**IBM MQ**

**Configuring SSL between two Queue Manager with two way communications**

SSL ( Secure Socket Layer ) – protocol design to allow transmission of secure data over a network.

SLL – uses hash functions to enable the detection of tampering and uses digital certificates to enabled authentications of partners.

SSL – can be established on channels In IBM MQ since v 5.3.x or via MQI channels between client and Managers

Establishing SSL connection between the MQ’s with two way communications on Windows server usig MQ V9.

Create both TEST1 and TEST2 Queue Manager for distribution setup and setup SSL on both QM’s

1. Make one way connectivity from TEST1 to TEST2 then followed by TEST2 to TEST1

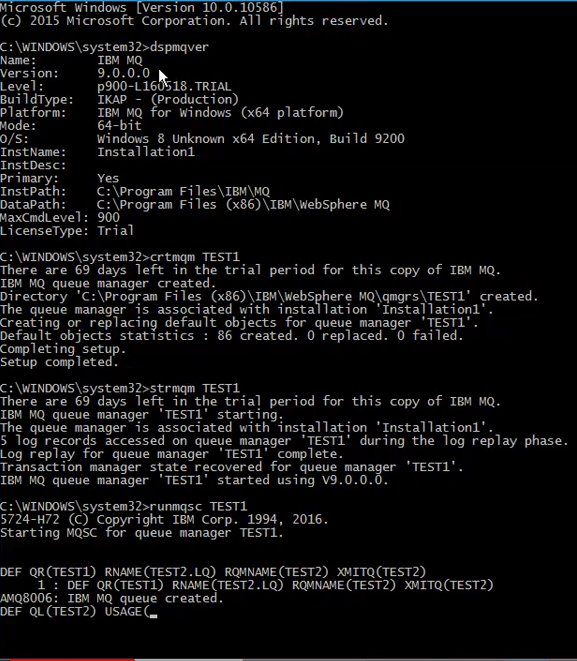
One-way connectivity between 2 Queue Manager TEST1(Source) and TEST2(Destination).

**TEST1: Source**

-dspmqver

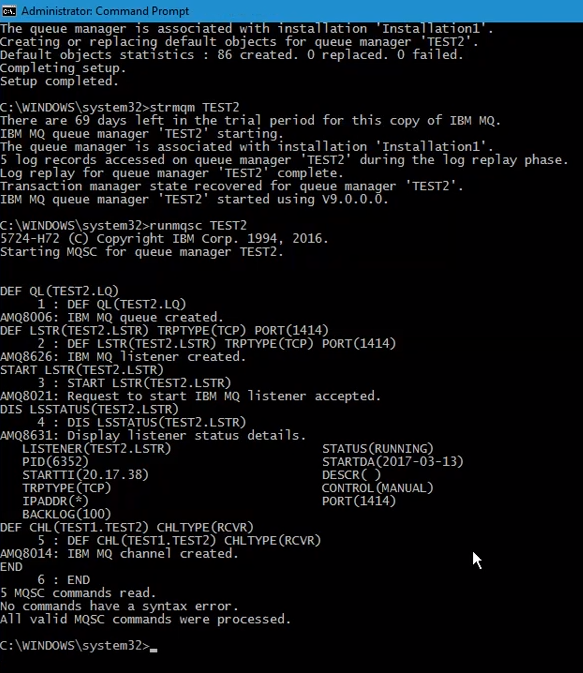
* Crtmqm TEST1
* Strmqm TEST1
* Runmqsc TEST1
* DEFINE QR(TEST1) RNAME(TEST2.LQ) RQMNAME(TEST2) XMITQ(TEST2)
* DEFINE QL(TEST2) USAGE(XMITQ)
* DEF CHL(TEST1.TEST2) CHLTYPE(SDR) CONNAME(‘LOCALHOST(1414) XMITQ(TEST2)
* END

Actual:



**TEST2 : Destination**

* Crtmqm TEST2
* Strmqm TEST2
* Runmqsc TEST2
* DEFINE QL(TEST2.LQ)
* DEFINE LSTR(TEST2.LSTR) TRTYPE(TCP) PORT(1414) CONTROL(QMGR)
* START LSTR(TEST2.LSTR)
* DIS LSTR(TEST2.LSTR)
* DEF CHL(TEST1.TEST2) CHLTYPE(RCVR)
* END



**TESTING THE Connectivity :**

* Start the channel on TEST1 (source)

START CHL(TEST1.TEST2)

