

How to utilize a user ID and a password in IBM MQ and validate using a client-server setup

<https://www.ibm.com/support/pages/node/6855791>

Date last updated: 17-Jan-2023

Abhishek Vishwakarma
IBM MQ Support

<https://www.ibm.com/products/mq/support>

Find all the support you need for IBM MQ

+++ Objective +++

This documentation has the steps for setting up a simple client-server configuration using user ID and password. The setup of the user and password is the first step for securing the system. After validating the user and password, SSL can be configured to add more layers of security. While altering the MQ configuration b/w server and client, the doc lists entire steps for both client and server. Follow the doc thoroughly and in each step, check the queue manager configuration for a better understanding of the topic.

This doc contains several failed scenarios and passed scenarios. Go through each one and hone your MQ skills

Servers used in the doc:

MQ server: Windows (9.46.97.147)

Client server: Linux (9.46.72.151)

NOTE: Use your server IPs in the setup.

Table of Contents

SCENARIO 1: QUEUE MANAGER SETUP IN WINDOWS AND CLIENT IN LINUX.....	3
SCENARIO 2: SET MQSAMP_USER_ID ENVIRONMENT VARIABLE ON CLIENT SIDE (PUT MESSAGES ON QUEUE) [FAILED CASE SCENARIO].....	5
SCENARIO 3: SET 'CHCKCLNT' ATTRIBUTE VALUE TO 'NONE' ON QUEUE MANAGER AND SEND MESSAGE FROM THE CLIENT SIDE [PASSED CASE SCENARIO]	7
SCENARIO 4: SET 'CHCKCLNT' ATTRIBUTE VALUE TO 'REQUIRED' ON QUEUE MANAGER AND SEND MESSAGE FROM THE CLIENT SIDE. [FAILED CASE SCENARIO]	8
SCENARIO 5: SET CHLAUTH FOR CHANNEL ON MQ SERVER SIDE AND SET MQSAMP_USER_ID ENVIRONMENT VARIABLE ON CLIENT SIDE. [PASSED CASE SCENARIO]	10
SCENARIO 6: PUT MESSAGE FROM CLIENT SERVER AND GET MESSAGE FROM MQ SERVER AS A CLIENT [PASSED CASE SCENARIO]	11
SCENARIO 7: PUT FROM CLIENT1 AND GET FROM CLIENT2 [PASSED CASE SCENARIO]	12
SCENARIO 8: QUEUE MANAGER SETUP IN MQ SERVER (LINUX) AND DOWNLOAD PROJECT IN CLIENT SERVER (WINDOWS).....	13
SCENARIO 9: UPDATE THE APPLICATION.PROPERTIES FILE AND SEND MESSAGES FROM JAVA CLIENT (IN WINDOWS) TO MQ SERVER (IN LINUX) [FAILED CASE SCENARIO]	14
SCENARIO 10: ADD 'ROOT' USER IN 'MQM' GROUP (IN LINUX) AND RUN JAVA PROGRAM AGAIN (IN WINDOWS).....	16
SCENARIO 11: GIVE PERMISSIONS TO THE 'ROOT' USER (IN LINUX) AND RUN JAVA PROGRAM	18

SCENARIO 1: QUEUE MANAGER SETUP IN WINDOWS AND CLIENT IN LINUX

Setting in Windows (MQ Server):

Create a test user ('ABHISHEK' in my case) and add in the mqm group. This user will be used as a MCA for authorization to access IBM MQ resources.

1. **crtmqm QM.FOR.CLIENT**
2. **strmqm QM.FOR.CLIENT**
3. **runmqsc QM.FOR.CLIENT**
4. **define listener(QM.FOR.CLIENT.LISTENER) trptype(tcp) port(1426) control(qmgr)**
5. **define channel(QM.FOR.CLIENT.CHAN) chltype(svrconn) mcauser(ABHISHEK)**
6. **define**
channel(QM.FOR.CLIENT.CHAN) chltype(clntconn) conname('9.46.97.147(1426)') qmname(QM.FOR.CLIENT)
7. **DEFINE QLOCAL (QUEUE.LOCAL)**
8. **SET CHLAUTH('QM.FOR.CLIENT.CHAN') TYPE (BLOCKUSER) USERLIST(NOBODY)**

