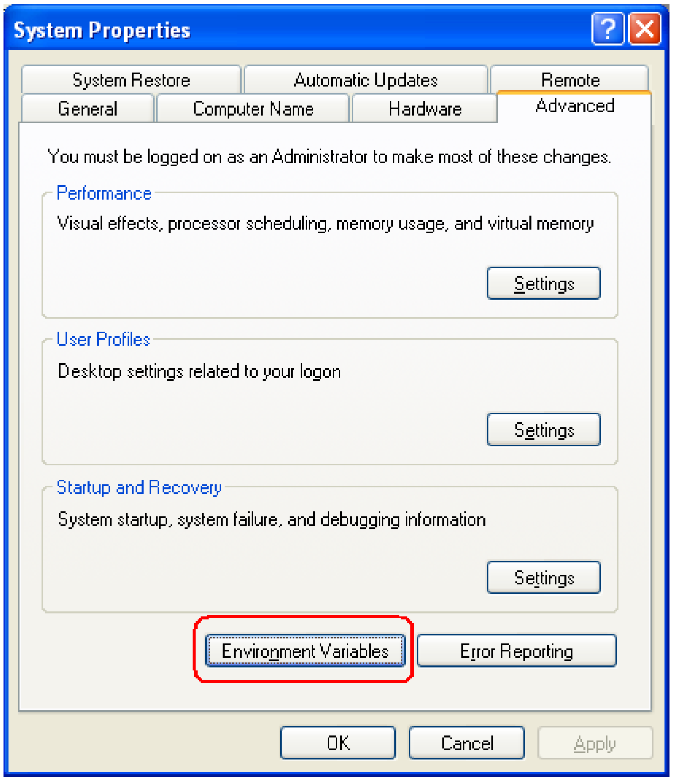
2. Set the following environment variable to specify a file that will contain the UserId and encrypted Password values as follows ('clnt.enc' file was created by enc_clnt program):

   ```
   set MQAUSX_ENCFILE=C:\Capitalware\MQAUSX\clnt.enc
   ```

3. Set the following environment variable to specify a file that will contain the UserId and Password values as follows:

   ```
   set MQAUSX_FILE=C:\Capitalware\MQAUSX\clnt.ini
   ```

On Windows, the user can globally set environment variables by going to the **System Properties** window of the **System** program of the **Control Panel** to select **Advanced** tab and clicking the **Environment Variables** button.



System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

4.1.1.2 Unix /Linux

Set MQAUSX_UID, MQAUSX_PWD and MQAUSX_SERVER **OR** set MQAUSX_FILE environment variables but do not set both groups of environment variables.

- Set the following environment variables to specify UserId and Password (server is optional) as follows:

  ```
  export MQAUSX_UID=fred
  export MQAUSX_PWD=abcdef
  ```

- Set the following environment variable to specify a file that will contain the UserId and encrypted Password values as follows ('clnt.enc' file was created by enc_clnt program):

  ```
  export MQAUSX_ENCFILE=/home/user/clnt.enc
  ```

- Set the following environment variable to specify a file that will contain the UserId and Password values as follows:

  ```
  export MQAUSX_FILE=/home/user/clnt.ini
  ```

4.1.2 Java based Applications

For Windows, Unix or Linux, set MQAUSX_UID, MQAUSX_PWD and MQAUSX_SERVER **OR** set MQAUSX_ENCFILE **OR** set MQAUSX_FILE JVM arguments. Do not set both groups of JVM arguments.

To use JVM arguments to specify the UserId and Pasword or a file that will contain the UserId and Password values, do the following:

1. Add the following JVM arguments to your java command-line parameters to specify the UserId and Password:

   ```
   java -DMQAUSX_UID=fred -DMQAUSX_PWD=abcdef   com.acme.run.Thing
   ```

2. Add the following JVM argument to your java command-line parameters to specify a file that will contain the UserId and encrypted Password values as follows ('clnt.enc' file was created by enc_clnt program)::

   On Windows:
   ```
   java -DMQAUSX_ENCFILE=C:\Capitalware\MQAUSX\clnt.enc  com.acme.run.Thing
   ```

