

IBM MQ Connection Authentication

Morag Hughson

hughson@uk.ibm.com

Capitalware's MQ Technical Conference v2.0.1.4

Agenda

- **Requests for Enhancement**
- **Connection Authentication**
 - ▶ Configuration
 - ▶ Application Changes (or not)
 - ▶ Protecting your password across a network
 - ▶ User Repositories

Capitalware's MQ Technical Conference v2.0.1.4

Request for Enhancement (22568)



Headline: Password validation

ID: 22568

[Details](#) | [Comments](#) | [Attachments](#) | [Reconsideration](#) | [Release plans](#)

Status: [Uncommitted Candidate](#)

Visibility: Public

Description: Password validation of Client connections to be delivered for all platforms. CSQ4BCX3 is supplied for z/OS. We need the similar functionality for various platforms (Windows, Linux, AIX, Solaris, HP-NSK). This would help us to prove to audit that we know who is connecting.

Use case: Ease a secure integration with MO71 and MQ Explorer, so we can please law and audit teams. This will remove the need for using SSL to assure the identity of MQ administrators.

Bookmarkable URL: http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=22568
A unique URL that you can bookmark and share with others.

Capitalware's MQ Technical Conference v2.0.1.4

Request for Enhancement (30709)



Headline: WMQ Authentication via LDAP

ID: 30709

[Details](#) | [Comments](#) | [Attachments](#) | [Reconsideration](#) | [Release plans](#)

Status: [Uncommitted Candidate](#)

Visibility: Public

Description: Authenticate client connections with a central LDAP server. Instead of using the O/S for authentication we would like to be able to hand off a user/password combination to an LDAP server for authentication.

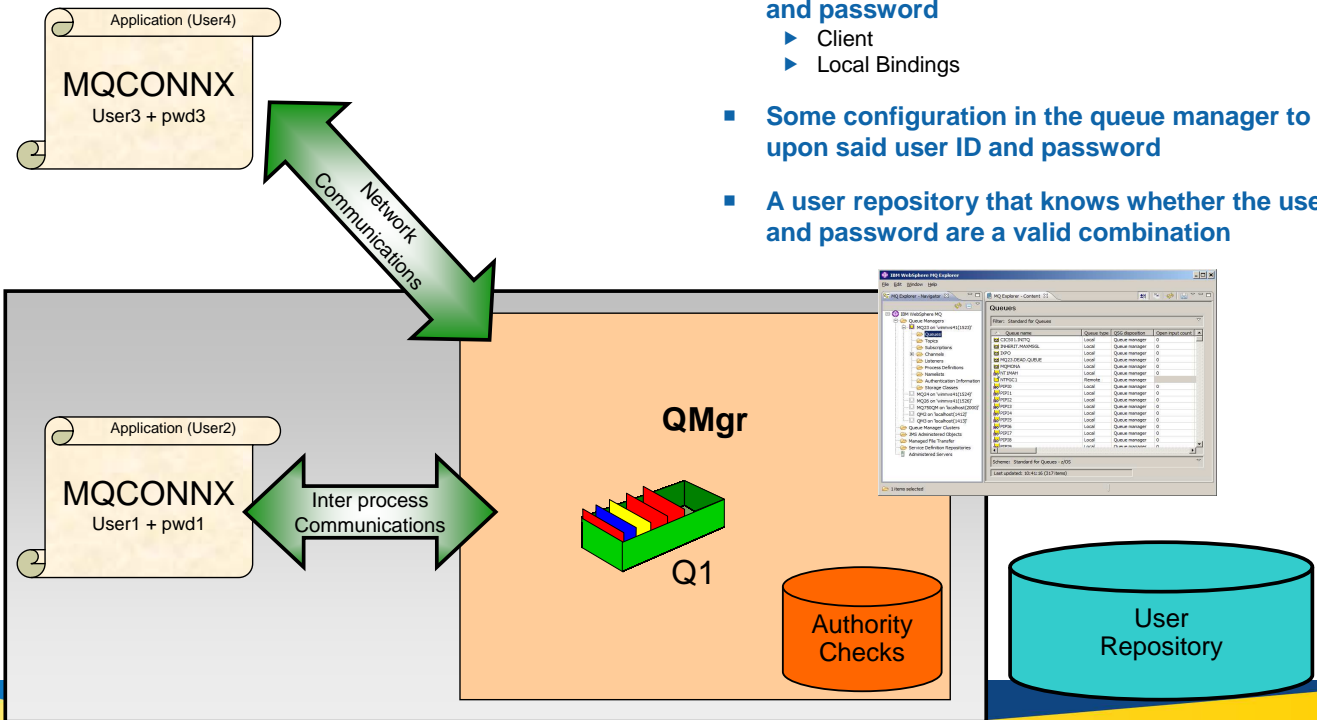
Use case: Clients would supply a user/password for authentication that would be validated by a central LDAP server, authorisation could be handled in the existing manner. The LDAP authentication could occur over SSL or plain TCP.

Bookmarkable URL: http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=30709
A unique URL that you can bookmark and share with others.

Capitalware's MQ Technical Conference v2.0.1.4

Connection Authentication – What is it?

- The ability for an application to provide a user ID and password
 - ▶ Client
 - ▶ Local Bindings
- Some configuration in the queue manager to act upon said user ID and password
- A user repository that knows whether the user ID and password are a valid combination



Capitalware's MQ Technical Conference v2.0.1.4

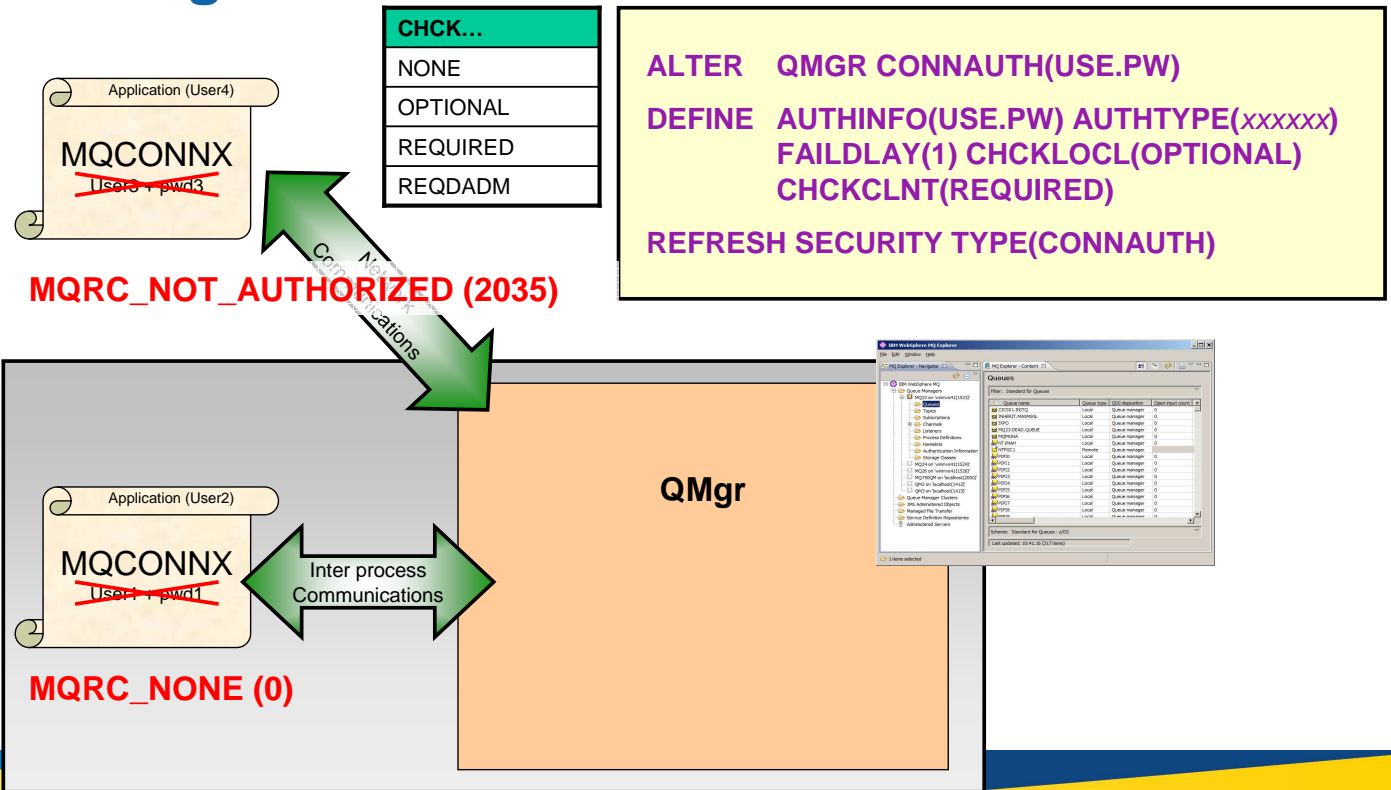
Connection Authentication – What is it? – Notes

