Authentication with MQAUSX & z/MQAUSX

Roger Lacroix
roger.lacroix@capitalware.com
http://www.capitalware.com

MQAUSX & z/MQAUSX Overview

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

MQAUSX & z/MQAUSX Overview

- The **same** 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, encrypt the data and send it to the server-side security exit.

MQAUSX & z/MQAUSX are 4 products in 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.***
- 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 (see #3). ***Excellent.***
- 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
- If the MQAdmin sets the MQAUSX IniFile parameter NoAuth to Y then it functions just like MQ Standard Security Exit (MQSSX or z/MQSSX). MQSSX does not authenticate. It filters the incoming connection based on UserID, IP address, hostname and/or SSL DN.

MQAUSX Secondary Features

- Allows or restricts the incoming UserID against a Group
- Provides support for Proxy UserIDs
- Allows or restricts the incoming IP address against a regular expression pattern
- Allows or restricts the incoming SSL DN against a regular expression pattern
- Allows or restricts the incoming UserID against a regular expression pattern
- Allows or restricts the incoming Active Directory server name against a regular expression pattern (Windows only)
- Limit the number of incoming channel connections on a SVRCONN channel.
- Allows or restricts the use of ‘mqm’, ‘MUSER_MQADMIN’ or ‘QMQM’ UserIDs
- Ability to turn off server-side authentication
- Provides monitoring tool tie-in by using custom MQ event messages
- Provides logging capability for all connecting client applications regardless if they are successful or not.

z/MQAUSX Secondary Features

- Allows or restricts the incoming UserID against a Group
- Provides support for Proxy UserIDs
- Allows or restricts the incoming IP address against a regular expression pattern
- Allows or restricts the incoming hostname against a regular expression pattern
- Limit the number of incoming channel connections on a SVRCONN channel.
- Allows or restricts the use of ‘CHIN’ or the CHIN’s Started-task UserIds
- Ability to turn off server-side authentication
- Allows or restricts the incoming UserID against a regular expression pattern when authentication is off
- Provides logging capability for all connecting client applications regardless if they were successful or not.
- Provides logging capability via Write To Operator (WTO) facility.

New Feature #1 in MQAUSX & z/MQAUSX

- Problem: How to add a higher-level of security to MQ than just authentication?

I was doing a lot of research on factors of authentication. i.e. Something you know, Something you have, and Something you are. Two-factor and multi-factor authentication are great when the system is protecting an end-user from hackers logging into their account. The problem is that only 5% (or less) of the connection attempts that MQAUSX processes are from actual people. 95% (or more) of the connection attempts are from back-end applications. Hence, having the back-end application use a chip-card or RSA fob is just not possible nor is the use of biometrics (fingerprint or voice print). I'll really become worried when a back-end application has a voice or finger!!!!

New Feature #1 in MQAUSX & z/MQAUSX

- Solution: Assign a Password to the Queue Manager.

An MQAdmin could define a Password for a queue manager via the MQAUSX configuration file (it would be encrypted of course). So, 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 like adding perimeter security to your system or putting your valuables in a safe and putting that safe in another safe.

- Queue Manager Password is available for MQAUSX v3.0.0 and z/MQAUSX v3.0.0.

New Feature #2 in MQAUSX & z/MQAUSX

- Problem

A customer wanted to know when applications issued an excessive number of connection requests over a particular period of time. (Connecting and disconnecting over & over again).

New Feature #2 in MQAUSX & z/MQAUSX

- Solution: Excessive Client Connections (ECC)

ECC is an alert 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. The reason the customer requested the ECC feature, is 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.

- ECC is available for MQAUSX v3.0.0 and z/MQAUSX v3.0.0.

New Feature #3 in MQAUSX & z/MQAUSX

- Problem

A customer has several applications that, shall we say, are poorly written and they will not likely be changed. These particular applications connect, open, put (and/or get), close and disconnect from the queue manager (over & over again). The applications generate up to 700 connection requests per minute (over 12,000 per hour). The MQAdmin recently changed the MQAUSX authentication target from Local OS to LDAP over SSL. This caused a noticeable impact on their LDAP server to the point that the LDAP server is intermittently unavailable.

New Feature #3 in MQAUSX & z/MQAUSX

- Solution: Credential Cache

MQAUSX will cache (when enabled) the user credentials (in an encrypted format) for 'x' minutes (default is 5 minutes) in shared memory. Hence, when there is a new connection, MQAUSX will first check the cache for the incoming UserID and if found then the entry's timestamp will be checked. If the cache entry has expired then the entry is removed from the cache. If the entry is valid then the cached password is compared to the incoming password. If the passwords match then the connection is allowed. If the passwords do not match then the entry is removed from the cache and MQAUSX will perform an authentication against the target (i.e. LDAP).

- Credential Cache is available now (v3.0.0.1). Will be officially announced for MQAUSX v3.1.0 and z/MQAUSX v3.1.0.

MQAUSX & z/MQAUSX IniFiles

- MQAUSX and z/MQAUSX use initialization files (IniFiles) to configure how the security exit is to run.
- One IniFile can be used for all of the channels of a queue manager.
- An IniFile can be used for a group of the channels of a queue manager.
- Every channel of a queue manager can have its own IniFile.

IBM MQ V8 Authentication

- IBM MQ V8 authenticates the user's UserId and Password against the server's native OS system or LDAP server.
- You cannot have a mixture of authentication types because authentication type is set at the queue manager level.

```
ALTER QMGR CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)

REFRESH SECURITY TYPE(CONNAUTH)

Or

DEFINE AUTHINFO('USE.LDAP') AUTHTYPE(IDPWLDAP) CONNAME('superldap.servers.uk') SHORTUSR('uid')
ADOPTCTX(NO) USRFIELD('uid') BASEDNU('ou=MQ,o=IBM,co=UK') CHCKCLNT(OPTIONAL)
CHCKLOCL(OPTIONAL) CLASSUSR('account') SECCOMM(NO)

ALTER QMGR CONNAUTH('USE.LDAP')

REFRESH SECURITY TYPE(CONNAUTH)
```

MQAUSX vs IBM MQ V8 Authentication

Authentication Functionality	IBM MQ V8	MQAUSX
Authentication against Local OS	Yes	Yes
Authentication against LDAP Server	Yes	Yes
Authentication against LDAP Server using SSL	Yes	Yes
Authentication against MS Active Directory from Windows	No*	Yes
Number of LDAP calls to perform Authentication	2	1*
Authentication against Quest Authentication Services	No	Yes
Authentication against Centrify's DirectControl	No	Yes
Authentication against PAM	Yes	Yes
Authentication against RACF – z/OS only	Yes	Yes
Authentication against ACF2 – z/OS only	Yes	Yes
Authentication against TopSecret – z/OS only	Yes	Yes
Authentication against File Based Authentication	No	Yes
Ability to use more than 1 authentication type per QMgr	No	Yes
Ability to set authentication order	No	Yes

MQAUSX vs IBM MQ V8 Authentication

Functionality	IBM MQ V8	MQAUSX
Only allow the connection if the UserId exists in a particular Group	No	Yes
Assign a Password to a Queue Manager	No	Yes
Credential Cache	No	Yes
Allow/Reject by IP Address	Yes	Yes
Allow/Reject by Hostname (DNS)	Yes	Yes
Allow/Reject by Host by Name	No	Yes
Allow/Reject by SSL DN	Yes	Yes
Allow/Reject by UserId	Yes	Yes
Allow/Reject by MS Active Directory Name	No	Yes
Ability to Reject Self Signed Certificates	No	Yes

MQAUSX vs IBM MQ V8 Authentication

Functionality	IBM MQ V8	MQAUSX
Limit the number of connections by Channel	Yes	Yes
Generate an alert when number of connections by Channel reaches a certain percentage	No	Yes
Ability to secure cluster channels	Yes	Yes
Map incoming UserID to another UserId to be used as the connection MCAUSER value	Yes	Yes
Map SSL UserId to the connection MCAUSER value	Yes	Yes
Map the channel's SSLCertUserID to the connection MCAUSER value – z/OS only	No	Yes
Excessive Client Connections	No	Yes
Logging of successful connections	Partial	Yes
Logging of failed connection attempts	Yes	Yes
Write event message for failed connection attempts	Yes*	Yes

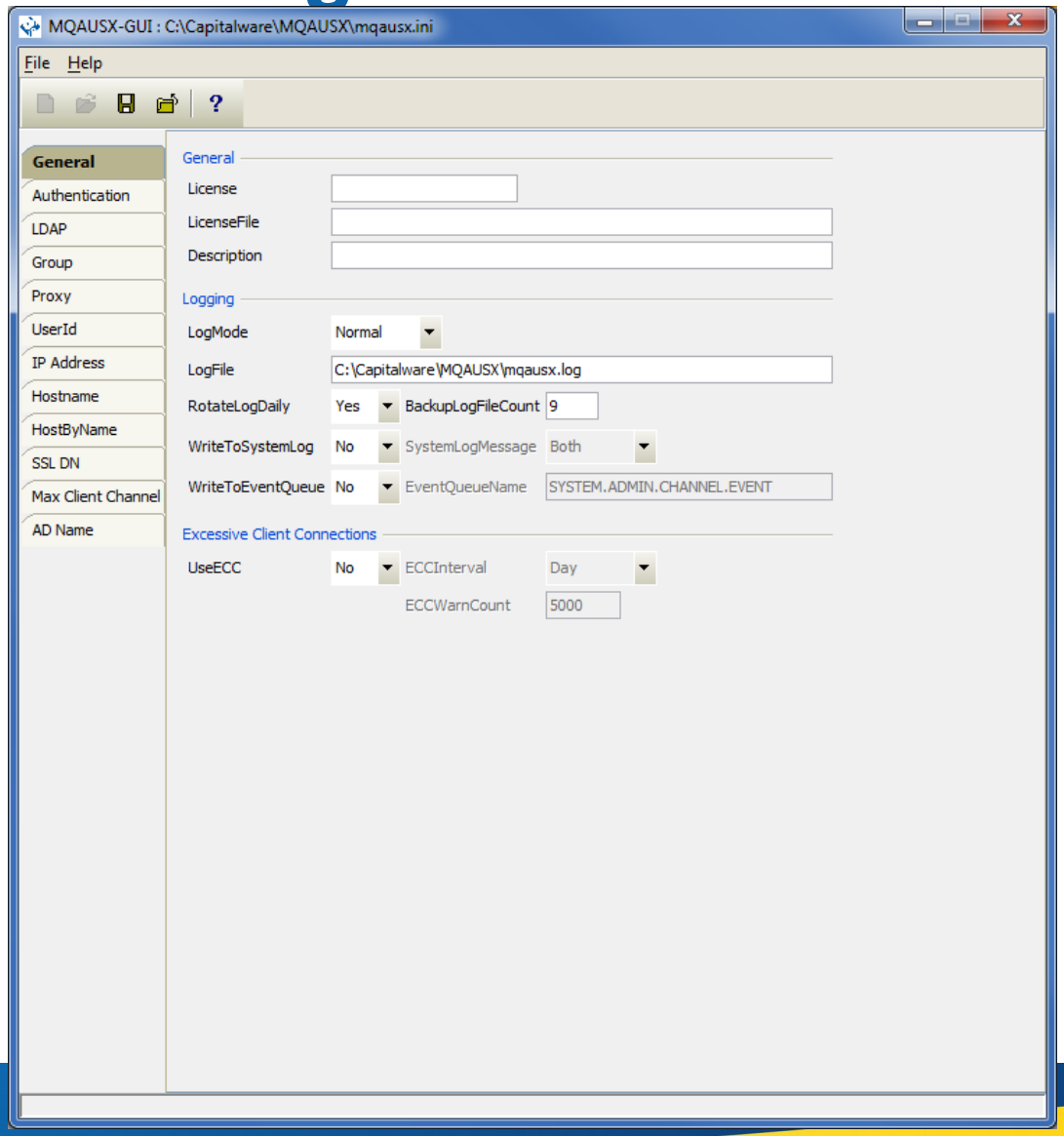
MQAUSX vs IBM MQ V8 Authentication

- MQAUSX and z/MQAUSX can authenticate UserID and Passwords (plain text – no client-side security exit) sent by non MQ V8 clients.
- MQAUSX and z/MQAUSX will log the MQAUSX client-side security exit's platform, version and type (Java/.NET/Native).
- MQAUSX and z/MQAUSX will log not only the incoming UserID for authentication, it will log the UserID that the application is running under (keep UserID spoofing to a minimum).