   On Unix / Linux:
   ```
   java -DMQAUSX_ENCFILE=/home/user/clnt.enc  com.acme.run.Thing
   ```

3. Add the following JVM argument to your java command-line parameters to specify a file that will contain the UserId and Password values:

   On Windows:
   ```
   java -DMQAUSX_FILE=C:\Capitalware\MQAUSX\clnt.ini  com.acme.run.Thing
   ```

   On Unix / Linux:
   ```
   java -DMQAUSX_FILE=/home/user/clnt.ini  com.acme.run.Thing
   ```

4.2 Client-side Security Exit using Security Exit Data (SCYDATA)

This section describes the necessary steps to enable the client-side security exit to use Security Exit Data (SCYDATA).

Note: SCYDATA must NOT exceed 32 characters.

4.2.1 Directly from Security Exit Data

The parameters are as follows:

- **u** - specifies the UserId to be used.
- **p** - specifies the password to be used.
- **s** - specifies the remote server name to be used. {Optional}
- **q** - specifies the queue manager password to be used. {Optional}
- *any-other-value-ending-with-enc* - specifies an ecrypted file that will contain the UserId and encrypted Password values (file name must end with 'enc')
- *any-other-value* - specifies a file that will contain the UserId and Password values

4.2.1.1 Windows

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

To explicitly set the UserID, Password and Server (non-GUI mode), set SCYDATA as follows:

- ➢ SCYEXIT

```
C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
```

- ➢ SCYDATA

```
u=youruserid;p=yourpassword;s=remoteservername;q=qmgrpwd
```

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)') +
       SCYDATA('u=youruserid;p=yourpassword') +
       REPLACE
```

4.2.1.2 Unix and Linux for WebSphere v6.0, v7.0, v7.1, v7.5 & v8.0 (32-bit)

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

To explicitly set the UserID, Password and Server (non-GUI mode), set SCYDATA as follows:

- SCYEXIT
  `/var/mqm/exits/mqausxclnt(ClntExit)`

- SCYDATA
  `u=youruserid;p=yourpassword;s=remoteservername;q=qmgrpwd`

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits/mqausxclnt(ClntExit)') +
       SCYDATA('u=youruserid;p=yourpassword') +
       REPLACE
```

4.2.1.3 Unix and Linux for WebSphere v6.0 or v7.0 (64-bit)

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

To explicitly set the UserID, Password and Server (non-GUI mode), set SCYDATA as follows:

- SCYEXIT
  `/var/mqm/exits64/mqausxclnt(ClntExit)`

- SCYDATA
  `u=youruserid;p=yourpassword;s=remoteservername`

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqausxclnt(ClntExit)') +
       SCYDATA('u=youruserid;p=yourpassword') +
       REPLACE
```

4.2.2 Indirectly from an IniFile or MQAUSX Encrypted File

To use a file to store the UserID, Password and Server (non-GUI mode), set SCYDATA with an IniFile (refer to ***Appendix A*** for how to format the file). The client-side security exit for z/OS, Unix and Linux does not support GUI popup mode.

To use an MQAUSX client-side encrypted file to hold the UserID, encrypted Password and Server , set SCYDATA with an encrypted file (refer to ***Appendix B***).

Note: SCYDATA must NOT exceed 32 characters.

4.2.2.1 Windows

On Windows, SCYEXIT and SCYDATA will contain the following values (assuming a default install):

- SCYEXIT
  `C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)`

- SCYDATA

  For an IniFile:
  `C:\Capitalware\MQAUSX\clnt.ini`

  For an Encrypted File:
  `C:\Capitalware\MQAUSX\clnt.enc`

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)') +
       SCYDATA('C:\Capitalware\MQAUSX\clnt.ini') +
       REPLACE
```