N
O
T
E
S

- This picture shows the landscape we're going to use to discuss various patterns and then the changes in WebSphere MQ V8 in order to support these patterns. Just to ensure everyone is familiar with the parts on the diagram we'll briefly look at them first from left to right.
- On the left of this picture we see applications making connections, one as a client and one using local bindings. These applications could be using a variety of different APIs to connect to the queue manager, but all have the ability to provide a user ID and a password. The user ID that the application is running under (the classic user ID presented to WebSphere MQ) may be different from the user ID provided by the application along with its password, so we illustrate both on the diagram.
- In the middle we have a queue manager with configuration commands and managing the opening of resources and the checking of authority to those resources. There are lots of different resources in WebSphere MQ that an application may require authority to, in this diagram we are just going to use the example of opening a queue for output, but the same applies to all others.
- On the right we have a representation of a user repository – i.e. containing user IDs and passwords, more on this later.

Capitalware's MQ Technical Conference v2.0.1.4

Configuration



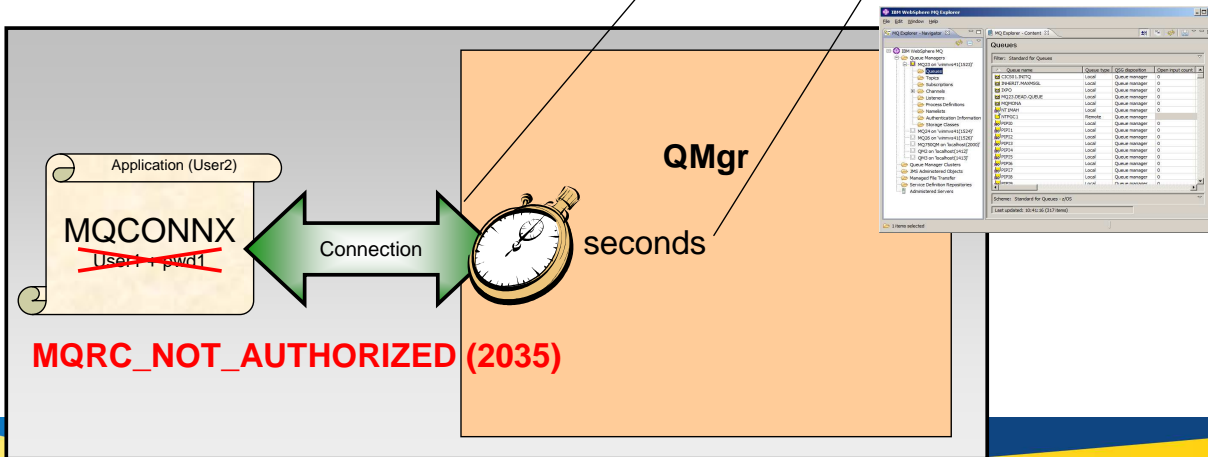
Configuration – Notes

N
O
T
E
S

- We'll start with the basic configuration side of things. How do I turn on this connection authentication feature on the queue manager.
- On the queue manager object there is a new attribute called CONNAUTH (short for connection authentication) which points to an object name. The object name it refers to is an authentication information object – one of two new types. There are two existing types of authentication information objects from earlier releases of WebSphere MQ, these original two types cannot be used in the CONNAUTH field.
- The two new types are similar in quite a few of the basic attributes so we will look at those first. We'll come back to more of the attributes later. We show here a new authentication information object which has two fields to turn on user ID and password checking, CHCKLOCL (Check Local connections) and CHCKCLNT (Check Client connections). Changes to the configuration of this must be refreshed for the queue manager to pick them up.
- Both of these fields have the same set of attributes, allowing for a strictness of checking. You can switch it off entirely with NONE; set it to OPTIONAL to ensure that if a user ID and password are provided by an application then they must be a valid pair, but that it is not mandatory to provide them – a useful migration setting perhaps; set it to REQUIRED to mandate that all applications provide a user ID and password; and, only on Distributed, REQDADM which says that privileged users must supply a valid user ID and password, but non-privileged users are treated as per the OPTIONAL setting.
- Any application that does not supply a user ID and password when required to, or supplies an incorrect combination even when it is optional will be told 2035 (MQRC_NOT_AUTHORIZED). N.B. When password checking is turned off using NONE – then invalid passwords will not be detected.

Connection Failure Delay

```
ALTER QMGR CONNAUTH(USE.PW)
DEFINE AUTHINFO(USE.PW) AUTHTYPE(XXXXXX)
FAILDLAY(1) CHCKLOCL(OPTIONAL)
CHCKCLNT(REQUIRED)
REFRESH SECURITY TYPE(CONNAUTH)
```



Capitalware's MQ Technical Conference v2.0.1.4

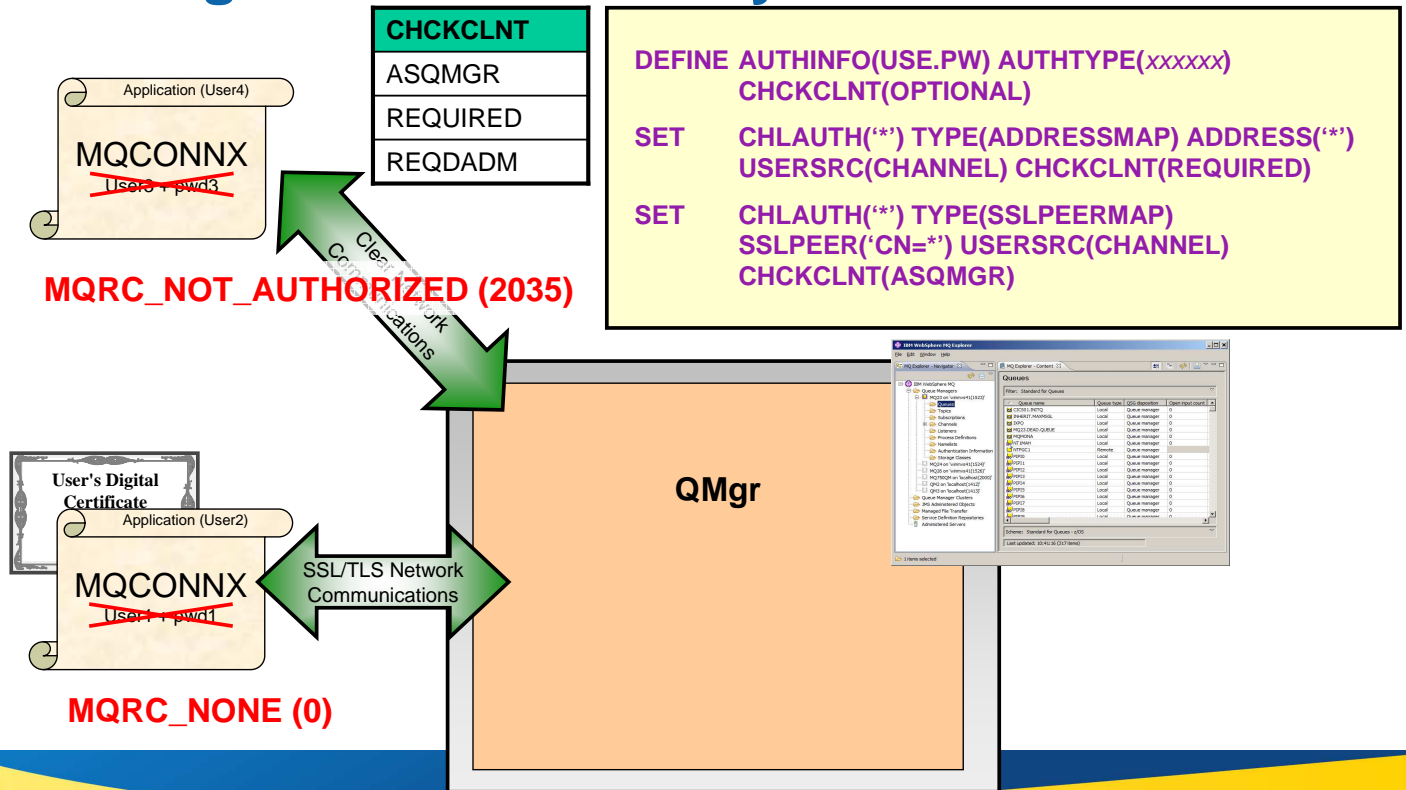
Connection Failure Delay - Notes

N
O
T
E
S

- Any failed authentications will be held for the number of seconds in the FAILDLAY attribute before the error is returned to the application – just some protection against a busy loop from an application repeatedly connecting.