Configuration details:

```
display qmgr connauth
```

AMQ8408I: Display Queue Manager details.

```
QMNAME(QM.FOR.CLIENT)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
```

```
display AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
```

AMQ8566I: Display authentication information details.

```
AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
AUTHTYPE(IDPWOS)           ADOPTCTX(YES)
DESCR( )                   CHCKCLNT(OPTIONAL)
CHCKLOCL(OPTIONAL)        FAILDLAY(1)
AUTHENMD(OS)              ALTDATA(2022-12-12)
ALTTIME(10.05.06)
```

```
dis chlauth(QM.FOR.CLIENT.CHAN)
```

AMQ8878I: Display channel authentication record details.

```
CHLAUTH(QM.FOR.CLIENT.CHAN) TYPE(BLOCKUSER)
DESCR( )                   CUSTOM( )
USERLIST(NOBODY)          WARN(NO)
ALTDATA(2022-12-12)       ALTTIME(10.44.54)
```

Setting in Linux (Client):

Try to put the message in the queue named QUEUE.LOCAL.

1. **export MQSERVER='QM.FOR.CLIENT.CHAN/TCP/9.46.97.147(1426)'**

2. `cd /opt/mqm/samp/bin`
3. `./amqsputc QUEUE.LOCAL QM.FOR.CLIENT`

Note: We have not passed any user from the client side.

SCENARIO 2: SET MQSAMP_USER_ID ENVIRONMENT VARIABLE ON CLIENT SIDE (PUT MESSAGES ON QUEUE) [FAILED CASE SCENARIO]

Changes in the Linux (Client):

By setting the below environment variable, we are providing user ID to the MQ server. Now, it depends on MQ, whether MQ wants to check the user/password or not. This functionality is defined in the **CHCKCLNT** attribute on the MQ server side.

```
export MQSAMP_USER_ID=root
```

[Note: The user "root" does not exist in the windows OS (MQ Server). It will throw an error]

Steps:

```
[root@gruffest1 bin]# ./amqsputc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSPUT0 start
Enter password: *****
MQCONN ended with reason code 2035
```

RC 2035 is recorded because MQ server doesn't know about the "root" user and CHCKCLNT value is OPTIONAL, which means it will check the user & password if supplied by the client.

Error reported in the windows (C:\ProgramData\IBM\MQ\qmgrs\QM!FOR!CLIENT\errors\AMQERR01):

```
12/13/2022 05:14:15 - Process(6012.22) User(Administrator) Program(amqzlaa0.exe)
Host(SERRANID1) Installation(Installation1)
VRMF(9.2.0.4) QMgr(QM.FOR.CLIENT)
Time(2022-12-13T05:14:15.367Z)
CommentInsert1(root)
CommentInsert2(amqsputc)
CommentInsert3(N/A)
```

AMQ5534E: User ID 'root' authentication failed

EXPLANATION:

The user ID and password supplied by the 'amqsputc' program could not be authenticated.

Additional information: 'N/A'.

ACTION:

Ensure that the correct user ID and password are provided by the application.
Ensure that the authentication repository is correctly configured. Look at previous error messages for any additional information.

```
----- amqzfuca.c : 4816 -----
12/13/2022 05:14:15 - Process(6012.22) User(Administrator) Program(amqzlaa0.exe)
Host(SERRANID1) Installation(Installation1)
VRMF(9.2.0.4) QMgr(QM.FOR.CLIENT)
Time(2022-12-13T05:14:15.368Z)
CommentInsert1(root)
CommentInsert2(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CommentInsert3(CHCKCLNT(OPTIONAL))
```

AMQ5542I: The failed authentication check was caused by the queue manager CONNAUTH CHCKCLNT(OPTIONAL) configuration.

EXPLANATION:

The user ID 'root' and its password were checked because the queue manager connection authority (CONNAUTH) configuration refers to an authentication information (AUTHINFO) object named 'SYSTEM.DEFAULT.AUTHINFO.IDPWOS' with CHCKCLNT(OPTIONAL).

This message accompanies a previous error to clarify the reason for the user ID and password check.

ACTION:

Refer to the previous error for more information.

Ensure that a password is specified by the client application and that the password is correct for the user ID. The authentication configuration of the queue manager connection determines the user ID repository. For example, the local operating system user database or an LDAP server.