4.2.2.2 Unix and Linux for WebSphere v6.0, v7.0, v7.1, v7.5 or v8.0 (32-bit)

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

- SCYEXIT
  `/var/mqm/exits/mqausxclnt(ClntExit)`

- SCYDATA

  IniFile:
  `/var/mqm/exits/clnt.ini`

  Encrypted File:
  `/var/mqm/exits/clnt.enc`

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits/mqausxclnt(ClntExit)') +
       SCYDATA('/var/mqm/exits/clnt.ini') +
       REPLACE
```

4.2.2.3 Unix and Linux for WebSphere v6.0, v7.0, v7.1, v7.5 & v8.0 (64-bit)

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

- SCYEXIT
  `/var/mqm/exits64/mqausxclnt(ClntExit)`

- SCYEXIT

  IniFile:
  `/var/mqm/exits64/clnt.ini`

  Encrypted File:
  `/var/mqm/exits64/clnt.enc`

Here is a sample MQSC definition of a CLNTCONN channel:

```
DEFINE CHANNEL ('TEST.CLIENT.CH01') CHLTYPE(CLNTCONN) +
       CONNAME('10.1.10.2(1414)') TRPTYPE(TCP) +
       SCYEXIT('/var/mqm/exits64/mqausxclnt(ClntExit)') +
       SCYDATA('/var/mqm/exits64/clnt.ini') +
       REPLACE
```

5 Appendix A - mqausxclnt.ini file (optional)

The table below is the supplied mqausxclnt.ini file. The IniFile supports the following keywords and their values.

```
LogMode=N
LogFile=C:\Capitalware\MQAUSX\mqausxclnt.log
UserID=fred
Password=abcdef
```

Note: Keywords are case sensitive.

Keyword	Description of client-side keywords

LogFile	**LogFile** specifies the location of the log file.
	For Windows:
	LogFile=C:\Capitalware\MQAUSX\mqausxclnt.log
	For Unix and Linux for IBM MQ (32-bit):
	LogFile=/var/mqm/exits/mqausxclnt.log
	For Unix and Linux for IBM MQ (64-bit):
	LogFile=/var/mqm/exits64/mqausxclnt.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
Password	**Password** specifies the password to be used for authentication. e.g. Password=abcdef
QMgrPassword	**QMgrPassword** specifies the queue manager password to be used for authentication. e.g. QMgrPassword=xyz123
RejectConName	**RejectConName** specifies a list of connection names that the exit will not connect to. e.g. RejectConName=192.168.10.*;192.168.20.*

Keyword	Description of client-side keywords
RejectQMgrName	**RejectQMgrName** specifies a list of queue manager names that the exit will not connect to. e.g. RejectQMgrName=MQWT1;MQWT2
ServerName	**ServerName** specifies a default server name for this entity. This value will be transmitted to the end-user. For a Windows Server, you may specify a domain name. The default is the hostname. e.g. ServerName=ABC123
SSO_TimeOut	**SSO_TimeOut** specifies the length of time to keep the encrypted client credentials in shared memory (value in minutes). e.g. SSO_TimeOut=5
UserID	UserID specifies the UserID to be used for authentication. e.g. UserID=fred

6 Appendix B - Client-side Encrypted File

The user can create a file that will contain the UserId, encrypted Password and remote ServerName. The enc_clnt program is used to create a file that will contain the client-side UserId, encrypted Password and remote ServerName.

Syntax:

```
enc_clnt -u UserId -p Password [-s ServerName] [-q QMgrPassword]
       [-f out_filename] [-R 1|2]
```

Where :

- UserId is the user's remote UserID (remote Logon Id)
- Password is the user's Password to be encrypted
- ServerName is the remote Server Name (optional)
- QmgrPassword is the queue manager's password (optional)
- out_filename is the output file name (optional)
- R (release) and the default is 2 (AES encryption). Use 1 for older TEA encryption

6.1 Windows

To use the enc_clnt program on Windows, open a Command prompt and change directory to `C:\Capitalware\MQAUSX\`

The following command will create a file called 'clnt.enc' in the current directory with the UserId and encrypted password:

```
enc_clnt.exe -u barney -p bedrock
```

To specify a different path and/or filename (the file extension must be 'enc'), do following command:

```
enc_clnt.exe -u barney -p bedrock -f C:\temp\myclnt.enc
```

6.2 Unix and Linux for WebSphere v6.0, v7.0, v7.1, v7.5 & v8.0 (32-bit)

To use the enc_clnt program on Unix/Linux for MQ v6.0, v7.0, v7.1, v7.5 & v8.0, open a shell prompt and change directory to `/var/mqm/exits/`

The following command will create a file called 'clnt.enc' in the current directory with the UserId and encrypted password:

```
enc_clnt -u barney -p bedrock
```

To specify a different path and/or filename (the file extension must be 'enc'), do following command:

```
enc_clnt -u barney -p bedrock -f /tmp/myclnt.enc
```

6.3 Unix and Linux for WebSphere v6.0, v7.0, v7.1, v7.5 & v8.0 (64-bit)

To use the enc_clnt program on Unix/Linux for MQ v6.0, v7.0, v7.1, v7.5 & v8.0 (64-bit), open a shell prompt and change directory to `/var/mqm/exits64/`

The following command will create a file called 'clnt.enc' in the current directory with the UserId and encrypted password:

```
enc_clnt -u barney -p bedrock
```

To specify a different path and/or filename (the file extension must be 'enc'), do following command:

```
enc_clnt -u barney -p bedrock -f /tmp/myclnt.enc
```

6.4 IBM i

To use the enc_clnt program on IBM i (OS/400), open a shell prompt (`QSH`) and change directory to `/QIBM/UserData/mqm/`

The following command will create a file called 'clnt.enc' in the current directory with the UserId and encrypted password:

```
CALL MQAUSX/ENC_CLNT PARM('-u' 'barney' '-p' 'bedrock')
```

To specify a different path and/or filename (the file extension must be 'enc'), do following command:

```
CALL MQAUSX/ENC_CLNT PARM('-u' 'barney' '-p' 'bedrock' '-f'
'/QIBM/UserData/mqm/mqausx/enc_clnt.enc')
```

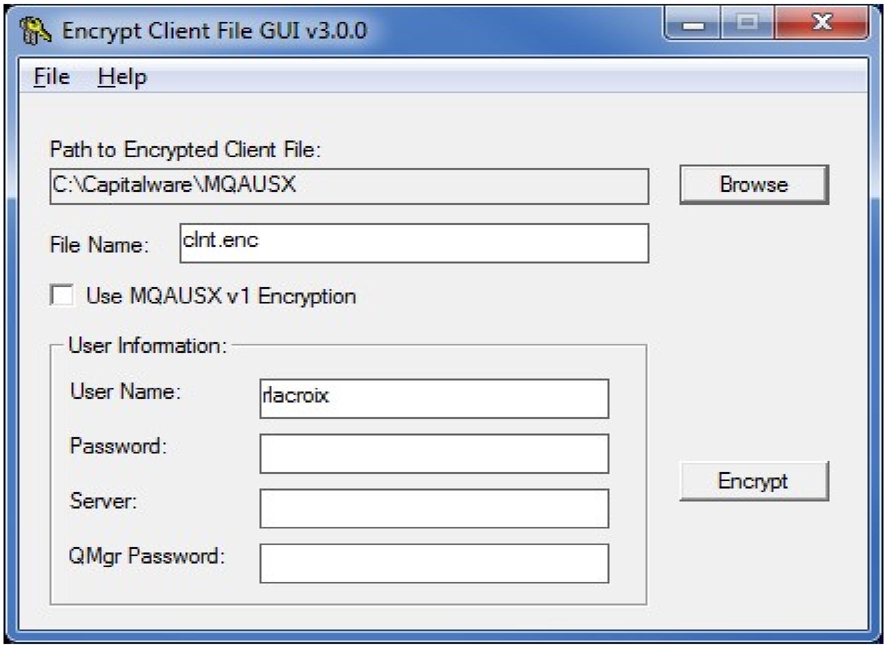
6.5 z/OS

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