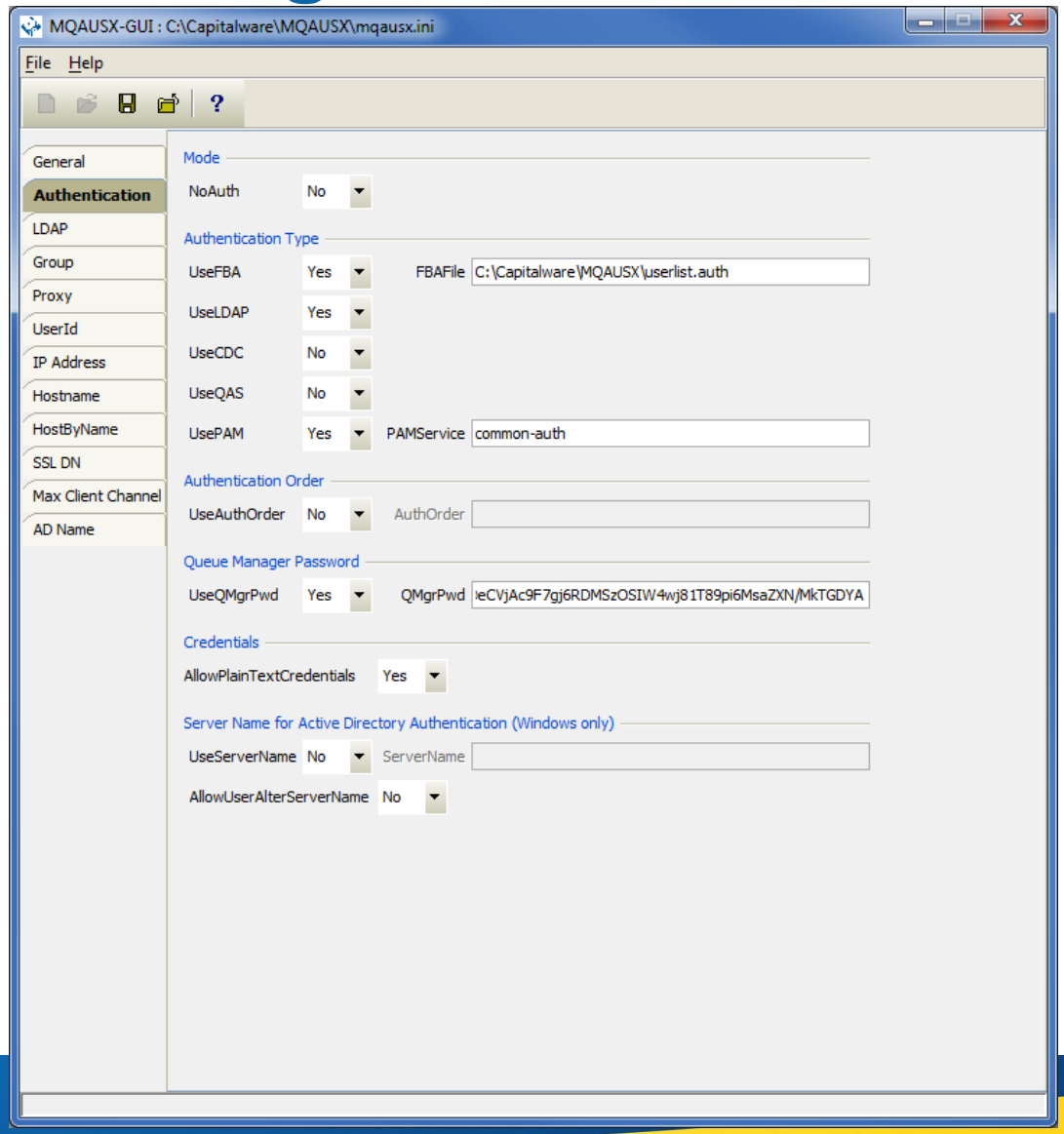
MQAUSX & z/MQAUSX Keywords

- MQAUSX has 112 keywords and values that can be used.
- z/MQAUSX has 71 keywords and values that can be used.
- It is best to use either MQAUSX-GUI or MQAUSX-ISPF-GUI to update values for keywords.