Note that if the authentication configuration specifies an LDAP user repository, a CHCKCLNT value of REQDADM is treated as equivalent to a value of OPTIONAL.

If the CHCKCLNT setting is OPTIONAL, the authentication check can be avoided by not passing a user ID across the channel. For example, by omitting the MQCSP structure from the client MQCONN API call.

To avoid the authentication check, you can amend the authentication configuration of the queue manager connection, but you should generally not allow unauthenticated remote access.

----- amqzfuca.c : 4839 -----

SCENARIO 3: SET 'CHCKCLNT' ATTRIBUTE VALUE TO 'NONE' ON QUEUE MANAGER AND SEND MESSAGE FROM THE CLIENT SIDE [PASSED CASE SCENARIO]

Changes in Windows:

1. **runmqsc QM.FOR.CLIENT**
2. **ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(NONE)**
3. **REFRESH SECURITY**

Now, it doesn't matter if you provide the correct/wrong password because the MQ server will not check the user & password as CHCKCLNT value is NONE.

In Linux:

Try to put messages in the queue.

```
[root@gruffest1 bin]# ./amqspuqc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSPUT0 start
Enter password: *****
target queue is QUEUE.LOCAL
message
Sample AMQSPUT0 end
```

Able to put messages in the queue.

SCENARIO 4: SET 'CHCKCLNT' ATTRIBUTE VALUE TO 'REQUIRED' ON QUEUE MANAGER AND SEND MESSAGE FROM THE CLIENT SIDE. [FAILED CASE SCENARIO]

Changes in Windows:

Set the value of attribute CHCKCLNT to REQUIRED. Now, MQ Server will always check the pair of user & password.

1. **runmqsc QM.FOR.CLIENT**
2. **ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(REQUIRED)**
3. **REFRESH SECURITY**

Action performed in Linux:

```
[root@gruffest1 bin]# export MQSAMP_USER_ID=root
[root@gruffest1 bin]# ./amqsputc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSPUT0 start
Enter password: *****
MQCONN ended with reason code 2035
```

RC 2035 is logged because no user with the name "root" is defined in the MQ Administrative group (Windows OS). Check what queue manager error log says.

Errors reported in the Windows

(C:\ProgramData\IBM\MQ\qmgrs\QM!FOR!CLIENT\errors\AMQERR01):

```
12/13/2022 06:18:20 - Process(6012.32) User(Administrator) Program(amqz1aa0.exe)
Host(SERRANID1) Installation(Installation1)
VRMF(9.2.0.4) QMgr(QM.FOR.CLIENT)
Time(2022-12-13T06:18:20.808Z)
CommentInsert1(root)
CommentInsert2(amqsputc)
CommentInsert3(N/A)
```

AMQ5534E: User ID 'root' authentication failed

EXPLANATION:

The user ID and password supplied by the 'amqsputc' program could not be authenticated.

Additional information: 'N/A'.

ACTION:

Ensure that the correct user ID and password are provided by the application. Ensure that the authentication repository is correctly configured. Look at previous error messages for any additional information.

```
----- amqzfuca.c : 4816 -----
12/13/2022 06:18:20 - Process(6012.32) User(Administrator) Program(amqz1aa0.exe)
Host(SERRANID1) Installation(Installation1)
VRMF(9.2.0.4) QMgr(QM.FOR.CLIENT)
Time(2022-12-13T06:18:20.808Z)
CommentInsert1(root)
CommentInsert2(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CommentInsert3(CHCKCLNT(REQUIRED))
```

AMQ5542I: The failed authentication check was caused by the queue manager

CONNAUTH CHCKCLNT(REQUIRED) configuration.

EXPLANATION:

The user ID 'root' and its password were checked because the queue manager connection authority (CONNAUTH) configuration refers to an authentication information (AUTHINFO) object named 'SYSTEM.DEFAULT.AUTHINFO.IDPWOS' with CHCKCLNT(REQUIRED).

This message accompanies a previous error to clarify the reason for the user ID and password check.

ACTION:

Refer to the previous error for more information.

Ensure that a password is specified by the client application and that the password is correct for the user ID. The authentication configuration of the queue manager connection determines the user ID repository. For example, the local operating system user database or an LDAP server.

Note that if the authentication configuration specifies an LDAP user repository, a CHCKCLNT value of REQDADM is treated as equivalent to a value of OPTIONAL.