Capitalware's MQ Technical Conference v2.0.1.4

Configuration Granularity

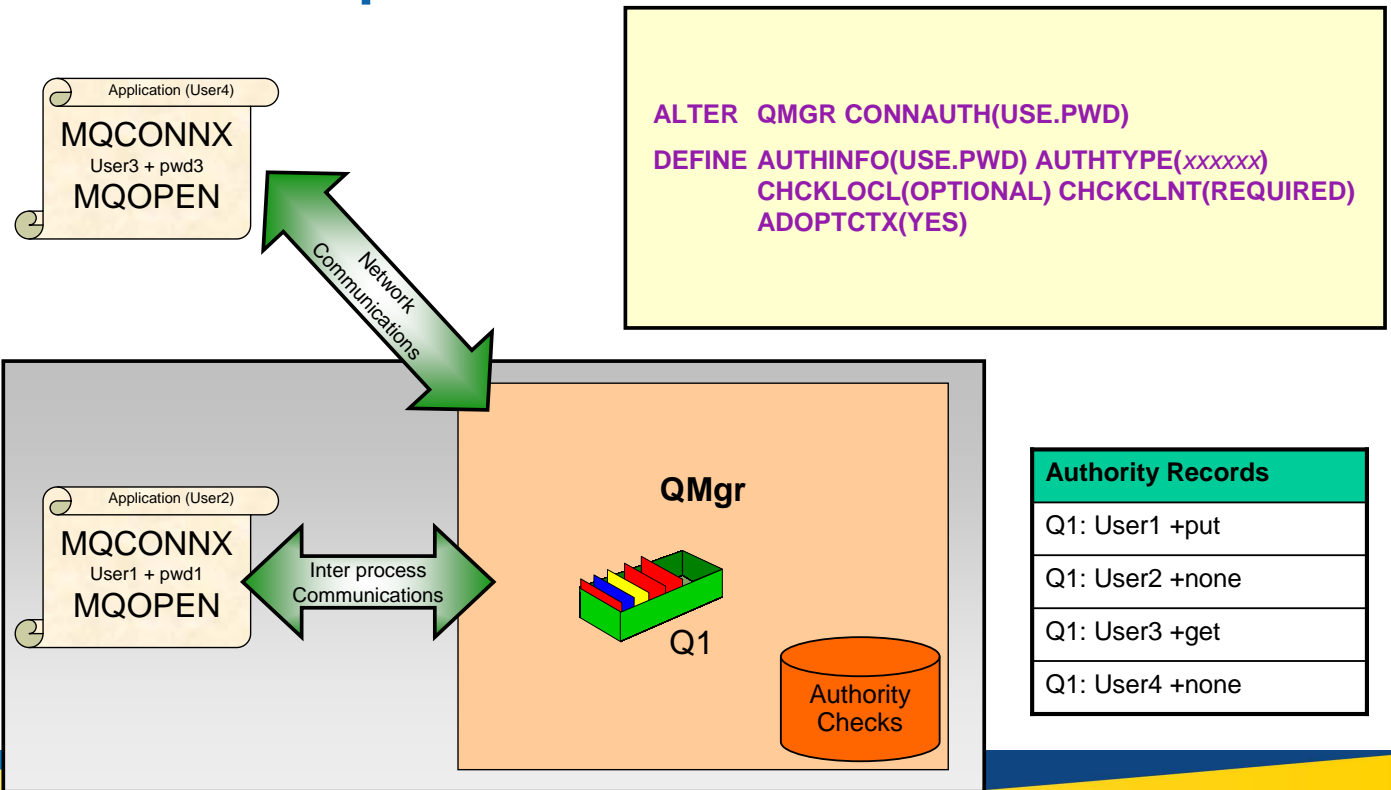


Configuration Granularity – Notes

N
O
T
E
S

- In addition to the two fields that turn this on overall for client and locally bound applications, there are enhancements to the CHLAUTH rules so that more specific configuration can be made using CHCKCLNT. You can set the overall CHCKCLNT value to OPTIONAL, and then upgrade it to be more stringent for certain channels by setting CHCKCLNT to REQUIRED or REQDADM on the CHLAUTH rule. By default, CHLAUTH rules will run with CHCKCLNT(ASQMGR) so this granularity does not have to be used.

Relationship to Authorization



Capitalware's MQ Technical Conference v2.0.1.4

Relationship to Authorization – Notes

N
O
T
E
S

- So we have seen that we can configure our queue manager to mandate user IDs and passwords are provided by certain applications. We know that the user ID that the application is running under may not be the same user ID that was presented by the application along with a password. So what is the relationship of these user IDs to the ones used for the authorization checks when the application, for example, opens a queue for output.
- There are two choices, in fact, controlled by an attribute on the authentication information object – ADOPTCTX.
- You can choose to have applications provide a user ID and password for the purposes of authenticating them at connection time, but then have them continue to use the user ID that they are running under for authorization checks. This may be a useful stepping stone when migrating, or even a desirable mode to run in, perhaps with client connections, because authorization checks are being done using an assigned MCAUSER based on IP address or SSL/TLS certificate information.
- Alternatively, you can choose the applications to have all subsequent authorization checks made under the user ID that you authenticated by password by selecting to adopt the context as the applications context for the rest of the life of the connection.
- If the user ID presented for authentication by password is the same user ID that the application is also running under, then of course this setting has no effect.

Capitalware's MQ Technical Conference v2.0.1.4

Adopting User - Interaction with CHLAUTH

Method	Notes
Client machine user ID flowed to server	This will be over-ridden by anything else. Rarely do you want to trust an unauthenticated client side user ID.
MCAUSER set on SVRCONN channel definition	A handy trick to ensure that the client flowed ID is never used is to define the MCAUSER as 'rubbish' and then anything that is not set appropriately by one of the next methods cannot connect.
MCAUSER set by ADOPTCTX(YES)	The queue manager wide setting to adopt the password authenticated user ID as the MCAUSER will over-ride either of the above.
MCAUSER set by CHLAUTH rule	To allow more granular control of MCAUSER setting, rather than relying on the above queue manager wide setting, you can of course use CHLAUTH rules
MCAUSER set by Security Exit	Although CHLAUTH gets the final say on whether a connection is blocked (security exit not called in that case), the security exit does get called with the MCAUSER CHLAUTH has decided upon, and can change it.

Capitalware's MQ Technical Conference v2.0.1.4

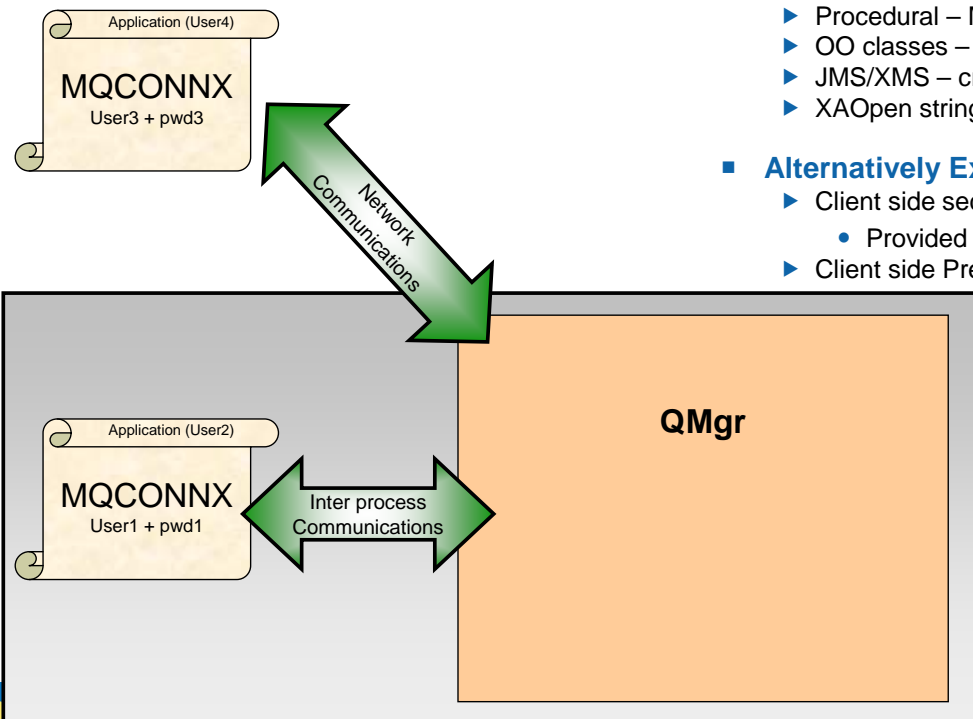
Adopting User - Interaction with CHLAUTH - Notes