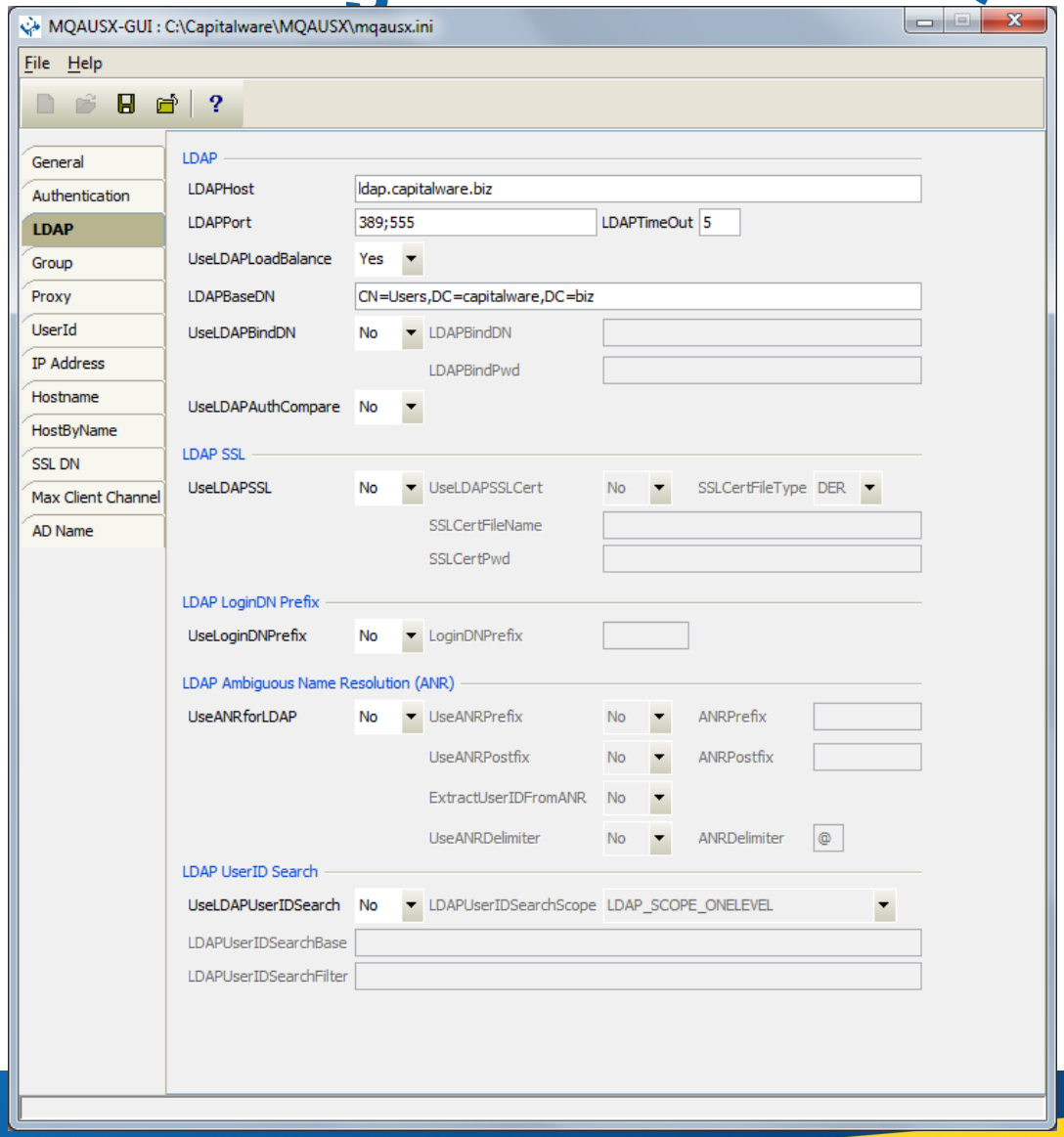
MQAUSX Configuration via MQAUSX-GUI



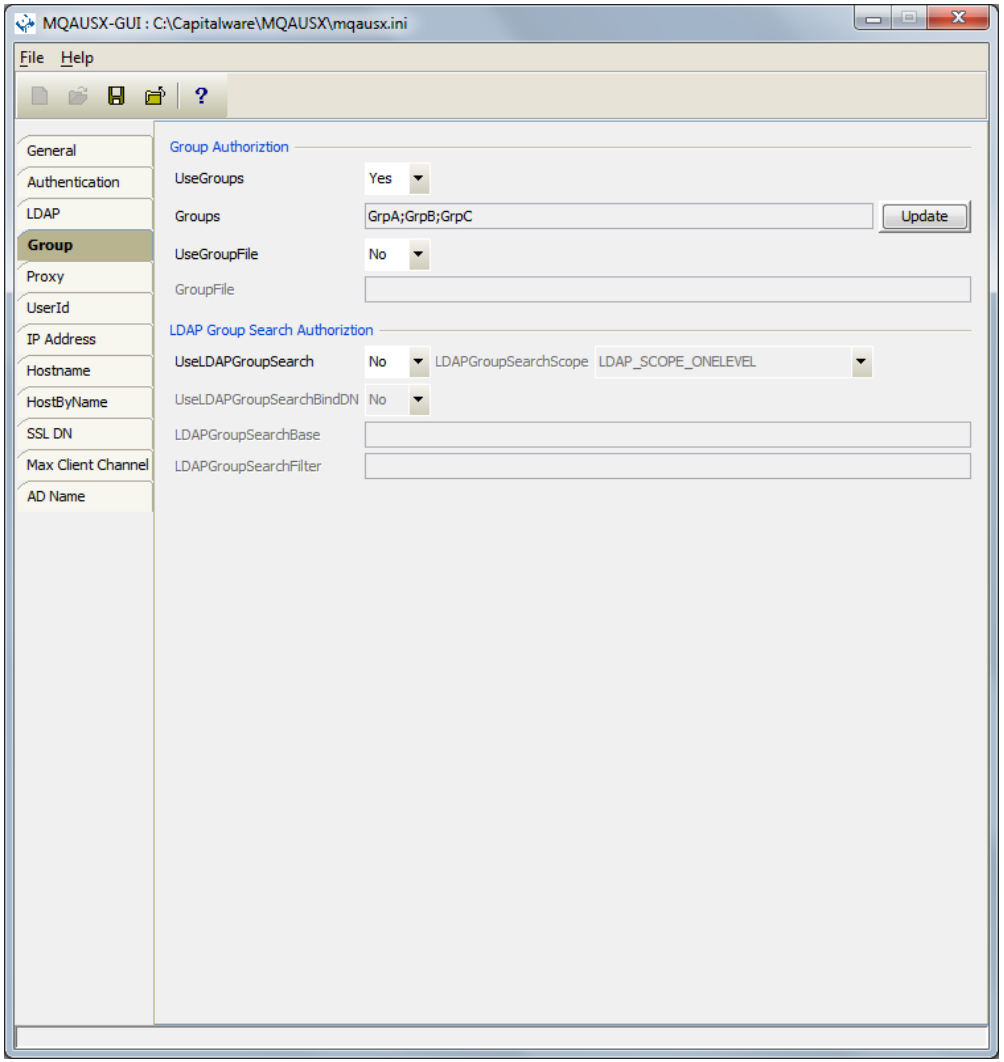
MQAUSX Configuration via MQAUSX-GUI



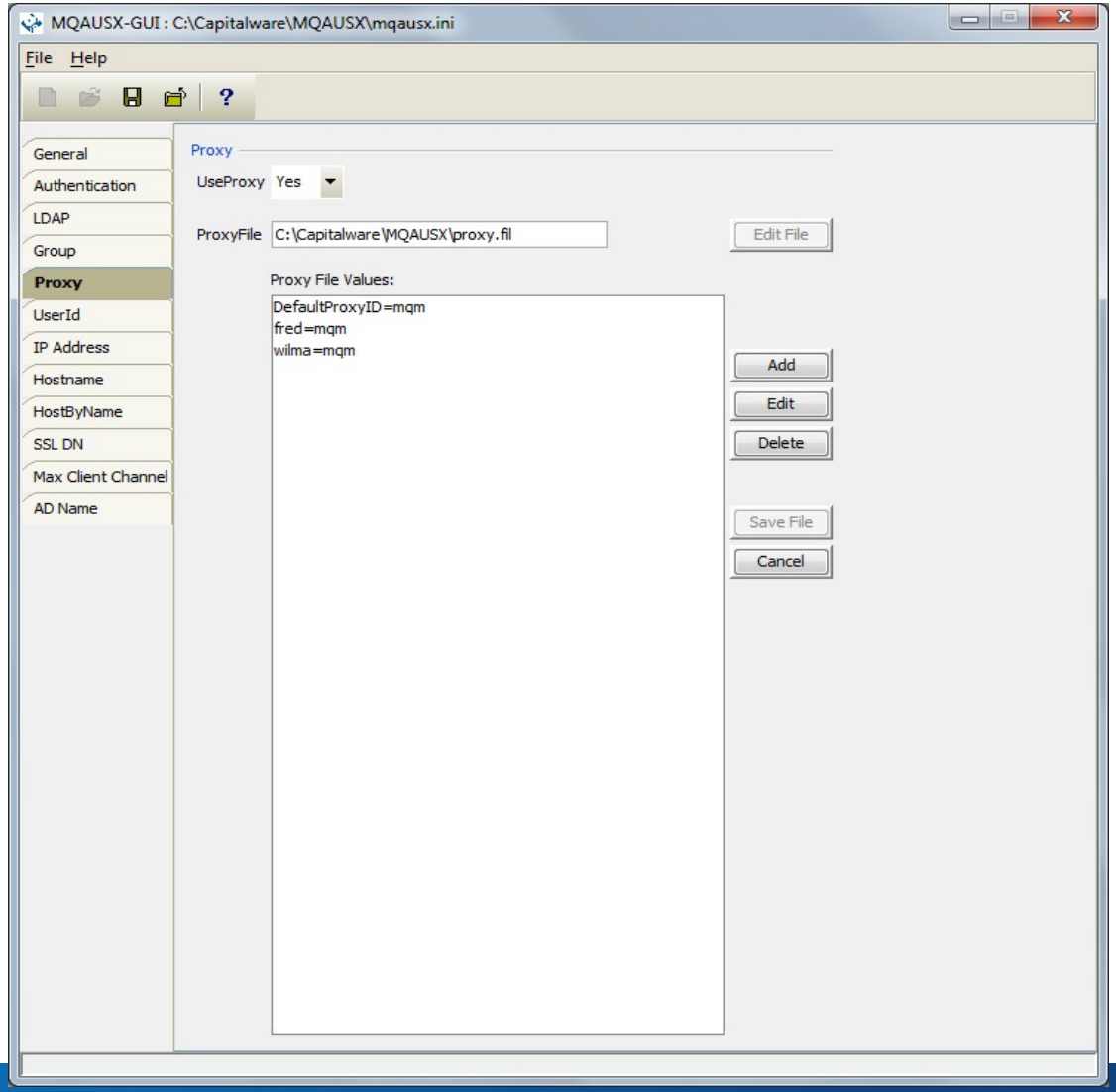
MQAUSX Configuration via MQAUSX-GUI



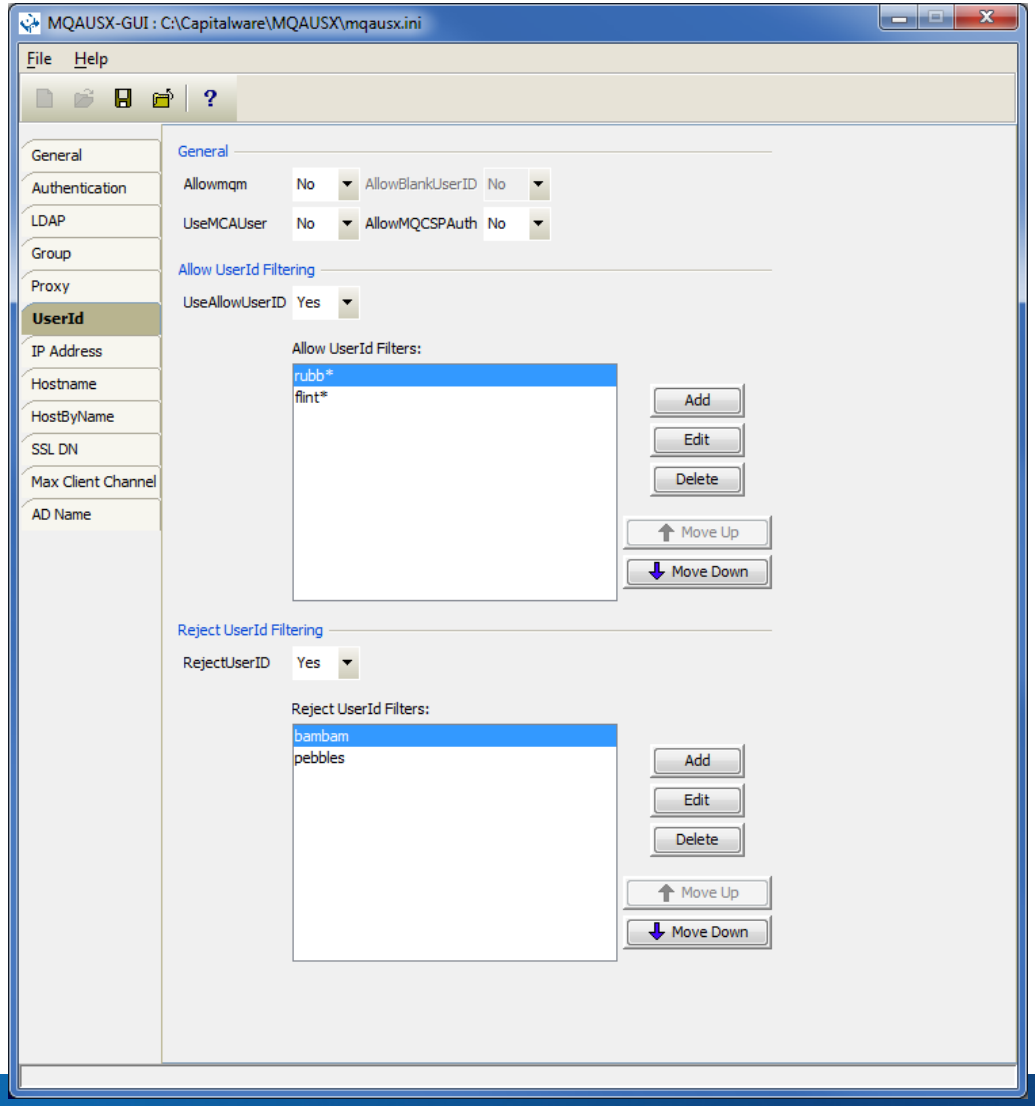
MQAUSX Configuration via MQAUSX-GUI



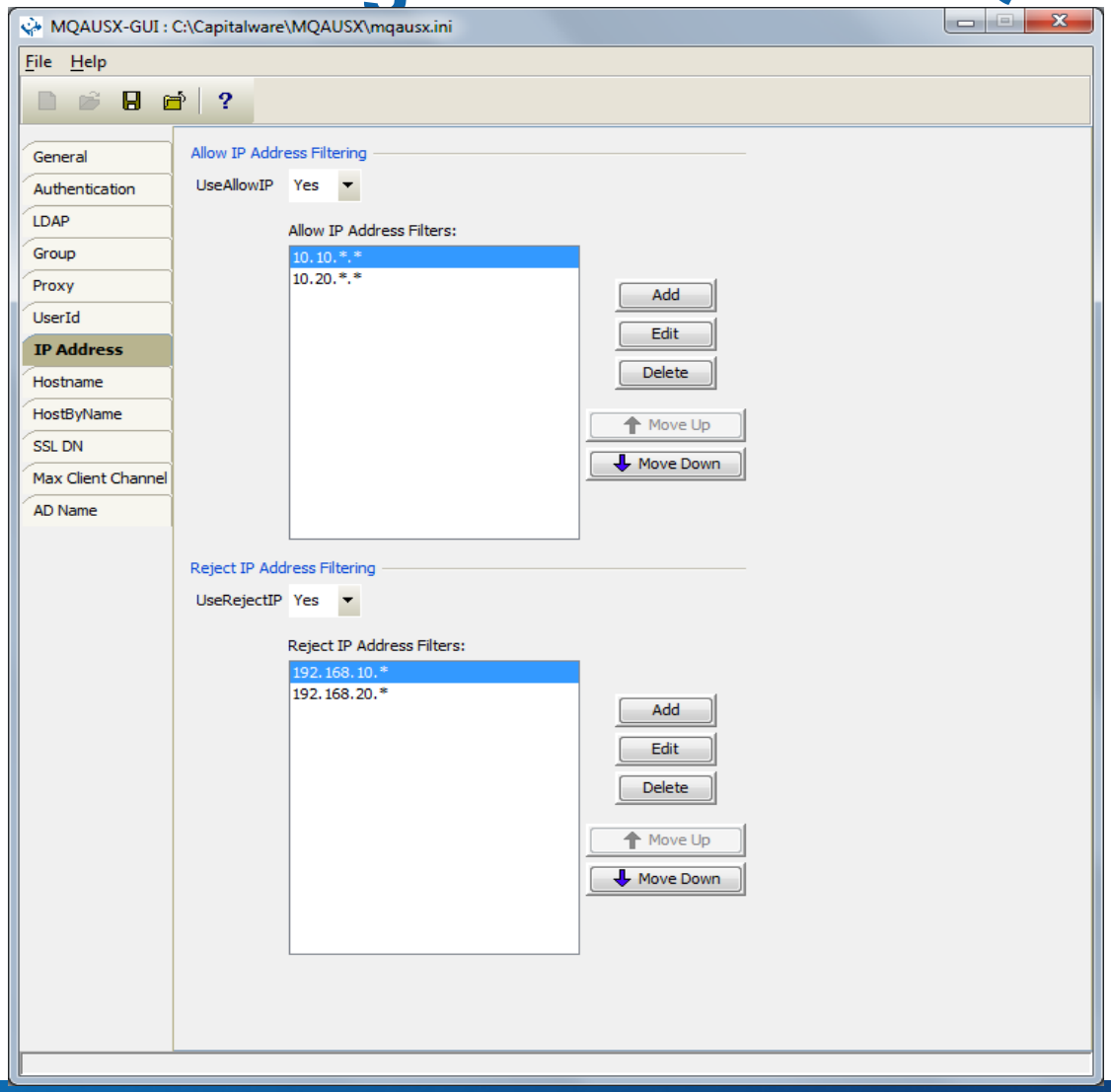
MQAUSX Configuration via MQAUSX-GUI



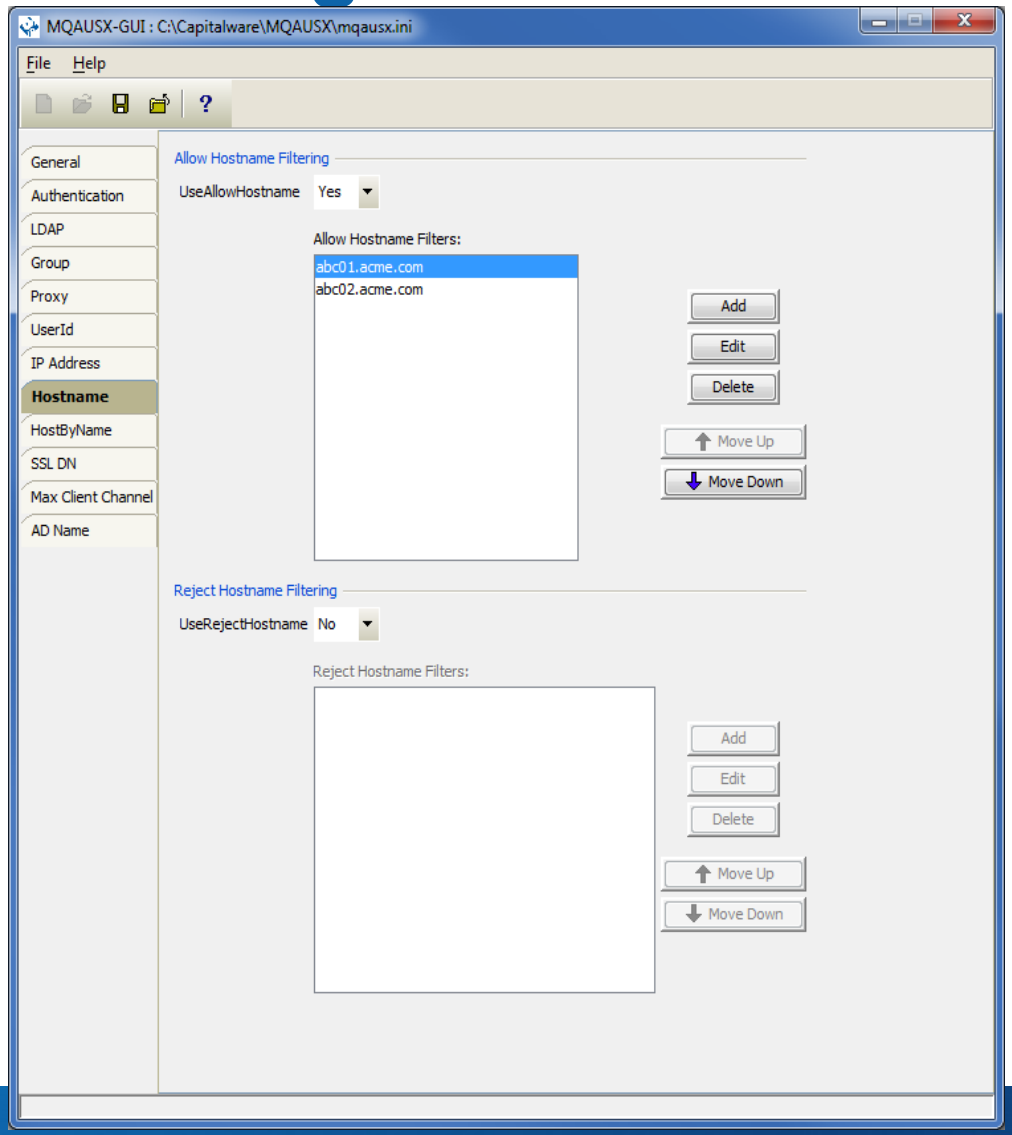
MQAUSX Configuration via MQAUSX-GUI



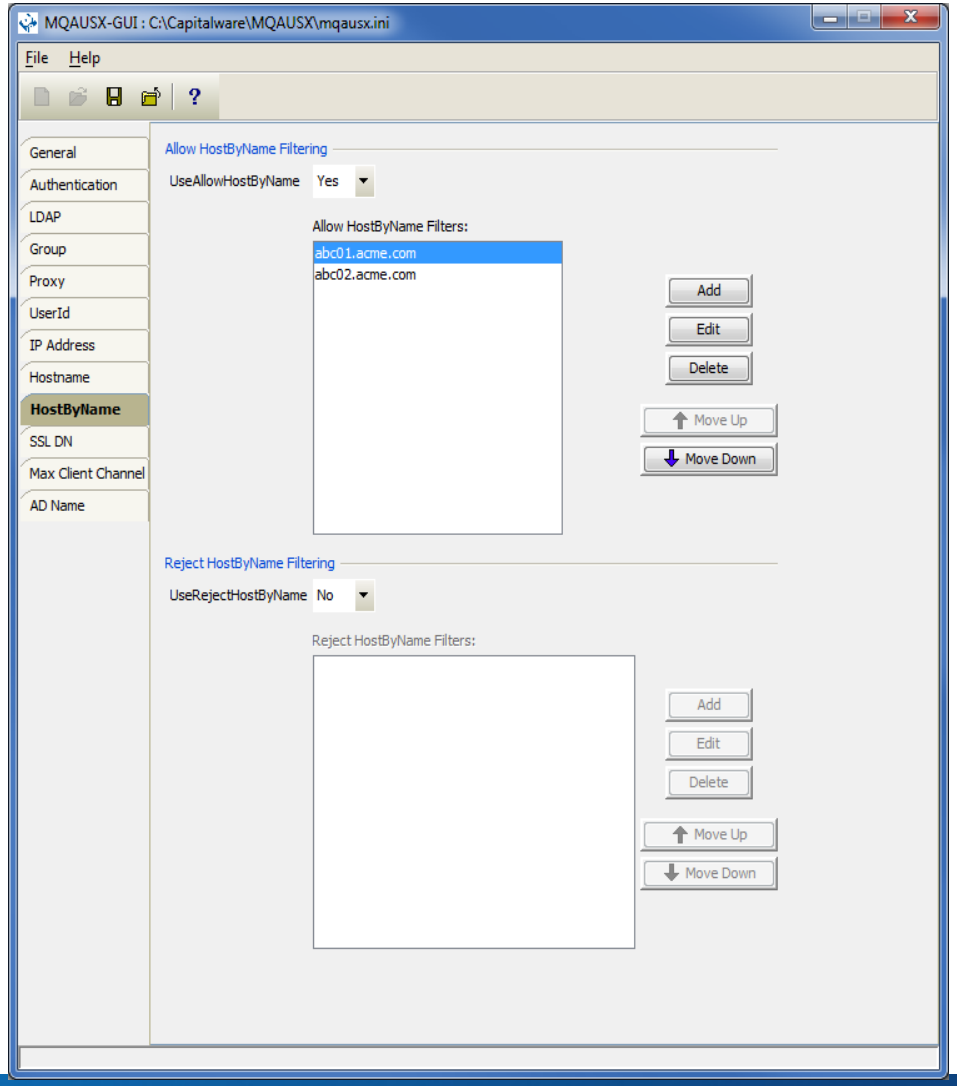