If the CHCKCLNT setting is OPTIONAL, the authentication check can be avoided by not passing a user ID across the channel. For example, by omitting the MQCSP structure from the client MQCONN API call.

To avoid the authentication check, you can amend the authentication configuration of the queue manager connection, but you should generally not allow unauthenticated remote access.

----- amqzfuca.c : 4839 -----

SCENARIO 5: SET CHLAUTH FOR CHANNEL ON MQ SERVER SIDE AND SET MQSAMP_USER_ID ENVIRONMENT VARIABLE ON CLIENT SIDE. [PASSED CASE SCENARIO]

In Windows:

Allow all IP addresses to connect to QM.FOR.CLIENT.CHAN. In the ADDRESS field, you can also define a particular IP for connection.

1. **runmqsc QM.FOR.CLIENT**
2. **SET CHLAUTH(QM.FOR.CLIENT.CHAN) TYPE(ADDRESSMAP) ADDRESS(*) USERSRC(CHANNEL)**

In Linux:

In Windows (MQ server), we have two users with name 'Administrator' and 'ABHISHEK'.

So, by setting below environment variable, we are providing userID and password (defined in MQ server) to MQ server from the client.

export MQSAMP_USER_ID=ABHISHEK

```
[root@gruffest1 bin]# ./amqsputc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSPUT0 start
Enter password: *****
target queue is QUEUE.LOCAL
message
Sample AMQSPUT0 end
```

Client can put messages in the queue. This scenario worked because user 'ABHISHEK' exists on Windows, and we are providing the correct password from the client.

SCENARIO 6: PUT MESSAGE FROM CLIENT SERVER AND GET MESSAGE FROM MQ SERVER AS A CLIENT [PASSED CASE SCENARIO]

Simultaneous put and get from queue:

In Linux (MQ Client):

Put messages to queue from client:

./amqsputc QUEUE.LOCAL QM.FOR.CLIENT

```
[root@gruffest1 bin]# ./amqsputc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSPUT0 start
Enter password: *****
target queue is QUEUE.LOCAL
hii
hello
Sample AMQSPUT0 end
```

In Windows (MQ Server):

Get message from queue using amqsgetc (It acts as a client for MQ server):

- 1. SET MQSERVER=QM.FOR.CLIENT.CHAN/TCP/ 9.46.97.147(1426)**
- 2. SET MQSAMP_USER_ID=ABHISHEK**
- 3. cd C:\Program Files\IBM\MQ\bin**
- 4. amqsgetc QUEUE.LOCAL QM.FOR.CLIENT**

```
C:\Program Files\IBM\MQ\bin>amqsgetc QUEUE.LOCAL QM.FOR.CLIENT
Sample AMQSGET0 start
Enter password: *****
message <hii>
message <hello>
no more messages
Sample AMQSGET0 end
```

SCENARIO 7: PUT FROM CLIENT1 AND GET FROM CLIENT2 [PASSED CASE SCENARIO]

Include three servers and test put & get operation:

Data flow:

Client server1 -----msg-----> MQ server (Windows) -----msg-----> Client server2

MQ server: Windows (9.46.97.147)

Client server1 (put): Linux (9.46.72.151)

Client server2 (get): Linux (9.46.79.230)

In Client server1:

./amqsputc QUEUE.LOCAL QM.FOR.CLIENT

In Client server2:

export MQSERVER='QM.FOR.CLIENT.CHAN/TCP/ 9.46.97.147(1426)'

export MQSAMP_USER_ID=ABHISHEK

./amqsgetc QUEUE.LOCAL QM.FOR.CLIENT

Works as expected!!!

SCENARIO 8: QUEUE MANAGER SETUP IN MQ SERVER (LINUX) AND DOWNLOAD PROJECT IN CLIENT SERVER (WINDOWS)

MQ server: Linux (9.46.72.151)

Client server: Windows (9.46.97.147)

(NOTE: Here, the MQ server is in Linux and client server is in Windows)

Changes in Linux (here the Linux OS is the MQ server):