N
O
T
E
S

- There are numerous ways that the running user can be set for a SVRCONN channel, i.e. the user which is representing the client application when it is running on the queue manager machine. The ADOPTCTX(YES|NO) attribute that we just saw is yet another one. How do all these different ways of setting the MCAUSER on the SVRCONN interact.
- There is an order of events and certain ways of setting the MCAUSER over-ride others. The table shows the order.

Capitalware's MQ Technical Conference v2.0.1.4

Application changes



- **Code changes**
 - ▶ Procedural – MQCSP on MQCONNX
 - ▶ OO classes – MQEnvironment
 - ▶ JMS/XMS – createConnection
 - ▶ XAOpen string
- **Alternatively Exits can provide MQCSP**
 - ▶ Client side security exit
 - Provided
 - ▶ Client side Pre-conn exit

Capitalware's MQ Technical Conference v2.0.1.4

Application changes – Notes

N
O
T
E
S

- Since WebSphere MQ V6.0, an application has been able to provide a user ID and password (in the Connection Security Parameters (MQCSP) structure in the MQCNO) at MQCONNX time. These were passed to a user written plug-point in the OAM on distributed to be checked. If the application was running client bound, this user ID and password were also passed to the client side and server side security exits for processing and can be used for setting the MCAUser attribute of a channel instance. The security exit is called with ExitReason MQXR_SEC_PARMS for this processing.
- This pre-existing feature of the MQI is being used to provide the user ID and password to the queue manager for checking. Previously a custom Authorization Service was required to check this (or a security exit if the applications were connecting as clients), now the Object Authority Manager (OAM) supplied with the queue manager and the z/OS Security component within the queue manager will deal with these user IDs and passwords. Whether z/OS or distributed, the component that deals with the user IDs and passwords will call out to a facility outside of MQ to do the check – more on that later.
- In WebSphere MQ V8 this will be available in all our interfaces listed, even where some of those were not made available in the WebSphere MQ V6 timeframe when the programming interface was originally provided.
- In prior releases the MQCSP had no architected limits on the user ID and password strings that were provided by the application. When using them with these MQ provided features there are limits which apply to the use of these features, but if you are only passing them to your own exits, those limits do not apply.
- The XAOpen string has also been updated to allow the provision of a user ID and password.
- Sometimes of course, it can be hard to get changes into applications, so the user ID and password can be provided using an exit instead of changing the code. Client side security exits or the pre-connect exit, can make changes to the MQCONN before it is sent to the queue manager, and the security exit in fact is designed to allow the setting of the MQCSP since V6 (so clients do not need to be updated to the new version in order to use this).

Capitalware's MQ Technical Conference v2.0.1.4

Procedural MQI changes

- **MQCSP structure**
 - ▶ Connection Security Parameters
 - ▶ User ID and password
- **MQCNO structure**
 - ▶ Connection Options
- **WebSphere MQ V6**
 - ▶ Passed to OAM (Dist only)
 - ▶ Also passed to Security Exit
 - Both z/OS and Distributed
 - MQXR_SEC_PARMS
- **WebSphere MQ V8**
 - ▶ Acted upon by the queue manager (all platforms)

```
MQCNO cno = {MQCNO_DEFAULT};  
  
cno.Version = MQCNO_VERSION_5;  
  
cno.SecurityParmsPtr = &csp;  
  
MQCONNX(QMName,  
        &cno,  
        &hConn,  
        &CompCode,  
        &Reason);
```

```
MQCSP csp = {MQCSP_DEFAULT};  
  
csp.AuthenticationType = MQCSP_AUTH_USER_ID_AND_PWD;  
csp.CSPUserIdPtr      = "hughson";  
csp.CSPUserIdLength  = 7;          /* Max: MQ_CLIENT_USER_ID_LENGTH */  
csp.CSPPasswordPtr   = "passw0rd";  
csp.CSPPasswordLength = 8;        /* Max: MQ_CSP_PASSWORD_LENGTH */
```

Capitalware's MQ Technical Conference v2.0.1.4

Object Oriented MQ classes changes

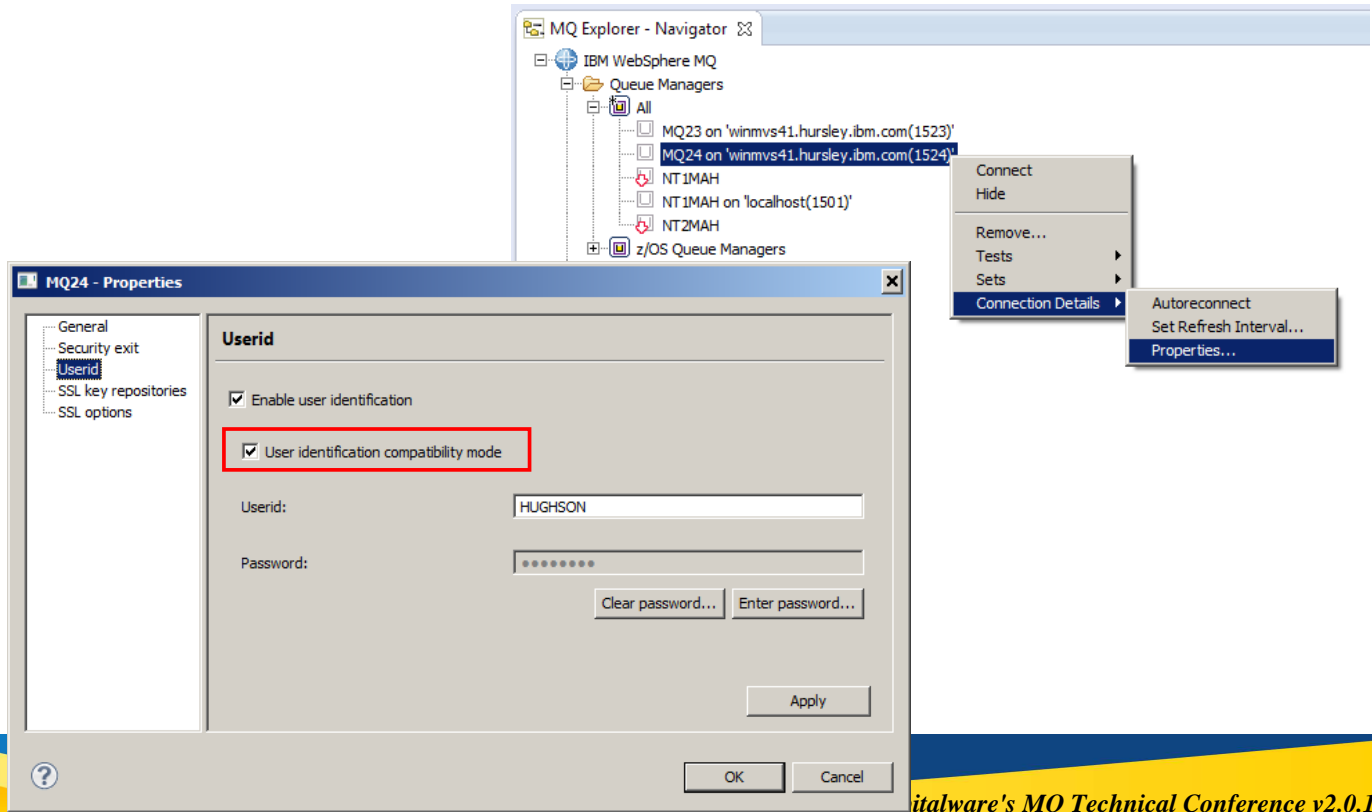
```
MQEnvironment.properties = new Hashtable();  
MQEnvironment.userID = "hughson";  
MQEnvironment.password = "passw0rd";  
  
System.out.println("Connecting to queue manager");  
MQQueueManager qMgr = new MQQueueManager(QMName);
```