MQAUSX Configuration via MQAUSX-GUI



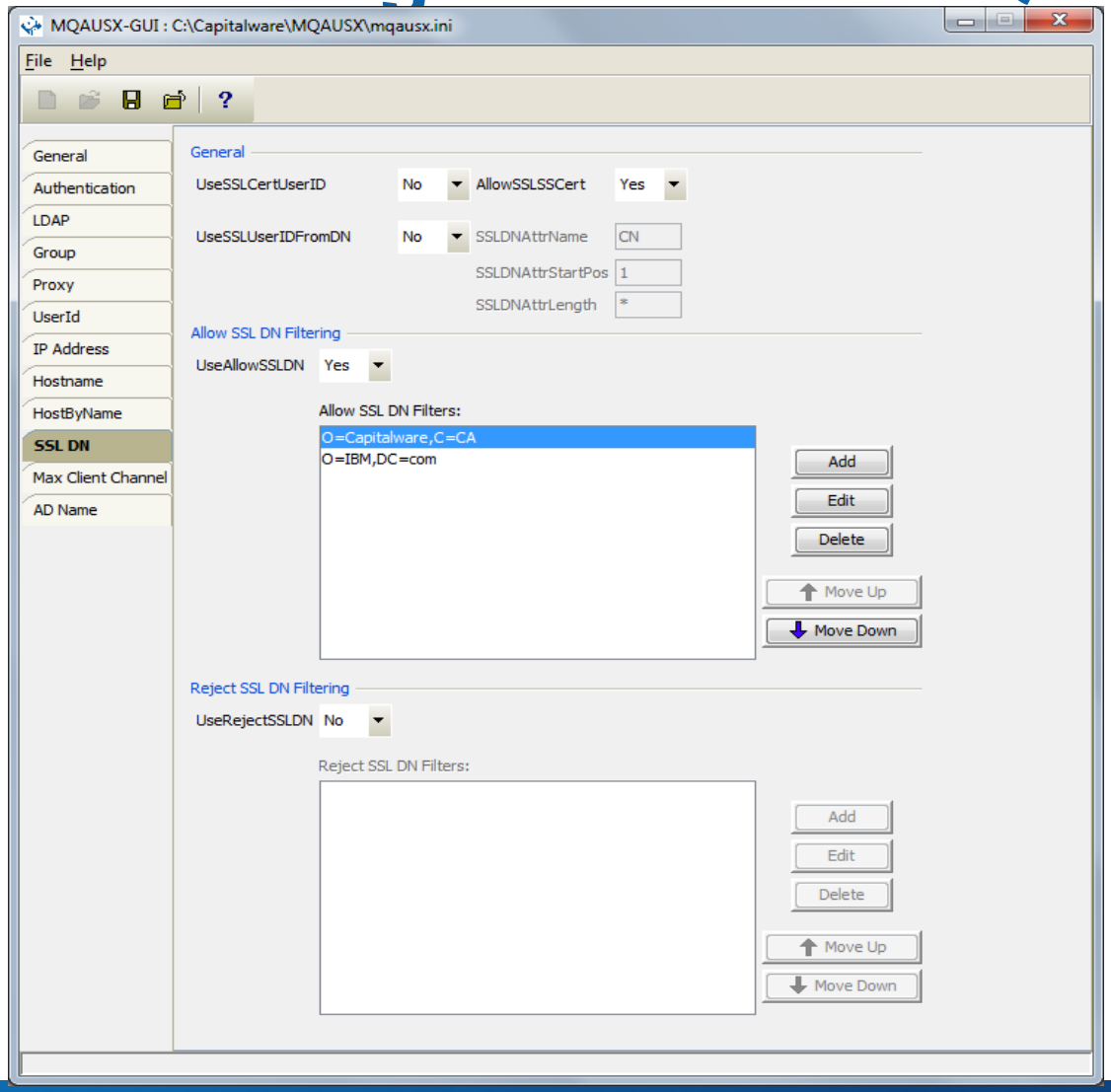
MQAUSX Configuration via MQAUSX-GUI



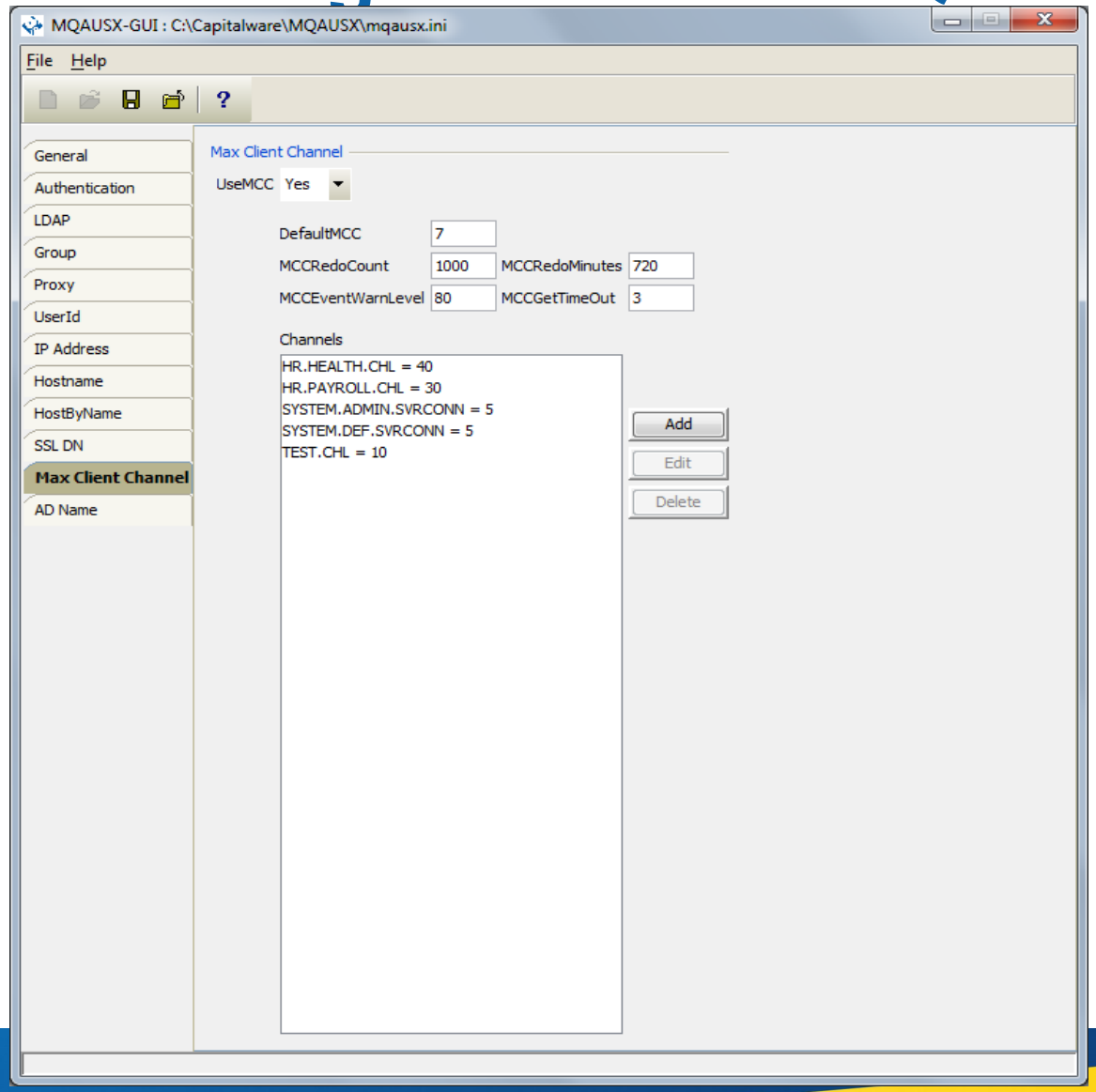
MQAUSX Configuration via MQAUSX-GUI



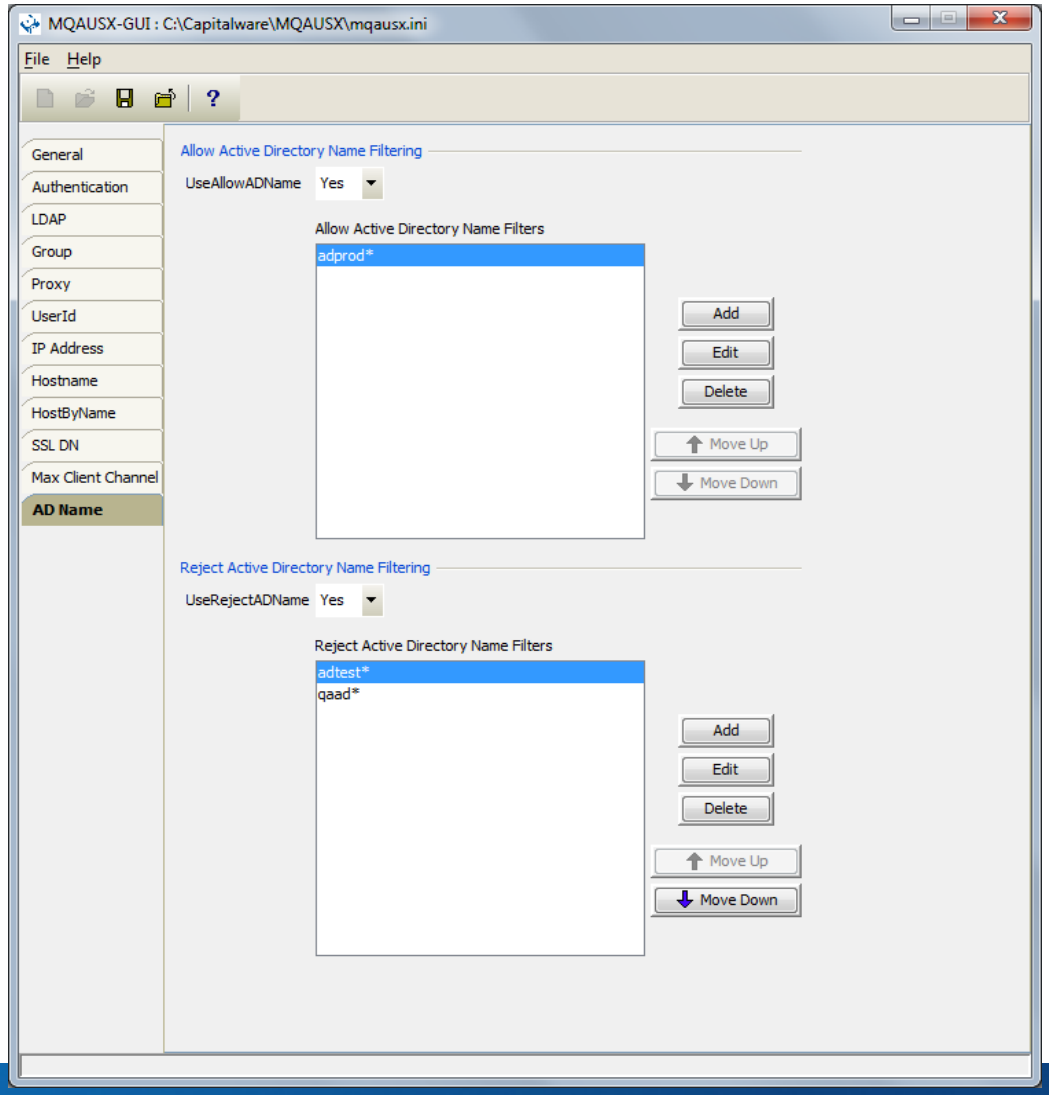
MQAUSX Configuration via MQAUSX-GUI



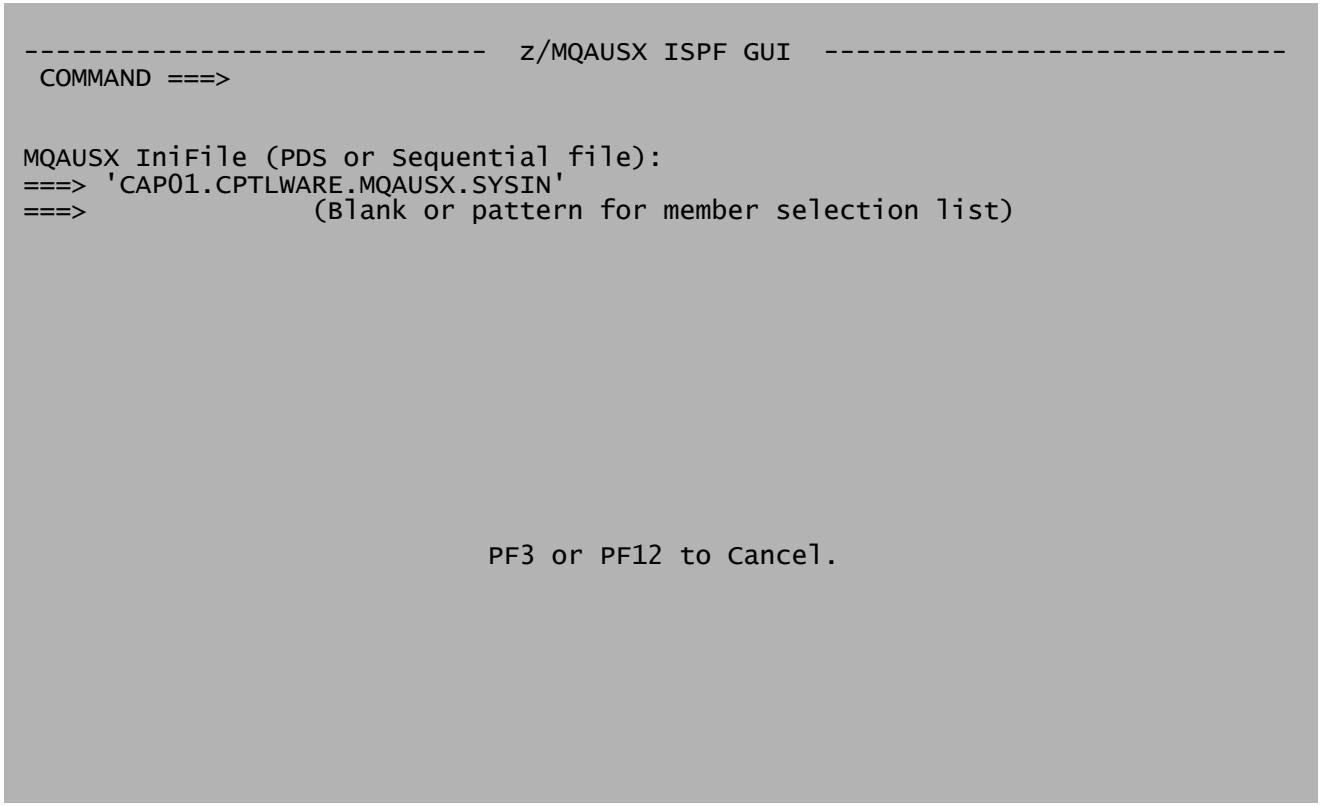
MQAUSX Configuration via MQAUSX-GUI



MQAUSX Configuration via MQAUSX-GUI



Configuration via MQAUSX-ISPF-GUI



Configuration via MQAUSX-ISPF-GUI

```
--------------------  z/MQAUSX ISPF GUI - General Setting  --------------------
COMMAND ===>

License Key ===>
LicenseFile ===>
Description ===>

Log Mode           ===> N        (Q/N/V/D)
Log File DD        ===> SYSPRINT
WriteToSystemLog   ===> N        (Y/N)
SystemLogMessage   ===> B        (B/A/R)
WriteToEventQueue  ===> N        (Y/N)
EventQueueName     ===> SYSTEM.ADMIN.CHANNEL.EVENT

Excessive Client Connections:
UseECC             ===> N        (Y/N)  ECCInterval  ===> D  (D/H/M)
ECCWarnCount       ===> 5000___

Sequence Number  ===> N         (Y/N)

                     PF3 to Return or PF12 to Cancel.
```