1. **crtmqm ECLIPSE.TO.LINUX**
2. **strmqm ECLIPSE.TO.LINUX**
3. **runmqsc ECLIPSE.TO.LINUX**
4. **define listener(ECLIPSE.TO.LINUX.LISTENER) trptype(tcp) port(1426) control(qmgr)**
5. **START LISTENER(ECLIPSE.TO.LINUX.LISTENER)**
6. **define channel(ECLIPSE.TO.LINUX.CHL) chltype(svrconn)**
7. **define**
channel(ECLIPSE.TO.LINUX.CHL) chltype(clntconn) conname('9.46.72.151(1426)') qmname(ECLIPSE.T
O.LINUX)
8. **DEFINE QLOCAL (QUEUE.LOCAL)**

Optional:

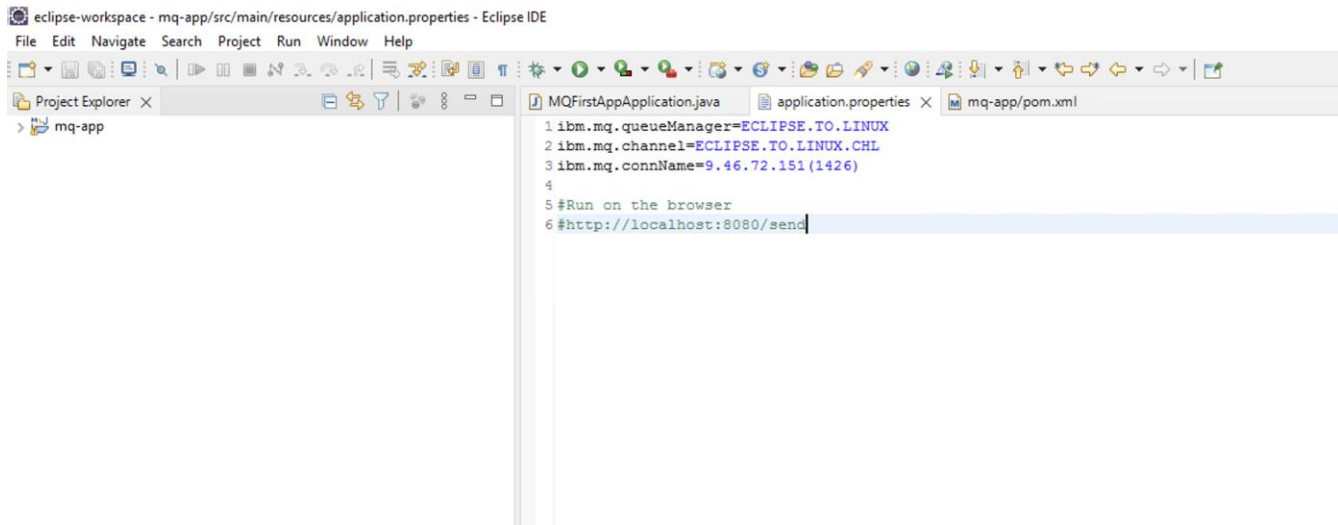
If qmgr directory path is not accessible, then use the below mentioned tips:

```
[mqm@gruffest1 qmgrs]$ cd ECLIPSE!TO!LINUX
bash: !TO!LINUX: event not found
[mqm@gruffest1 qmgrs]$ set +H
[mqm@gruffest1 qmgrs]$ cd ECLIPSE!TO!LINUX
[mqm@gruffest1 ECLIPSE!TO!LINUX]$
```

Changes in Windows (client):

Download the project: <https://github.com/ABHISHEKVISHWAKARMA80/mq-app>

SCENARIO 9: UPDATE THE APPLICATION.PROPERTIES FILE AND SEND MESSAGES FROM JAVA CLIENT (IN WINDOWS) TO MQ SERVER (IN LINUX) [FAILED CASE SCENARIO]



Run the java program “MQFirstAppApplication”.

The program will fail, check the queue manager error logs for error details.

Error reported in MQ server error logs (/var/mqm/qmgrs/ECLIPSE!TO!LINUX/errors):

```

12/17/22 03:06:27 - Process(265028.20) User(mqm) Program(amqzlaa0)
Host(gruffest1.fyre.ibm.com) Installation(Installation1)
VRMF(9.2.0.3) QMgr(ECLIPSE.TO.LINUX)
Time(2022-12-17T11:06:27.899Z)
CommentInsert1(admin)
CommentInsert2(app.MQFirstAppApplication)
CommentInsert3(Pipe returned 2292 [FAILED])

```

AMQ5534E: User ID 'admin' authentication failed

EXPLANATION:

The user ID and password supplied by the 'app.MQFirstAppApplication' program could not be authenticated.