JMS/XMS classes changes

```
cf = getCF();  
  
System.out.println("Creating the Connection with UID and Password");  
Connection conn = cf.createConnection("hughson", "passw0rd");
```

Capitalware's MQ Technical Conference v2.0.1.4

Using it from the MQ Explorer GUI



Capitalware's MQ Technical Conference v2.0.1.4

Using it from the MQ Explorer GUI – Notes

N
O
T
E
S

- The WebSphere MQ Explorer GUI is an MQ Java™ application, so since there is a programming interface for MQ Java to supply a user ID and password, the Explorer GUI can use this.
- To configure the Explorer to use a user ID and password on a connection to a queue manager (whether local or client connection), select Connection Details->Properties... from the right-mouse context menu on the queue manager. In the dialog that appears, choose Userid. This panel is the same for both local or client connections in WebSphere MQ V8, although the Properties dialog will have less selections for other things in the local case.
- Explorer has a password cache which will need to be enabled in order to use passwords. If you have never used it before there will be a link on this panel to take you through it.
- The other interesting item here is the "User identification compatibility mode" check box. This is for those of you who have been using Security exits with the Explorer in the past. The Java client previously did not use the MQCSP structure to supply its user ID and password in previous releases, and there are many exits written that have discovered where the user ID and password were provided instead. In order to retain compatibility for this, the Java client has two modes. It can run in compatibility mode and maintain what you had before, or it can run with the V8 mode and use the MQCSP. The check box shown is how you set that property in the Explorer GUI. For other Java applications, you need to set property to indicate you are happy to use the MQCSP method.
- At the queue manager, if no MQCSP is sent by a client, but the user ID and password are provided in this alternate method that was utilised by Java Clients, the V8 queue manager will accept this and drive the same password check as is used for the MQCSP provided passwords.

Capitalware's MQ Technical Conference v2.0.1.4

Using MQCSP from Java Client

- **Java client (not local bindings) has two ways to send password**
 - ▶ FAP Flow
 - ▶ MQCSP structure
- **FAP Flow**
 - ▶ Mechanism used by many customer security exits
 - ▶ Retained as default
 - ▶ Restricted to 8 characters user IDs and passwords
 - ▶ Not protection by password protection algorithm
 - ▶ Used by Connection Authentication if seen and no MQCSP found
- **MQCSP structure**
 - ▶ Used by Java Client when property set
 - ▶ Non-default
 - ▶ Allows longer user IDs and passwords
 - ▶ Can be protection by password protection algorithm

MQ Classes for Java

set the property **MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY** to true in the properties hashtable passed to the `com.ibm.mq.MQQueueManager` constructor.

MQ Classes for JMS

set the property **JMSConstants.USER_AUTHENTICATION_MQCSP** to true on the appropriate connection factory prior to creating the connection

Globally

set the **System Property "com.ibm.mq.cfg.jmqi.useMQCSPauthentication"** to a value indicating true, for example by adding **"-Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=Y"** to the command line

Capitalware's MQ Technical Conference v2.0.1.4

Using MQCSP from Java - Notes

N
O
T
E
S

- We saw on a previous page the example code you might use to provide the user ID and password from a Java classes application or a JMS application. This is actually nothing new. Java clients have been able to send a user ID and password across the channel FAP before. This part of the FAP was very restrictive though, it only allowed or 8 character user IDs and 8 character passwords. And, of course, it was only for clients. The MQCSP interface was designed not to have such limitations.
- There are quite a number of customers pre-V8 who have security exits written to pull the user ID and password sent by Java clients in this way. Because of this, we could not change the default of the Java clients over to use the MQCSP or all these security exits would have to be changed. So by default, Java clients continue to send the user ID and password as this restrictive FAP flow.
- On the queue manager end, if we receive a user ID and password in this FAP flow, and no MQCSP structure, we will use the user ID and password in the FAP flow for Connection Authentication, so you don't have to make any changes in order to remove a security exit that is checking the user ID and password in this way.
- However, there are benefits to using the MQCSP structure, including password protection and the increased length of the fields, so when you are ready to change over to use MQCSP instead of the FAP flow in a Java client, you need to set the system property.

Capitalware's MQ Technical Conference v2.0.1.4

Client side Security Exit

mqccred.ini

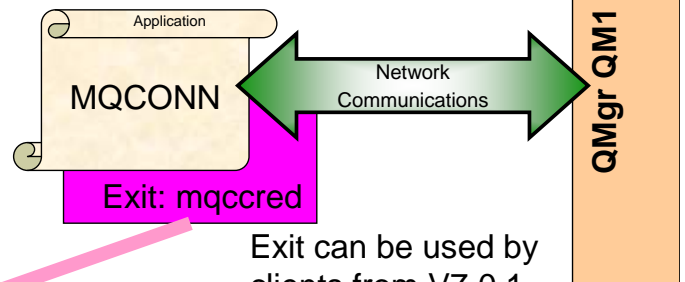
```
AllQueueManagers:  
  User=abc  
  password=newpw  
QueueManager:  
  Name=QMA  
  User=user1  
  password=passw0rd
```

Tool: runmqccred

mqccred.ini

```
AllQueueManagers:  
  User=abc  
  OPW=%^&aervrgtsr  
QueueManager:  
  Name=QM1  
  User=user1  
  OPW=H&^dbgfh
```

File permissions



Exit can be used by clients from V7.0.1 and later (by copying from a V8 installation)

Capitalware's MQ Technical Conference v2.0.1.4

Client side Security Exit – Notes

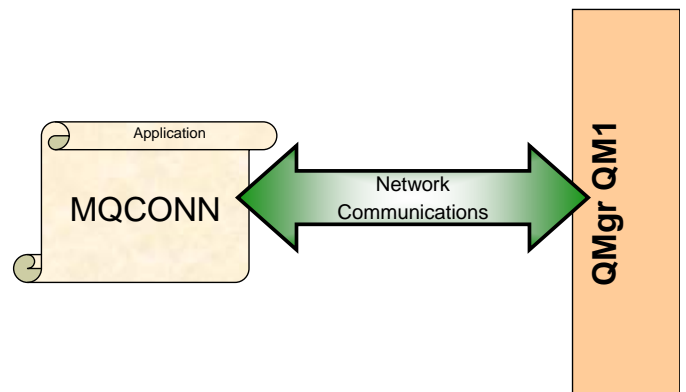
N
O
T
E
S

- To make changes to applications, especially the very prevalent client attached applications where we see the strongest use case for using user ID and password, is difficult for customers. To aid with this issue, WebSphere MQ V8 provides a client side security exit which can set the user ID and password instead of making changes in the application to do this.
- The exit runs at the CLNTCONN end of the channel and pulls the user ID and the password from a file. This file is controlled by means of OS file permissions. If the exit discovers that the file permissions are too open, it will cause a failure thus ensuring that this important part of protecting the passwords does not go unnoticed.
- The file is additionally obfuscated from casual browsers. The algorithm for this obfuscation is not published, and neither is the source of the exit.
- The exit will be built in such a way that it can be picked up from a V8 installation and copied to a V7.0.1 client installation (or later). Note that using a client installation of < V8 will mean you have the password flowed in the clear. Only V8 and later at both ends will provide the ability to protect the flowed password without the need to use SSL/TLS.
- Along with the exit, we also supply a tool which is used to obfuscate the file containing the passwords.
- See blog post:-
https://www.ibm.com/developerworks/community/blogs/messaging/entry/bitesize_blogging_mq_v8_mqccred_exit

Capitalware's MQ Technical Conference v2.0.1.4

Protecting your password across a network

- **Use SSL/TLS**
 - ▶ Perhaps with anonymous clients
- **If no SSL/TLS**
 - ▶ If both ends are V8
 - ▶ MQ Code will protect the password – so not sent in the clear
- **If client is < V8**
 - ▶ No MQ password protection
 - ▶ Consider SSL/TLS