```
//ENCCLNT  EXEC PGM=ENCCLNT,
//         PARM='-u barney -p bedrock -f CLNTENC'
//SYSPRINT DD  SYSOUT=*
//STEPLIB  DD  DISP=SHR,DSN=+HLQ+.CPTLWARE.MQAUSX.LOAD
//CLNTENC  DD  DISP=SHR,DSN=+HLQ+.CPTLWARE.MQAUSX.CLNT(MEM-NAME)
```

7 Appendix C - Client-side Encrypted File Windows GUI

MQAUSX client-side security exit installation package includes a new tool called: ***Encrypt Client File***. The Encrypt Client File is a Windows GUI program that enables the user to quickly create an encrypted client file.



Path to Encrypted Client File: C:\Capitalware\MQAUSX

File Name: clnt.enc

□ Use MQAUSX v1 Encryption

User Information:

User Name: rlacroix

Password:

Server:

QMgr Password:

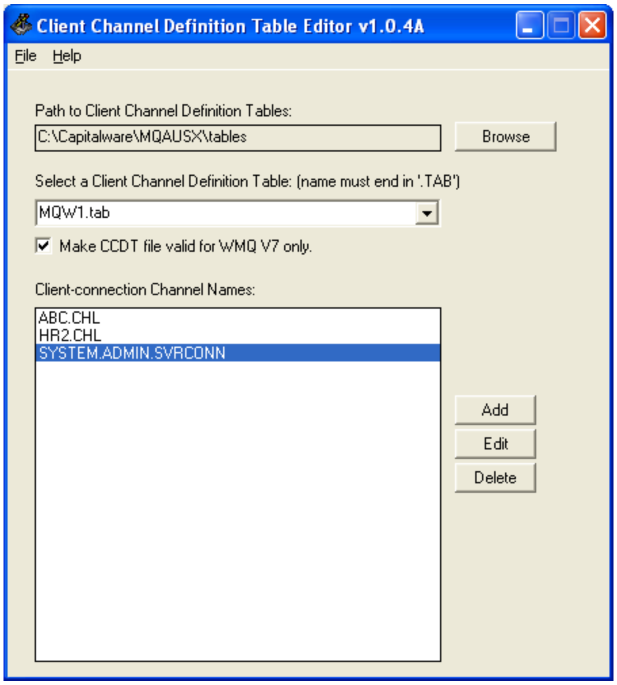
- To start the Encrypt Client File, click ***Start*** -> ***All Programs*** -> ***MQ Authenticate User Security Exit - Client*** -> ***Encrypt Client File***
- Use the ***Browse*** button to set the directory where the encrypted file is to be stored.
- Input the name of the client encrypted file name. Note: The file type must be “enc”.
- Input the UserId and Password to stored in the encrypted client IniFile.
- Optional: Input a Server Name to be sent to the server-side security exit.
- Optional: Input the queue manager password.
- Click the ***Encrypt*** button to create the encrypted client file.

8 Appendix D - Client Channel Definition Table Editor

MQAUSX client-side security exit installation package includes a new tool called: ***Client Channel Definition Table Editor***. The Client Channel Definition Table Editor is a Windows GUI program that enables the user to quickly create a Client Channel Definition Table or to edit an existing table in order to add, update and delete CLNTCONN channels.

The Client Channel Definition Table Editor does not require IBM MQ Server or IBM MQ Client to be installed on the PC. The Client Channel Definition Table Editor uses SupportPac MO72 to perform the adding, updating and deleting of CLNTCONN channels of an MQ client channel definition table.

- To start the Client Channel Definition Table Editor, click ***Start*** -> ***All Programs*** -> ***MQ Authenticate User Security Exit - Client*** -> ***Client Channel Definition Table Editor***
- Select the client channel definition table to be edited from the drop-down list or input a new client channel definition table name (the name MUST end in '.tab')
- Click the ***Add*** button to insert a new CLNTCONN channel or click the ***Edit*** button to edit an existing CLNTCONN channel.



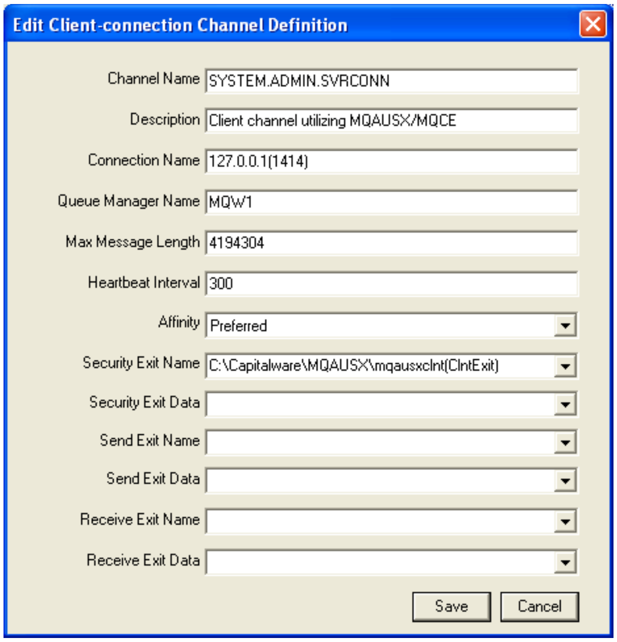
Path to Client Channel Definition Tables: C:\Capitalware\MQAUSX\tables

Select a Client Channel Definition Table: (name must end in '.TAB') MQW1.tab

☑ Make CCDT file valid for WMQ V7 only.

Client-connection Channel Names:

ABC.CHL
HR2.CHL
SYSTEM.ADMIN.SVRCONN



Channel Name	SYSTEM.ADMIN.SVRCONN
Description	Client channel utilizing MQAUSX/MQCE
Connection Name	127.0.0.1(1414)
Queue Manager Name	MQW1
Max Message Length	4194304
Heartbeat Interval	300
Affinity	Preferred
Security Exit Name	C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)
Security Exit Data	
Send Exit Name	
Send Exit Data	
Receive Exit Name	
Receive Exit Data	

For the ***Security Exit Name*** field, the user can input their own data or use 1 of the 6 predefined values as shown below:

Values	Description
biz.capitalware.mqausx.MQAUSXJ	Use this value for stand-alone Java applications.
biz.capitalware.mqausx.MQAUSXJE6	Use this value for MQ Explorer v6.
biz.capitalware.mqausx.MQAUSXJ2EE	Use this value for J2EE applications.
C:\Capitalware\MQAUSX\mqausxclnt(ClntExit)	Use this value for native Windows applications.
/var/mqm/exits64/mqausxclnt(ClntExit)	Use this value for native Unix/Linux 64-bit applications.
/var/mqm/exits/mqausxclnt(ClntExit)	Use this value for native Unix/Linux 32-bit applications.

For the ***Security Exit Data*** field, the user can input their own data or use 1 of the 3 predefined values as shown below:

Values	Description
C:\Capitalware\MQAUSX\clnt.ini	Use this value for native Windows applications.
/var/mqm/exits64/clnt.ini	Use this value for native Unix/Linux 64-bit applications.
/var/mqm/exits/clnt.ini	Use this value for native Unix/Linux 32-bit applications.

A client channel definition table will be created in the 'tables' directory under the default install directory. For the example above, a client channel definition table will be found (assuming a default install) at this location:

```
C:\Capitalware\MQAUSX\tables\MQW1.TAB
```

9 Appendix E – Client-side Environment Variables

The following describes the environment variables / JVM environment variables are available for native client-side security exit, MQAUSXJ Java client-side security exit and MQAUSXDN DotNet client-side security exit:

- **MQAUSX_UID** specifies the UserId to be used.
- **MQAUSX_PWD** specifies the password to be used.
- **MQAUSX_SERVER** specifies the server name to be used. (Optional).
- **MQAUSX_QMGR_PWD** specifies the queue manager pasword to be used. (Optional).
- **MQAUSX_FILE** specifies the file that will contain the UserId and Password values
- **MQAUSX_ENCFILE** specifies an encrypted file (created by the enc_clnt program) that will contain the UserId and encrypted Password values. The file name must end with 'enc'.
- **MQAUSX_REJECT_CONNAME** specifies a list of connection names that the exit will not connect to
- **MQAUSX_REJECT_QMGR_NAME** specifies a list of queue manager names that the exit will not connect to
- **MQAUSX_DEBUG** specifies that the client-side security exit is to output debug information to a log file
- **MQAUSXCLNT_HOME** specifies the location of the client-side IniFile.

The following describes the environment variables / JVM environment variables are only available for Java MQAUSXJ class and the native Windows DLL (not MQAUSXDN):

- **MQAUSX_NO_SSO** specifies to not to use the built-in Single Sign On (SSO) feature
- **MQAUSX_SSO_TIMEOUT** specifies the length of time to keep the encrypted client credentials in shared memory (value in minutes).

10 Appendix F – Client-side Single Sign On (SSO)

The following section describes items related to Single Sign On (SSO). The MQAUSX client-side security exit uses the SSO feature so that the user is not inundated with popup windows requesting the user's UserID and Password. By default, the SSO feature will store the user credentials in encrypted format in shared memory for up to 12 hours. The SSO feature is only available for Java MQAUSXJ class and Windows DLL.

SSO related environment variables / JVM environment variables:

- **MQAUSX_NO_SSO** specifies to not to use the built-in Single Sign On (SSO) feature
- **MQAUSX_SSO_TIMEOUT** specifies the length of time to keep the encrypted client credentials in shared memory (value in minutes).

SSO related keywords:

- **SSO_TimeOut** specifies the length of time to keep the encrypted client credentials in shared memory (value in minutes).

The table below is a sample SSO Group IniFile. The SSO Group IniFile must be called: ***mqausx_group.ini*** and is located in the user's home directory or the directory contained in the MQAUSXCLNT_HOME environment variable. The IniFile supports the following keywords and their values.