Additional information: 'Pipe returned 2292 [FAILED]'.

ACTION:

Ensure that the correct user ID and password are provided by the application.

Ensure that the authentication repository is correctly configured. Look at previous error messages for any additional information.

----- amqzfuca.c : 4806 -----

```

12/17/22 03:06:27 - Process(265028.20) User(mqm) Program(amqzlaa0)
Host(gruffest1.fyre.ibm.com) Installation(Installation1)
VRMF(9.2.0.3) QMgr(ECLIPSE.TO.LINUX)
Time(2022-12-17T11:06:27.899Z)
CommentInsert1(admin)
CommentInsert2(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CommentInsert3(CHCKCLNT(REQDADM))

```

AMQ5542I: The failed authentication check was caused by the queue manager

CONNAUTH CHCKCLNT(REQDADM) configuration.

EXPLANATION:

The user ID 'admin' and its password were checked because the queue manager connection authority (CONNAUTH) configuration refers to an authentication information (AUTHINFO) object named 'SYSTEM.DEFAULT.AUTHINFO.IDPWOS' with CHCKCLNT(REQDADM).

This message accompanies a previous error to clarify the reason for the user ID and password check.

ACTION:

Refer to the previous error for more information.

Ensure that a password is specified by the client application and that the password is correct for the user ID. The authentication configuration of the queue manager connection determines the user ID repository. For example, the local operating system user database or an LDAP server.

Note that if the authentication configuration specifies an LDAP user repository, a CHCKCLNT value of REQDADM is treated as equivalent to a value of OPTIONAL.

If the CHCKCLNT setting is OPTIONAL, the authentication check can be avoided by not passing a user ID across the channel. For example, by omitting the MQCSP structure from the client MQCONN API call.

To avoid the authentication check, you can amend the authentication configuration of the queue manager connection, but you should generally not allow unauthenticated remote access.

----- amqzfuca.c : 4829 -----

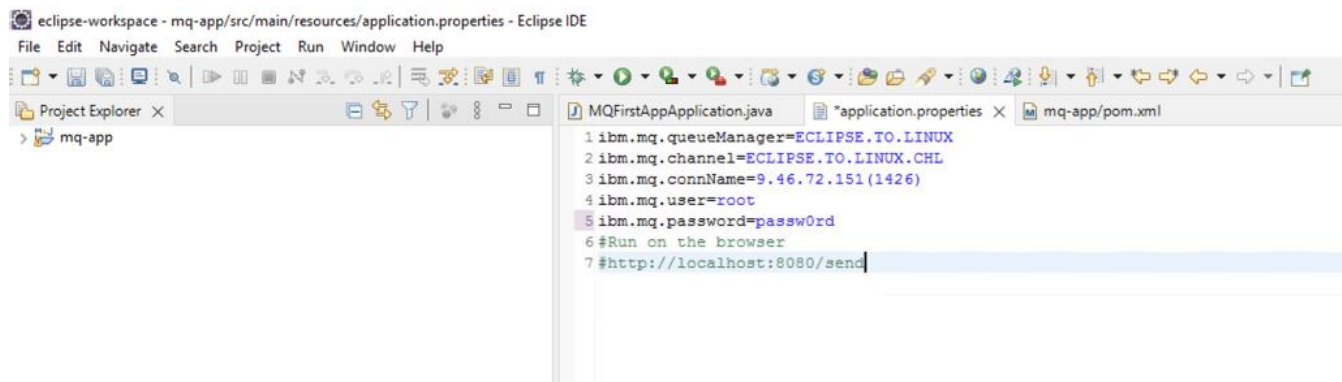
SCENARIO 10: ADD 'ROOT' USER IN 'MQM' GROUP (IN LINUX) AND RUN JAVA PROGRAM AGAIN (IN WINDOWS)

Changes in Linux (MQ Server):

Add root in the 'mqm' group:

```
[root@gruffest1 errors]# sudo usermod -a -G mqm root
[root@gruffest1 errors]# id root
uid=0(root) gid=0(root) groups=0(root),501(mqm)
```

Changes in Windows (mq-client):



Run the Java program again.