Capitalware's MQ Technical Conference v2.0.1.4

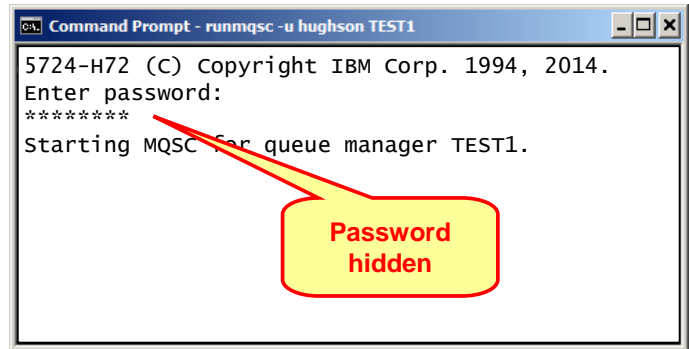
Protecting your password across a network – Notes

- N**
- When an application connects to a WebSphere MQ V8 queue manager across the network, i.e. making a client connection, the password it sends for connection authentication purposes travels across the network from the client application to the queue manager for checking. This password should be protected as it does so, so that network sniffers cannot obtain your password.
- O**
- For best possible protection, you can of course use SSL/TLS. You might imagine using anonymous SSL/TLS, i.e. the client does not have a certificate, since you are using user ID and password as the means by which to verify the identity of the client application.
- T**
- If you do not use SSL/TLS, and your client is at V8.0 or later, the WebSphere MQ product code will protect your password so that it is not sent in the clear. A good reason to get your clients upgraded to V8!
- E**
- If your WebSphere MQ Client is at a version earlier than V8.0, it can still send user ID and passwords (since the MQCSP structure has been around since V6) but the password will not be protected, so you should consider using SSL/TLS.
- S**

Capitalware's MQ Technical Conference v2.0.1.4

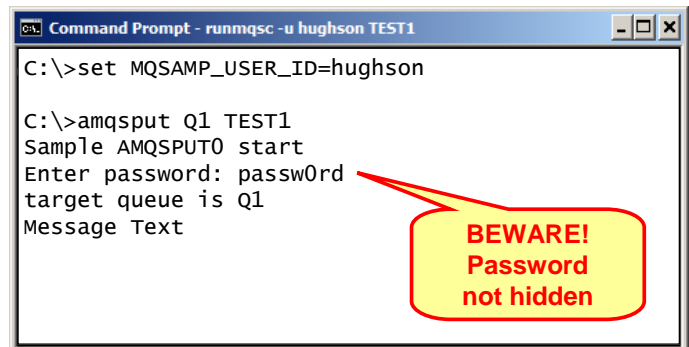
MQ Samples/Tools that can use it

- MQ Explorer (earlier page)
- runmqsc
 - ▶ New -u parameter and password prompt
- 'C' samples
 - ▶ amqscnxc
 - New -u parameter and password prompt
 - ▶ amqsput(c), amqsget(c) and amqsbcg(c)
 - Positional parameters
 - Use MQSAMP_USER_ID=<userid> to provide user ID and cause password prompt
- JMS samples
 - ▶ JmsProducer and JmsConsumer
 - New -u and -w parameters to supply user ID and password respectively



```
Command Prompt - runmqsc -u hughson TEST1
5724-H72 (C) Copyright IBM Corp. 1994, 2014.
Enter password:
*****
Starting MQSC for queue manager TEST1.
```

Annotation: Password hidden



```
Command Prompt - runmqsc -u hughson TEST1
C:\>set MQSAMP_USER_ID=hughson
C:\>amqsput Q1 TEST1
Sample AMQSPUT0 start
Enter password: passw0rd
target queue is Q1
Message Text
```

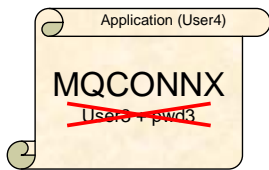
Annotation: BEWARE! Password not hidden

MQ Samples/Tools that can use it - Notes

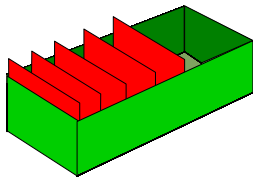
N
O
T
E
S

- There are various tools and samples that are part of MQ where you can supply a user ID and password when they connect to MQ.
- We already saw the screenshots of MQ Explorer on an earlier page.
- The runmqsc tool, which in MQ V8 can run as a client or as a locally bound connection, has been updated to take a new parameter -u, which supplies your user ID, and then prompt you for a password. This password prompt will also hide your password as you type it so no-one can look over you shoulder!
- Various samples have also been updated, in varying ways, to allow you to provide a user ID and password when using them. We've tried to update the samples we thought were most likely to be used by our customers, without updating every single one. The samples don't contain code to hide the password when it is being typed in. This was code that would detract from the purpose of the sample which is to show how to write MQ applications.
- See blog posts:-
 - https://www.ibm.com/developerworks/community/blogs/messaging/entry/bitesize_blogging_mq_v8_samples_can_use_user_id_and_password
 - https://www.ibm.com/developerworks/community/blogs/messaging/entry/bitesize_blogging_mq_v8_client_mqsc

Error notification



MQRC_NOT_AUTHORIZED (2035)



SYSTEM.ADMIN.QMGR.EVENT

ALTER QMGR AUTHOREV(ENABLED)

- **Application**
 - ▶ MQRC_NOT_AUTHORIZED (2035)

- **Administrator**
 - ▶ Error message

- **Monitoring Tool**
 - ▶ Not Authorized Event message (Type 1 – Connect)
 - ▶ MQRQ_CONN_NOT_AUTHORIZED (existing)
 - Connection not authorized.
 - ▶ MQRQ_CSP_NOT_AUTHORIZED (new)
 - User ID and password not authorized.
 - ▶ Additional field to existing connect event
 - MQCACF_CSP_USER_IDENTIFIER

Capitalware's MQ Technical Conference v2.0.1.4

Error notification – Notes

N
O
T
E
S

- When an application provides a user ID and password which fail the password check, the application is returned the standard MQ security error, 2035 – MQRC_NOT_AUTHORIZED.
- The MQ administrator will see this reported in the error log and can therefore see that the application was rejected due to the user ID and password failing the check, rather than, for example, a lack of connection authority (+connect).
- A monitoring tool can also be notified of this failure if authority events are on - ALTER QMGR AUTHOREV(ENABLED) – via an event message to the SYSTEM.ADMIN.QMGR.EVENT queue. This Not Authorized event is a Type 1 – Connect – event and provides all the same fields as the existing Type 1 event, along with one, additional field, the MQCSP user ID provided. The password is not provided in the event message. This means that there are two user IDs in the event message, the one the application is running as and the one the application presented for user ID and password checking.

Capitalware's MQ Technical Conference v2.0.1.4

Error Messages

- **Incorrect password**
 - ▶ Distributed - **AMQ5534: User ID 'hughson' authentication failed**
 - Followed by AMQ5542 (giving hint for why)
 - ▶ z/OS MSTR - RACF ICH408I message
 - Or equivalent for other External Security Managers
 - Issue DISPLAY SECURITY for current CONNAUTH settings
- **Missing password**
 - ▶ Distributed - **AMQ5540: Application 'D:\nttools\q.exe' did not supply a user ID and password**
 - ▶ z/OS MSTR - **CSQH045E cpf csect app-identifier did not provide a password**
 - *app-identifier* has different contents for locally bound app and client app.
- **Missing password due to CHLAUTH CHCKCLNT upgrade**
 - ▶ Distributed - **AMQ9791: The client application did not supply a user ID and password.**
 - ▶ z/OS CHIN - **CSQX791I cpf csect Client application app-name from address ip-address did not supply a user ID and password, Detail: conndetails**

Error Messages - Notes

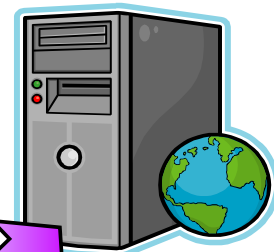
N
O
T
E
S

- This feature introduces some new error messages that it is helpful to be aware of to allow you to work out why your application is receiving an MQRC_NOT_AUTHORIZED (2035) reason code.
- See also blog post:-
https://www.ibm.com/developerworks/community/blogs/messaging/entry/bitesize_blogging_mq_v8_connection_authentication_on_z_os