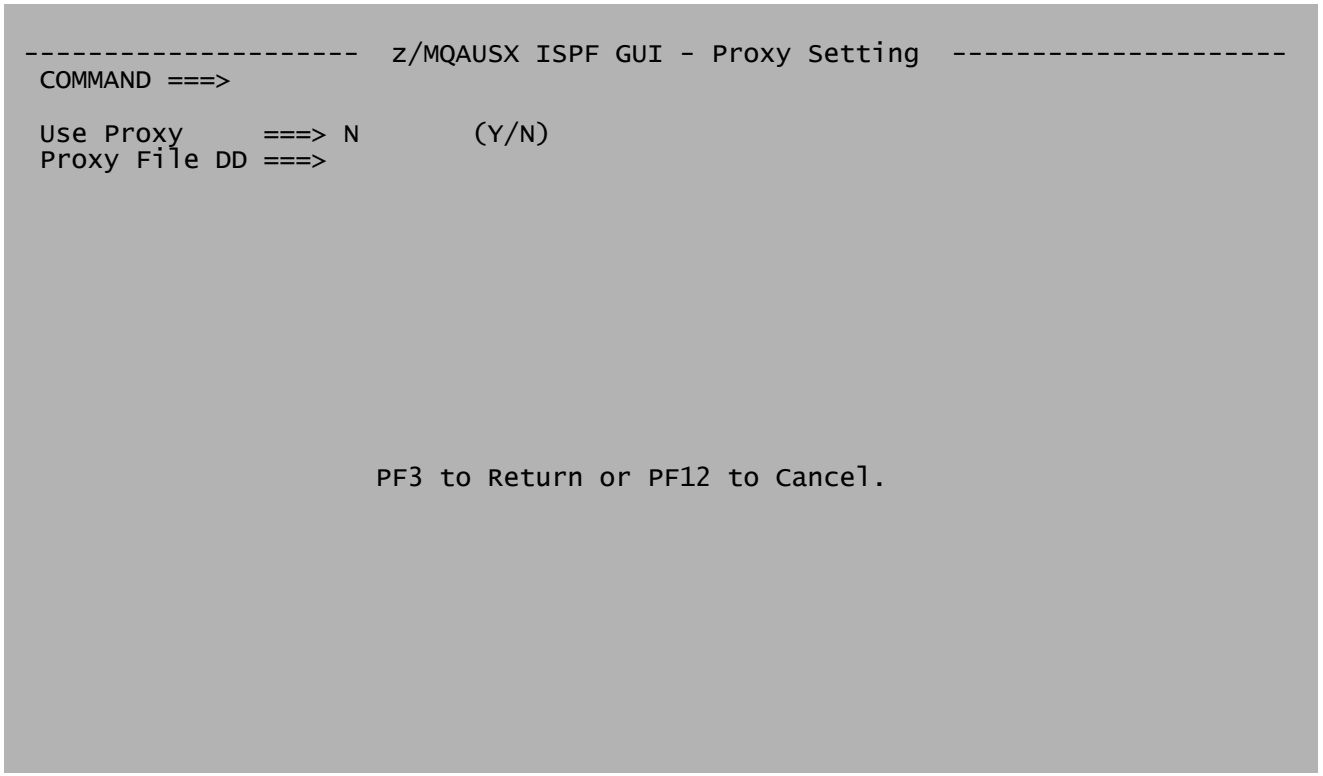
Configuration via MQAUSX-ISPF-GUI

```
-----------------  z/MQAUSX ISPF GUI - Authentication Setting  ----------------
 COMMAND ===>

 NoAuth ===> N (Y/N)

 File Based Access:
 UseFBA ===> N (Y/N)       FBAFile ===>

 Authentication Order:
 UseAuthOrder ===> N       AuthOrder ===>

 Queue Manager Password:
 UseQMgrPwd ===> N
 QMgrPwd    ===>

 Credentials:
 AllowPlainTextCredentials ===> Y  (Y/N)
```

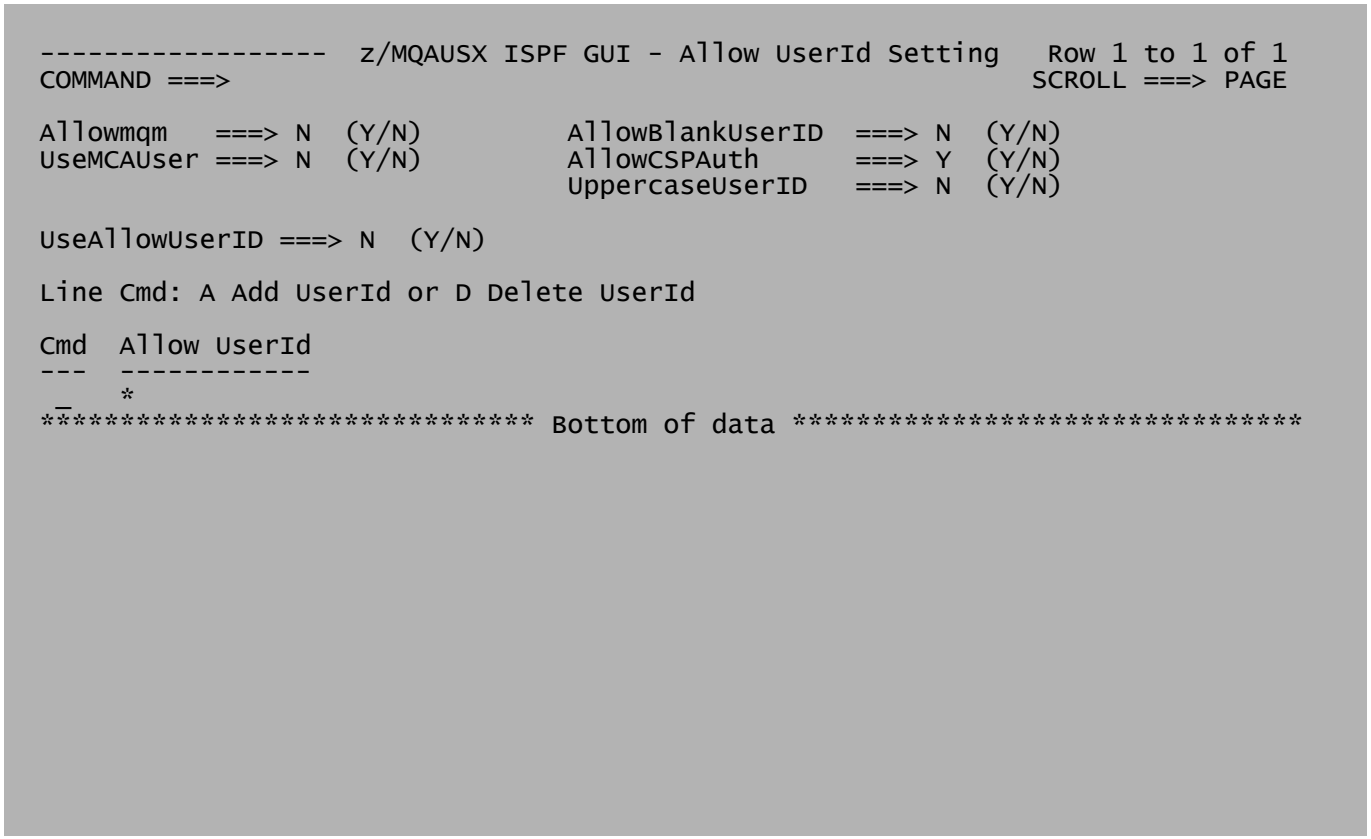
Configuration via MQAUSX-ISPF-GUI

```
----------------------  z/MQAUSX ISPF GUI - Group Setting  ----------------------
COMMAND ===>

Use Groups    ===> Y       (Y/N)
Groups        ===> GrpA,GrpB,GrpC
Group File DD ===> 'CAP01.CPTLWARE.MQAUSX.SYSIN(GRPIN)'

                    PF3 to Return or PF12 to Cancel.
```