The program fails again. Check the queue manager error logs.

The following error is reported in MQ queue manager error logs (/var/mqm/qmgrs/ECLIPSE!TO!LINUX/errors):

```
----- amqzfuca.c : 4829 -----
12/17/22 08:20:09 - Process(265028.21) User(mqm) Program(amqzlaa0)
Host(gruffest1.fyre.ibm.com) Installation(Installation1)
VRMF(9.2.0.3) QMgr(ECLIPSE.TO.LINUX)
Time(2022-12-17T16:20:09.600Z)
CommentInsert1(root)
CommentInsert2(ECLIPSE.TO.LINUX [qmgr])
CommentInsert3(connect)
AMQ8077W: Entity 'root' has insufficient authority to access object
ECLIPSE.TO.LINUX [qmgr].
EXPLANATION:
The specified entity is not authorized to access the required object. The
following requested permissions are unauthorized: connect
ACTION:
Ensure that the correct level of authority has been set for this entity against
the required object, or ensure that the entity is a member of a privileged
group.
----- amqzfuca.c : 1637 -----
12/17/22 08:20:09 - Process(265875.6) User(mqm) Program(amqrmppa)
Host(gruffest1.fyre.ibm.com) Installation(Installation1)
VRMF(9.2.0.3) QMgr(ECLIPSE.TO.LINUX)
Time(2022-12-17T16:20:09.600Z)
ArithInsert1(2) ArithInsert2(2035)
```


CommentInsert1(root)
CommentInsert2(root)
CommentInsert3(root)

AMQ9557E: Queue Manager User ID initialization failed for 'root'.

EXPLANATION:

The call to initialize the User ID 'root' failed with CompCode 2 and Reason 2035. If an MQCSP block was used, the User ID in the MQCSP block was 'root'. If a userID flow was used, the User ID in the UID header was 'root' and any CHLAUTH rules applied prior to user adoption were evaluated case-sensitively against this value.

ACTION:

Correct the error and try again.

----- cmqxsrvc : 2556 -----

Reason for the above error:

[mqm@gruffest1 errors]\$ **dspmqaout -m ECLIPSE.TO.LINUX -t qmgr -p root**

Entity root has the following authorizations for object ECLIPSE.TO.LINUX:

The above output shows that the 'root' user is not authorized to access the objects ECLIPSE.TO.LINUX.

SCENARIO 11: GIVE PERMISSIONS TO THE 'ROOT' USER (IN LINUX) AND RUN JAVA PROGRAM

Changes in Linux:

Give permissions to the root user:

```
setmqaut -m ECLIPSE.TO.LINUX -t qmgr -p root +all
```

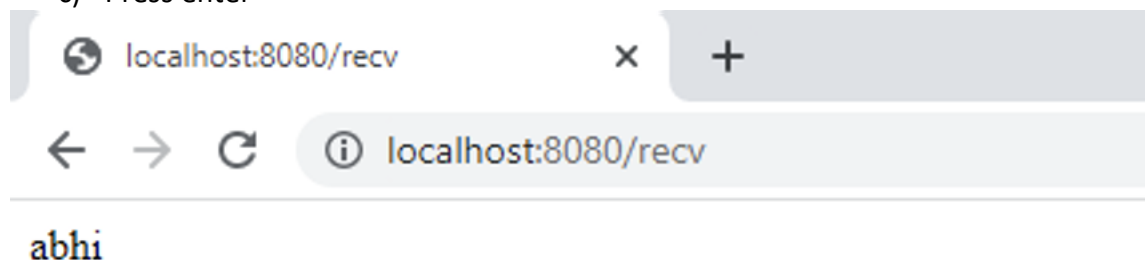
```
setmqaut -m ECLIPSE.TO.LINUX -n QUEUE.LOCAL -t q -p root +all
```

In Windows:

Run the java code and it will work.

Get message from browser as suggested below:

- a) Open any browser.
- b) Type: `http://localhost:8080/recv`
- c) Press enter



In Linux (MQ Server):

Get message from queue in MQ server:

```
[mqm@gruffest1 bin]$ ./amqsget QUEUE.LOCAL ECLIPSE.TO.LINUX  
Sample AMQSGET0 start
```

message <RFH >
no more messages
Sample AMQSGETO end

Working as expected!!!

+++ end +++