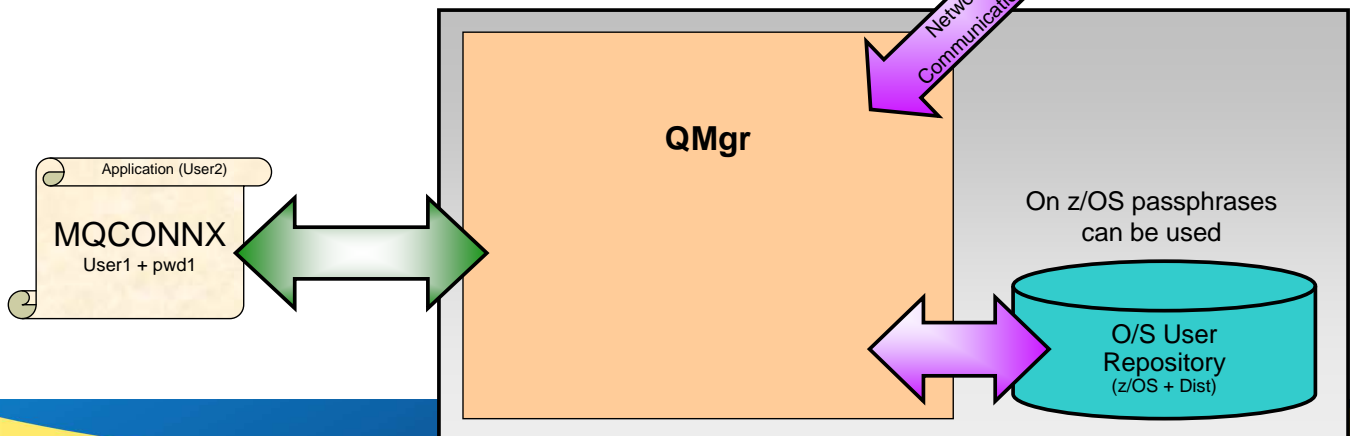
User Repositories

```

DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) AUTHTYPE(IDPWLDAP)
  CONNAME('ldap1(389),ldap2(389)')
  LDAPUSER('CN=QMGR1')
  LDAPPWD('passw0rd') SECCOMM(YES)
    
```



LDAP Server (Dist only)



Capitalware's MQ Technical Conference v2.0.1.4

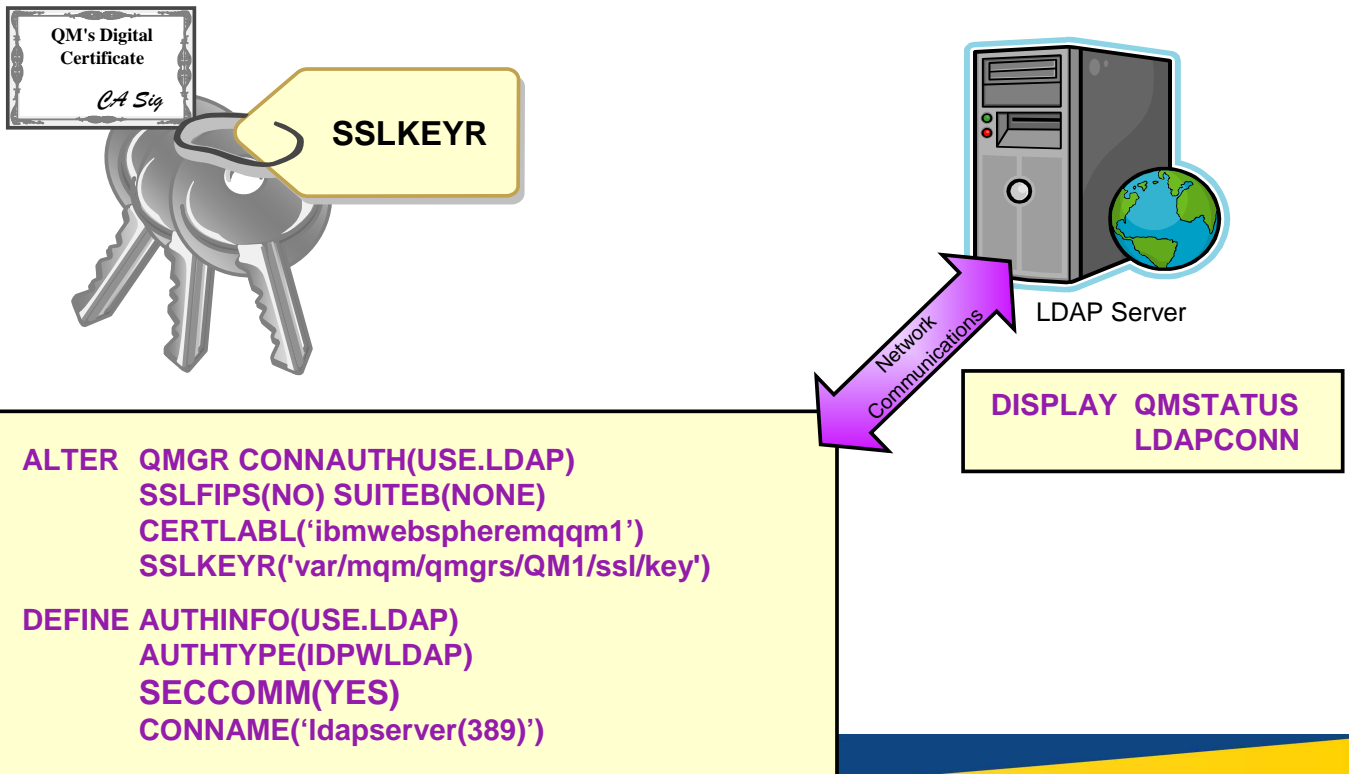
User Repositories – Notes

N
O
T
E
S

- So far we have spoken about user ID and password authentication without mentioning what is actually doing the authentication. We've also shown that there is a new type of authentication information object without showing you the object type. Here we introduce two new object types of authentication information objects.
- The first type is used to indicate that the queue manager is going to use the local O/S to authenticate the user ID and password. This type is IDPWOS. This includes the use of password phrases on z/OS
- The second type is used to indicate that the queue manager is going to use an LDAP server to authenticate the user ID and password. This type is IDPWLDAP and is not applicable on z/OS.
- Only one type can be chosen for the queue manager to use by naming the appropriate authentication information object in the queue manager's CONNAUTH attribute.
- We have already covered everything there is to say about the configuration of the O/S as the user repository as the common attributes are all there is for the O/S. There is more to say about the LDAP server as an option though.
- Some of the LDAP server configuration attributes are probably fairly obvious. The CONNAME is how the queue manager knows where the LDAP server is, and SECCOMM controls whether connectivity to the LDAP server will be done using SSL/TLS or not. The LDAPUSER and LDAPPWD attributes are how the queue manager binds to the LDAP server so that it can look-up information about user records. It is likely this may be a public area of an LDAP server, so these attributes may not be needed.
- It is worth highlighting that the CONNAME field can be used to provide additional addresses to connect to for the LDAP server in a comma-separated list. This can aid with redundancy if the LDAP server does not provide such itself.

Capitalware's MQ Technical Conference v2.0.1.4

Secure connection to an LDAP Server



Capitalware's MQ Technical Conference v2.0.1.4

Secure connection to an LDAP Server – Notes

N
O
T
E
S

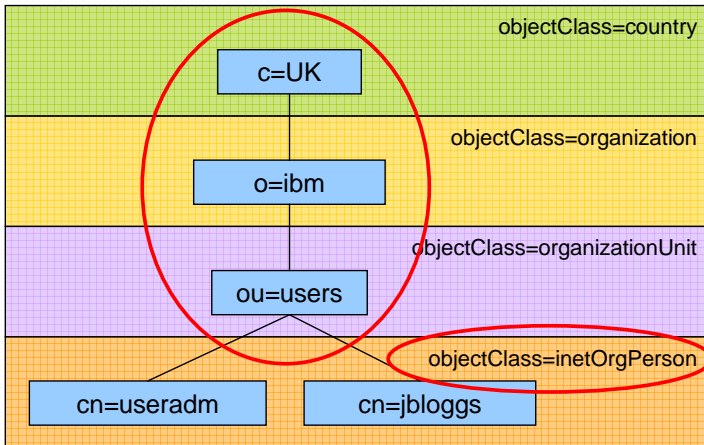
- Unlike on channels, there is no SSLCIPH parameter to turn on the use of SSL/TLS for the communication with the LDAP server. In this case MQ is acting as a client to the LDAP server so much of the configuration will be done at the LDAP server. Some existing parameters in MQ will be used to configure how that connection will work as shown on this slide.
- The overall switch to choose SSL/TLS communication or not, we already saw on the previous page – SECCOMM.
- In addition to this attribute, we will also pay attention to the queue manager attributes SSLFIPS and SUITEB to restrict the set of cipher specs that will be chosen. The certificate that will be used to identify the queue manager to the LDAP server will be the queue manager certificate, either 'ibmwebspheremq<qmgr-name>' or the newly added CERTLABL attribute which we'll talked about in an earlier section of this presentation.
- Certificate revocation will be checked by using the OCSP servers that are named in the AuthorityInfoAccess (AIA) certificate extensions. This can be turned off by using the qm.ini SSL stanza attribute OCSPCheckExtensions.
- Connection to an LDAP Server is made as a network connection (which is why you may wish to consider using a secure connection). The status of this connection from the queue manager to the LDAP server is shown in DISPLAY QMSTATUS.