```
Group1=192.168.10.*;192;168.20.*
Group2=svrab*;svrxx*
Group3=abc*;xyz*
```

Note: Keywords are case sensitive.

Keyword	Description of client-side keywords
Group#	Group# specifies a group of IP addresses and / or hostnames where the UserId and Password are the same. e.g. Group1=192.168.10.*;192;168.20.*

11 Appendix G - Encryption

MQ Authenticate User Security Exit Solution uses the Advanced Encryption Standard (AES) for encryption and decryption of the user's password between the client-side security exit and the server-side security exit.

Wikipedia

> *the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor,[3] the Data Encryption Standard (DES).*
>
> *AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable (see Advanced Encryption Standard process for more details). It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce. It is available in many different encryption packages. AES is the first publicly accessible and open cipher approved by the NSA for top secret information*

12 Appendix H - License Agreement

This is a legal agreement between you (either an individual or an entity) and Capitalware Inc. By opening the sealed software packages (if appropriate) and/or by using the SOFTWARE, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, promptly return the disk package and accompanying items for a full refund.
SOFTWARE LICENSE

1. GRANT OF LICENSE. This License Agreement (License) permits you to use one copy of the software product identified above, which may include user documentation provided in on-line or electronic form (SOFTWARE). The SOFTWARE is licensed as a single product, to an individual user, or group of users for Muliple User Licenses and Site Licenses. This Agreement requires that each user of the SOFTWARE be Licensed, either individually, or as part of a group. A Multi-User License provides for a specified number of users to use this SOFTWARE at any time. This does not provide for concurrent user Licensing. Each user of this SOFTWARE must be covered either individually, or as part of a group Multi-User License. The SOFTWARE is in use on a computer when it is loaded into the temporary memory (i.e. RAM) or installed into the permanent memory (e.g. hard disk) of that computer. This software may be installed on a network provided that appropriate restrictions are in place limiting the use to registered users only.

2. COPYRIGHT. The SOFTWARE is owned by Capitalware Inc. and is protected by United States Of America and Canada copyright laws and international treaty provisions. You may not copy the printed materials accompanying the SOFTWARE (if any), nor print copies of any user documentation provided in on-line or electronic form. You must not redistribute the registration codes provided, either on paper, electronically, or as stored in the files mqausx.ini or any other form.

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

LIMITED WARRANTY

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

CUSTOMER REMEDIES. Capitalware Inc. entire liability and your exclusive remedy shall be, at Capitalware Inc. option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet this Limited Warranty and that is returned to Capitalware Inc. with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

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

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

13 Appendix I - 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.