Configuration via MQAUSX-ISPF-GUI



Configuration via MQAUSX-ISPF-GUI



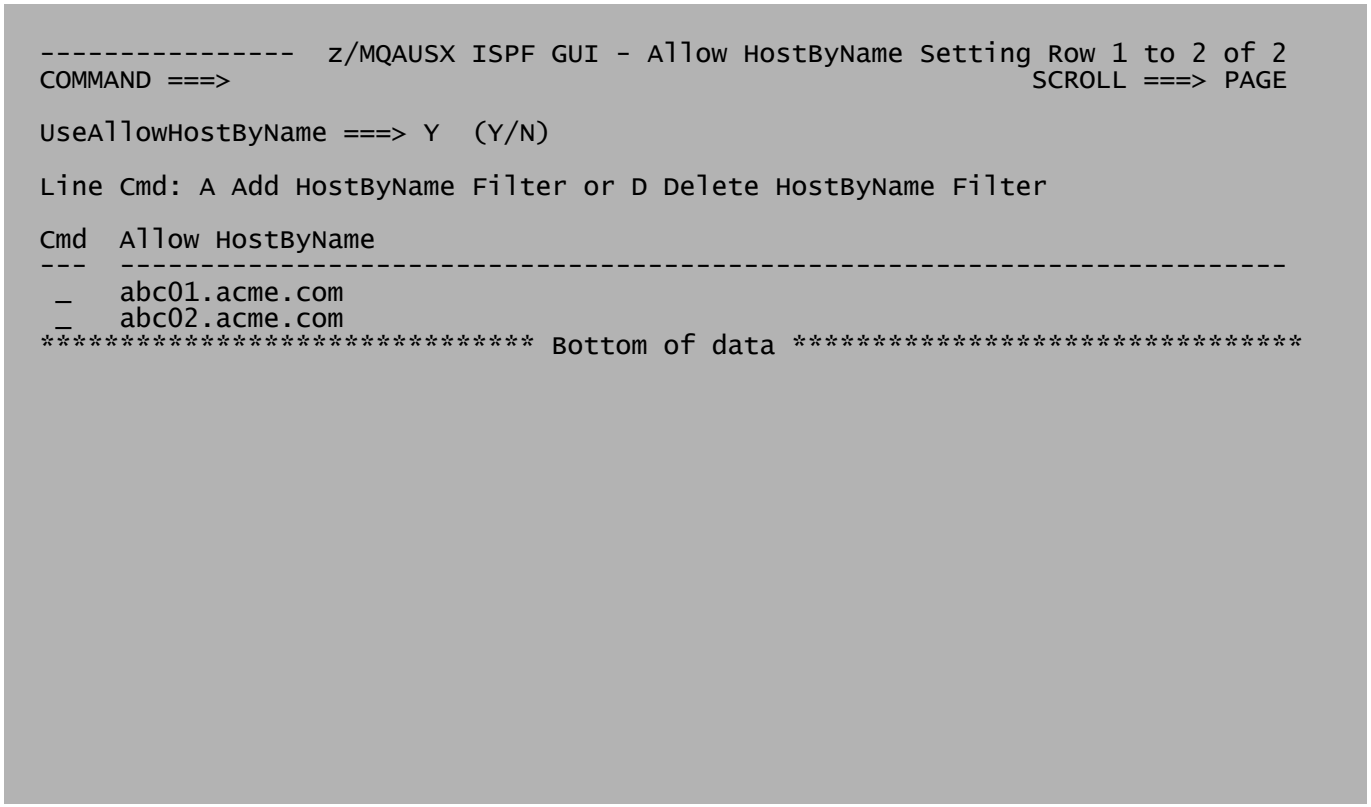
Configuration via MQAUSX-ISPF-GUI

```
----------------  z/MQAUSX ISPF GUI - Allow IP Address Setting Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> PAGE

UseAllowIP ===> Y  (Y/N)

Line Cmd: A Add IP Filter or D Delete IP Filter

Cmd  Allow IP Address
---  ------------------------------------------------------------------------
 _   192.168.10.*
 _   192.168.200.*
 _   10.10.*.*
****************************** Bottom of data ********************************
```

Configuration via MQAUSX-ISPF-GUI

```
----------------  z/MQAUSX ISPF GUI - Allow Hostname Setting Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

UseAllowHostname ===> Y  (Y/N)

Line Cmd: A Add Hostname Filter or D Delete Hostname Filter

Cmd  Allow Hostname
---  ------------------------------------------------------------------------
 _   abc01.acme.com
 _   abc02.acme.com
****************************** Bottom of data ********************************
```

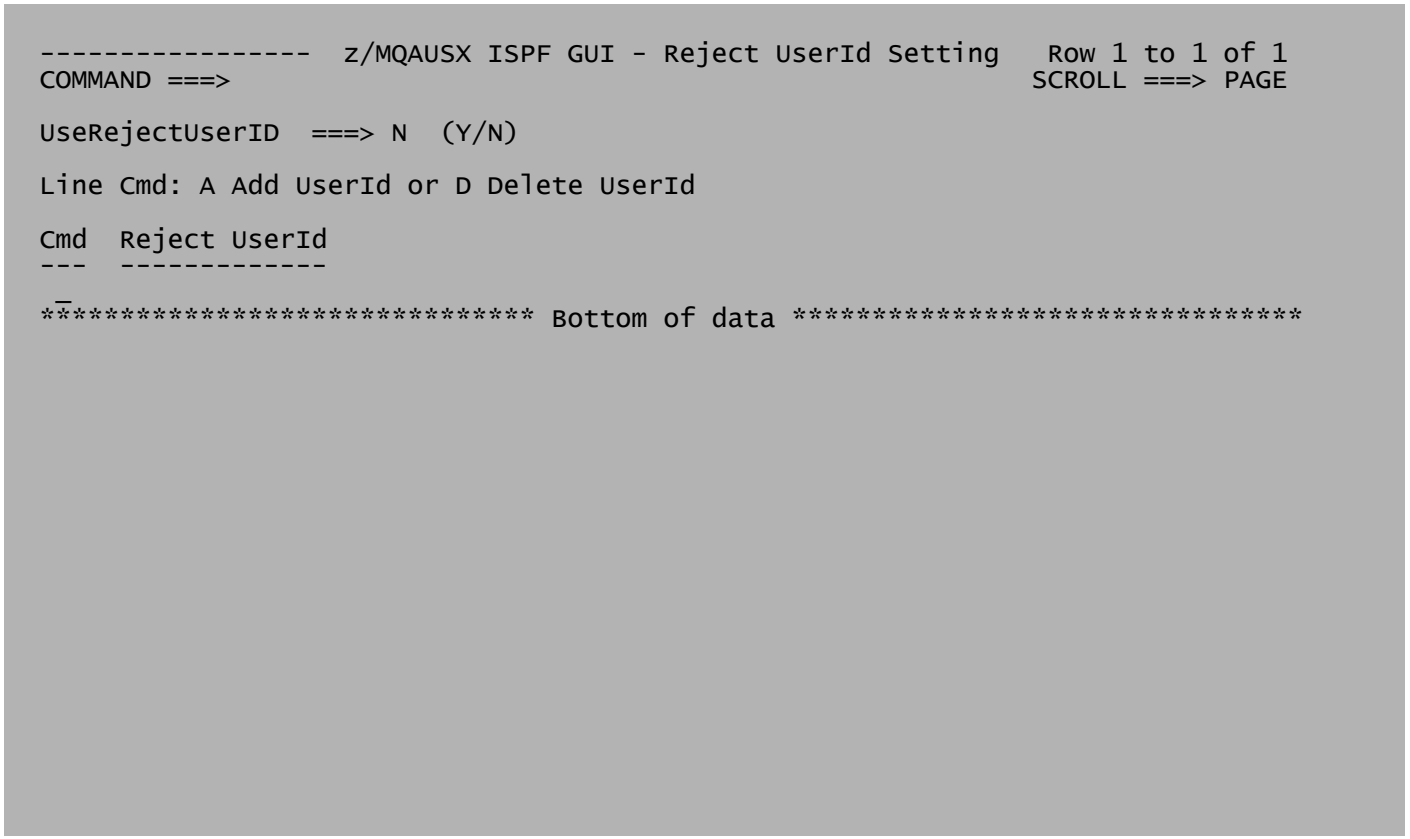
Configuration via MQAUSX-ISPF-GUI



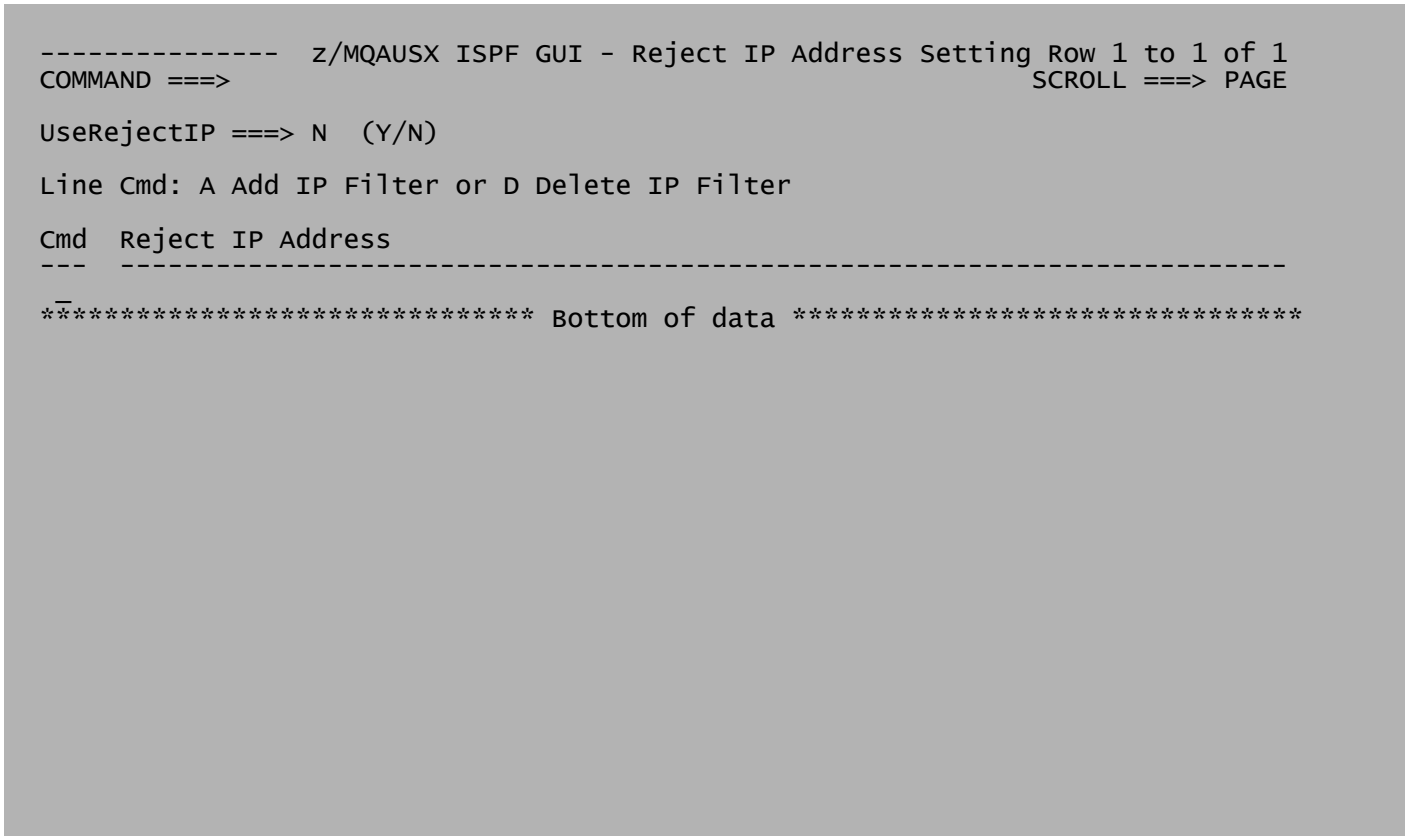
Configuration via MQAUSX-ISPF-GUI

```
----------------  z/MQAUSX ISPF GUI - Allow SSL DN Setting Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

UseSSLCertUserID   ===> N  (Y/N)    AllowSSLSSCert    ===> Y   (Y/N)
UseSSLUserIDFromDN ===> N  (Y/N)    SSLDNAttrName     ===> CN
                                    SSLDNAttrStartPos ===> 1
                                    SSLDNAttrLength   ===> *   (* for all)

UseAllowSSLDN ===> Y  (Y/N)

Line Cmd: A Add SSL DN Filter or D Delete SSL DN Filter

Cmd  Allow SSL DN
---  ------------------------------------------------------------------------
 _   O=Capitalware,C=CA
 _   O=IBM,DC=com
****************************** Bottom of data ********************************
```