Capitalware's MQ Technical Conference v2.0.1.4

LDAP User Repository

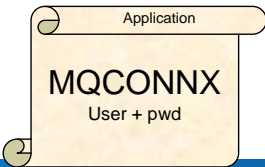


LDAP Server



```

DEFINE AUTHINFO(USE.LDAP)
  AUTHTYPE(IDPWLDAP)
  CONNAME('ldapserver(389)')
  CLASSUSR('inetOrgPerson')
  BASEDNU('ou=users,o=ibm,c=uk')
  USRFIELD('cn')
  
```



Application provides	USRFIELD	BASEDNU
cn=useradm,ou=users,o=ibm,c=uk		
cn=useradm		Adds ou=users,o=ibm,c=uk
useradm	Adds cn=	Adds ou=users,o=ibm,c=uk

LDAP User Repository – Notes

N
O
T
E
S

- When using an LDAP user repository there is some more configuration to be done on the queue manager other than just to tell the queue manager where the LDAP repository resides.
- User IDs records defined in an LDAP server have a hierarchical structure in order to uniquely identify them. So an application could connect to the queue manager and present its user ID as being the fully qualified hierarchical user ID. This however is a lot to provide and it would be simpler if we could configure the queue manager to say, assume all user IDs that are presented are found in this area of the LDAP server and add that qualification onto anything you see. This is what the BASEDNU attribute is for. It identifies the area in the LDAP hierarchy that all the user IDs are to be found. Or to look at it another way, the queue manager will add the BASEDNU value to the user ID presented by an application to fully qualify it before looking it up in the LDAP server.
- Additionally, your application may only want to present the user ID without providing the LDAP attribute name, e.g. CN=. This is what the USRFIELD is for. Any user ID presented to a queue manager without an equals sign (=) will have the attribute and the equals sign pre-pended to it, and the BASEDNU value post-pended to it before looking it up in the LDAP server. This may be a useful migratory aid when moving from O/S user IDs to LDAP user IDs as the application could very well be presenting the same string in both cases, thus avoiding any change to the application.

Relationship to Authorization – LDAP

Logfiles Help

Edit an entry cn=useradm,ou=users,o=ibm,c=uk

Edit an entry
 → Edit an entry
 Optional attributes

Object class inheritance:
 inetOrgPerson

Distinguished name (DN)

Relative DN: *cn=useradm Parent DN: ou=users,o=ibm,c=uk

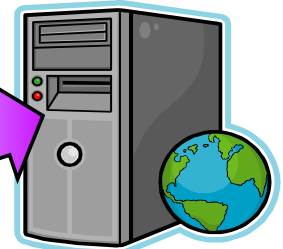
Required attributes

Enter the values for the attributes of the entry. For multiple values click **Multiple values** next to the attribute.

cn: useradm Multiple values

sn: mqmadm

DEFINE AUTHINFO(USE.LDAP)
 AUTHTYPE(IDPWLDAP)
 CONNAME('ldap(389)')
 ADOPTCTX(YES)
 SHORTUSR('sn')

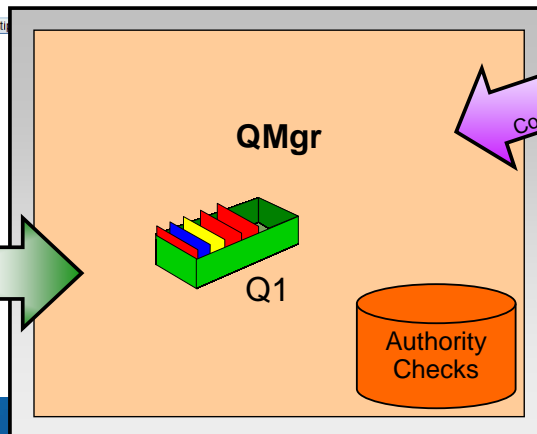


LDAP Server

Network Communications

Application

MQCONNX
 cn=useradm
 MQOPEN



Authority Records

Q1: mqmadm +put

Relationship to Authorization – LDAP - Notes

N
O
T
E
S

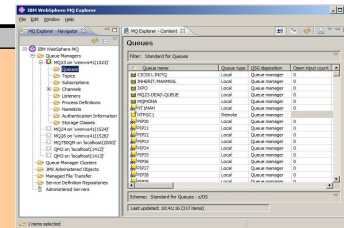
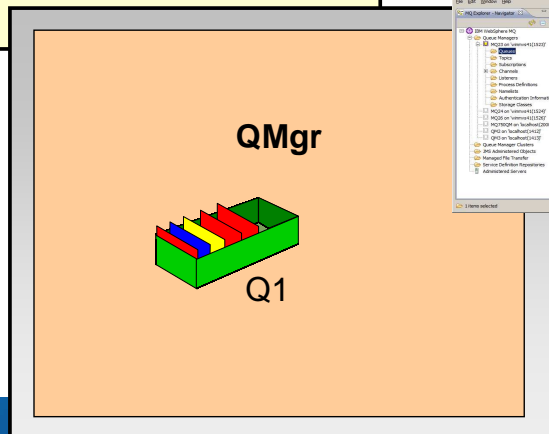
- We spoke earlier about the ability to adopt the authenticated user ID as the context for this connection. So how does this work if you are using LDAP as the user repository but your authorization is being done using O/S user IDs?
- We need to get a user to represent the LDAP user that has been presented, as an O/S user ID. We find this from the LDAP user record. This can be any field that is defined in the user record, perhaps something like the short name field (sn=) that is a mandatory part of the definition of the inetOrgPerson class, or perhaps something defined more specifically for the purpose such as a user ID (uid=) field.
- The queue manager will use that information to determine what O/S user ID will be used as the context for this connection. You configure it using SHORTUSR to say what the field to locate in the user record is.

Migration / Defaults

AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
 AUTHTYPE(IDPWOS)
 CHCKLOCL(OPTIONAL)
 CHCKCLNT(REQDADM)
 FAILDLAY(1)
 DESCR()
 ALTDATA(2013-12-25)
 ALTTIME(12.00.00)

Defaults

- ▶ Migrated queue manager
 - CONNAUTH('')
- ▶ New queue manager
 - CONNAUTH(←)



Migration / Defaults – Notes

N
O
T
E
S

- By default, a migrated queue manager will find that CONNAUTH is blank – and therefore connection authentication is switched off.
- A brand new queue manager created with the WebSphere MQ V8 binaries will find that the CONNAUTH field points to the SYSTEM.DEFAULT.AUTHINFO.IDPWOS authentication information object.

Summary - Connection Authentication

- **Application provides User ID and password in MQCSP**
 - ▶ Or uses mqccred exit supplied
- **Queue Manager checks password against OS or LDAP**
 - ▶ `ALTER QMGR CONNAUTH('CHECK.PWD')`
 - ▶ `DEFINE AUTHINFO('CHECK.PWD')`
 - `AUHTYPE(IDPWOS | IDPWLDAP)`
 - `CHCKLOCL(NONE | OPTIONAL | REQUIRED | REQDADM)`
 - `CHCKCLNT(NONE | OPTIONAL | REQUIRED | REQDADM)`
 - `ADOPTCTX(YES)`
 - + various LDAP attributes
 - ▶ `REFRESH SECURITY TYPE(CONNAUTH)`
- **Password protection is provided when SSL/TLS not in use**
 - ▶ Both ends of client channel are V8 or above