Configuration via MQAUSX-ISPF-GUI



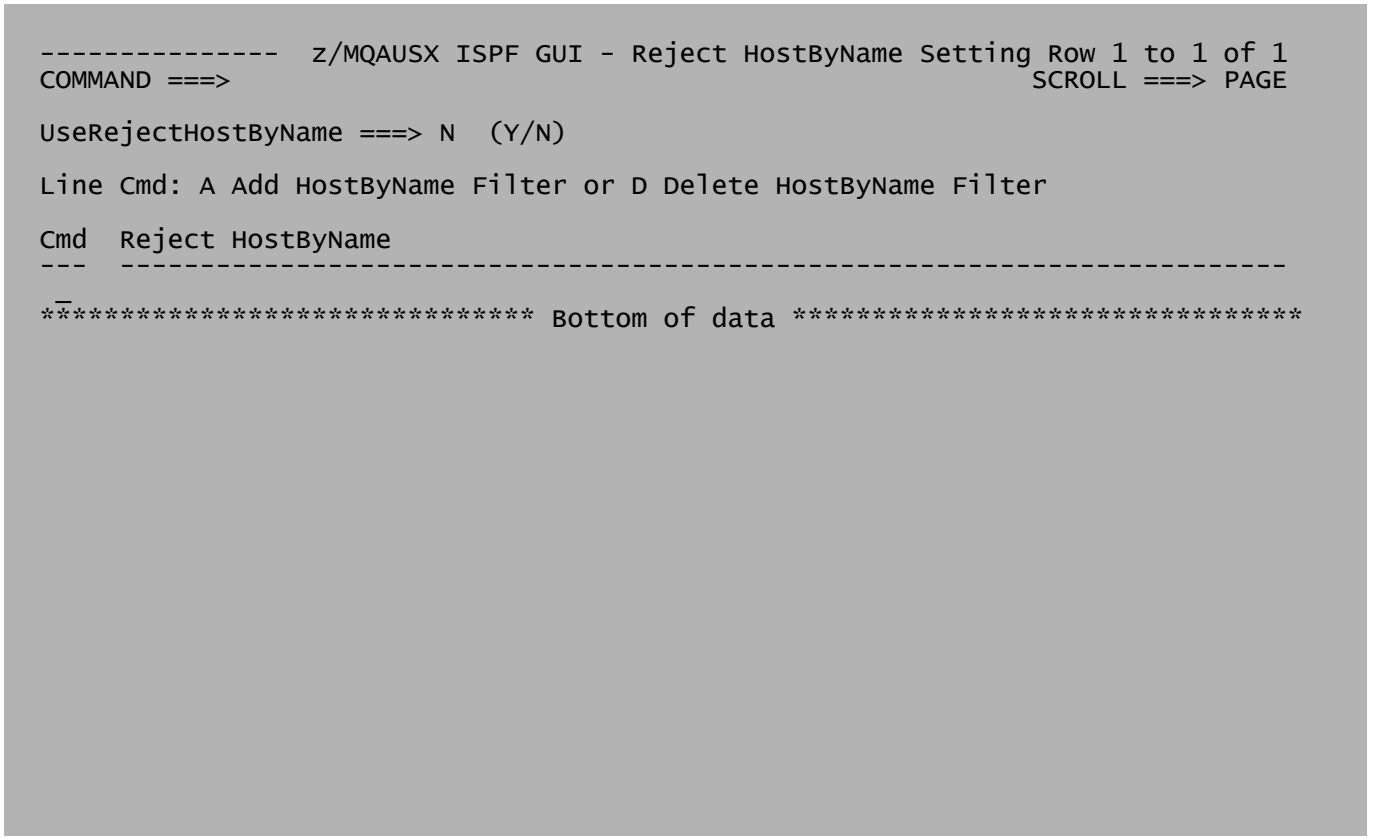
Configuration via MQAUSX-ISPF-GUI



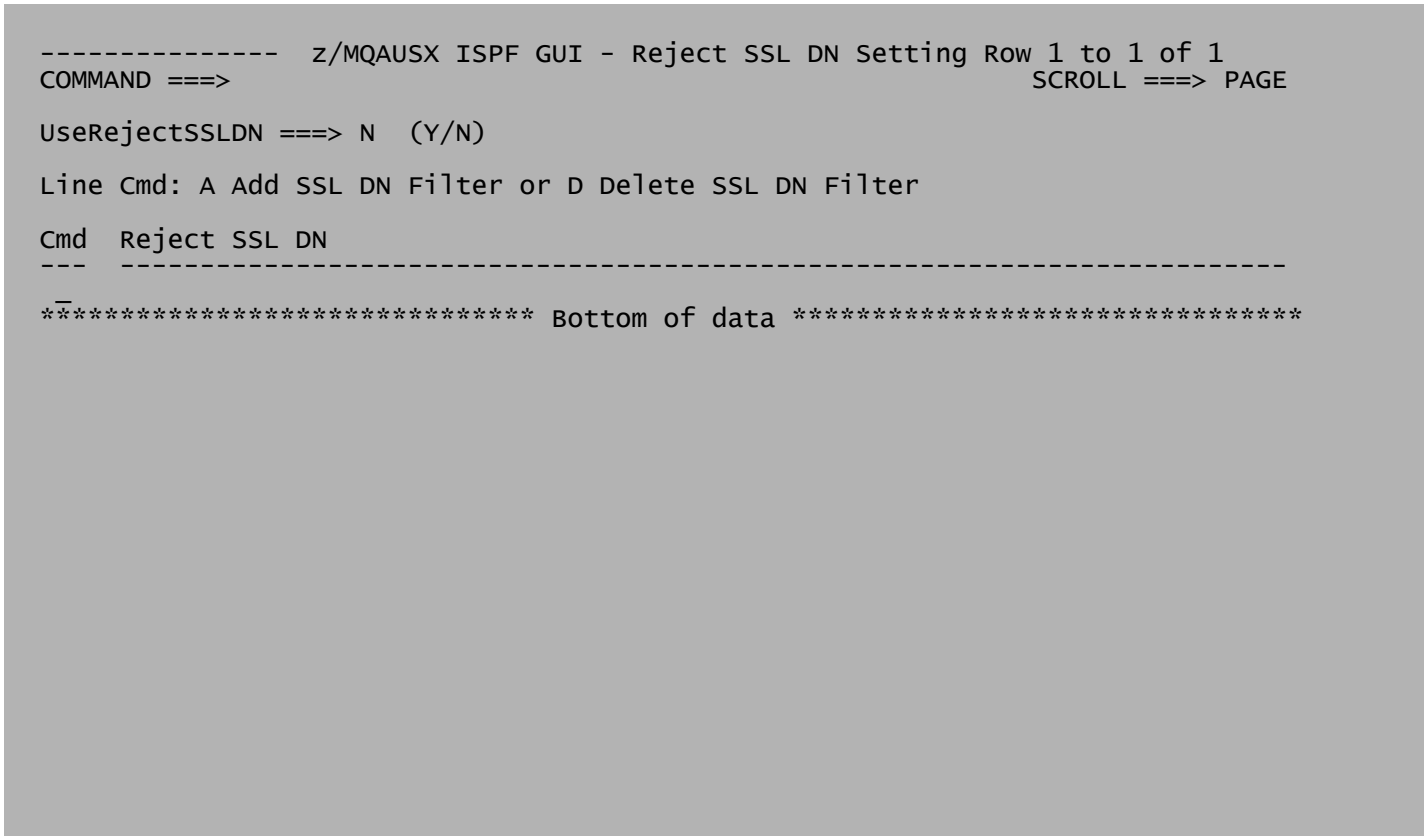
Configuration via MQAUSX-ISPF-GUI

```
---------------  z/MQAUSX ISPF GUI - Reject Hostname Setting Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE

UseRejectHostname ===> N  (Y/N)

Line Cmd: A Add Hostname Filter or D Delete Hostname Filter

Cmd  Reject Hostname
---  ------------------------------------------------------------------------
 _
****************************** Bottom of data ********************************
```

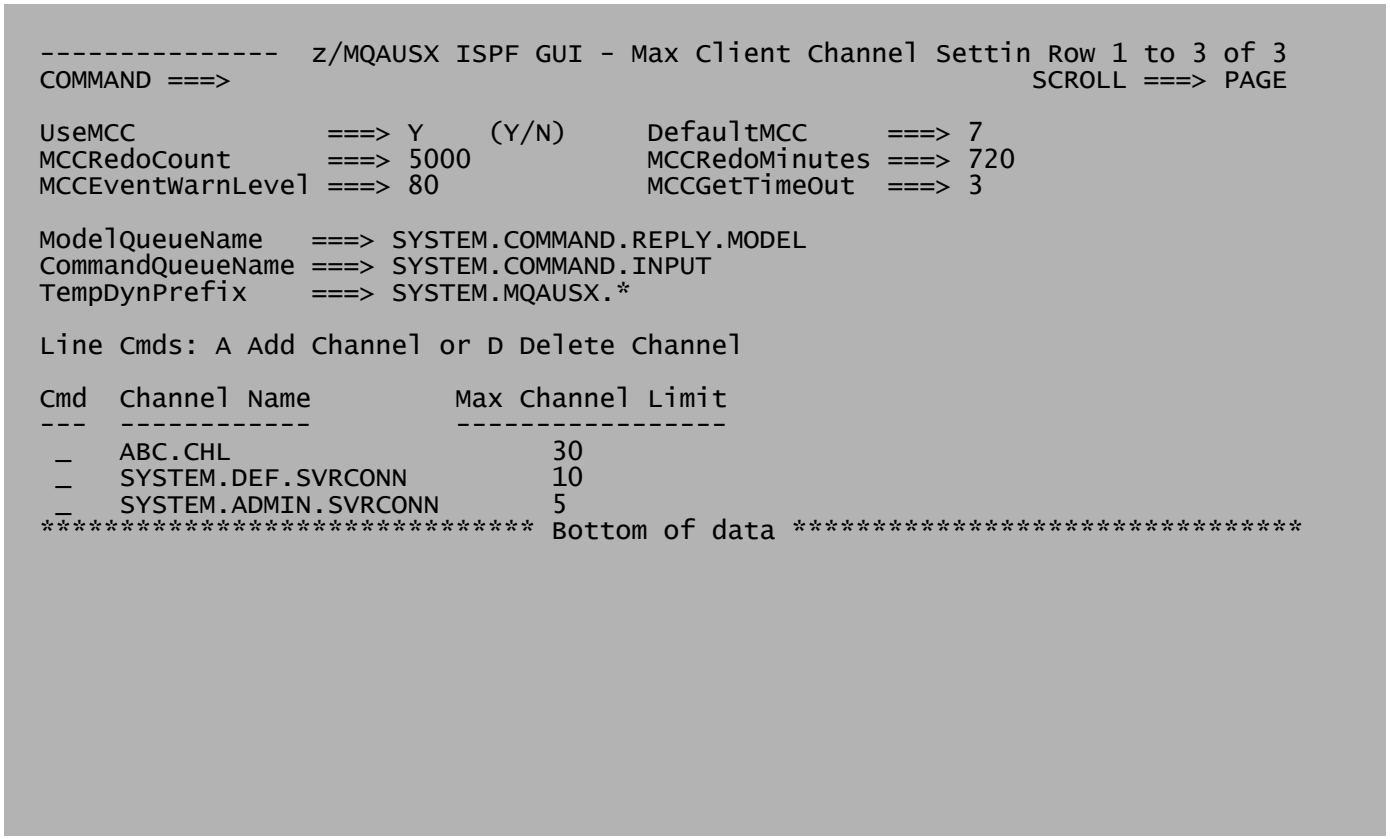
Configuration via MQAUSX-ISPF-GUI



Configuration via MQAUSX-ISPF-GUI



Configuration via MQAUSX-ISPF-GUI



Questions & Answers