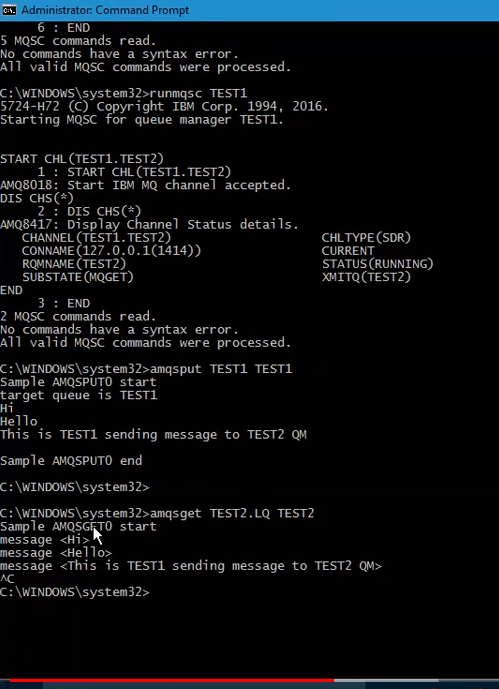
* Put the message from the source remote queue and try to get the message from TEST2.LQ

Of destination queue manager TEST2

Amqsput TEST1 TEST1

Amqsput TEST2.LQ TEST2

* If able to received means connectivity has been established



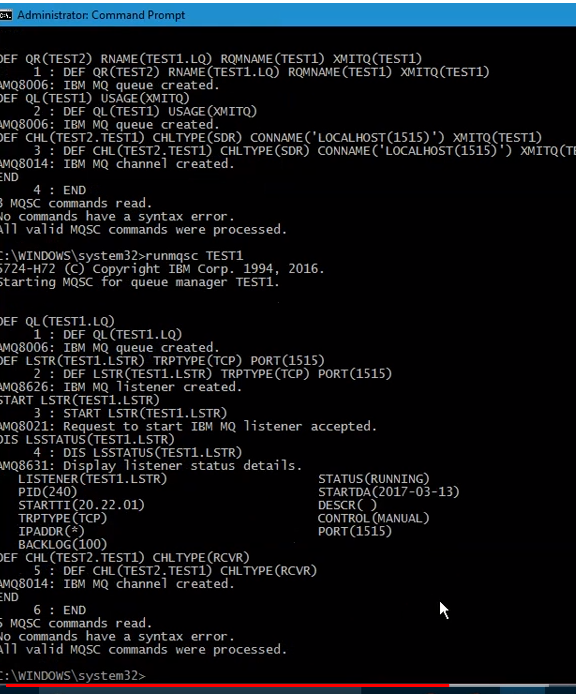
**Now configure TEST2 to TEST1 connectivitiy**

TEST2 : Source

* Runmqsc TEST2
* DEFINE QR(TEST2) RNAME(TEST1.LQ) RQMNAME(TEST1) XMITQ(TEST1)
* DEFINE QL(TEST1) USAGE(XMITQ)
* DEF CHL(TEST2.TEST1) CHLTYPE(SDR) CONNAME(‘LOCALHOST(1515)’) XMITQ(TEST1)
* END

TEST1 : Destination

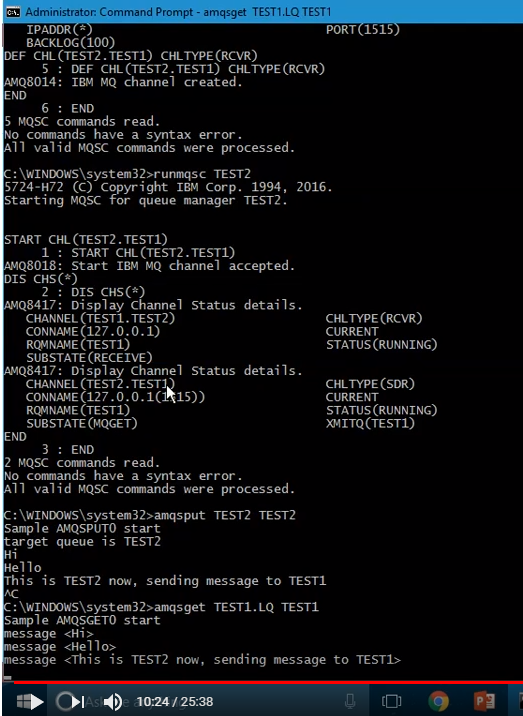
* Runmqsc TEST1
* DEFINE QL(TEST1.LQ)
* DEFINE LSTR(TEST1.LSTR) TRTYPE(TCP) PORT(1515) CONTROL(QMGR)
* START LSTR(TEST1.LSTR)
* DEF CHL(TEST2.TEST1) CHLTYPE(RCVR)
* END



Perform connectivity test from TEST 2 to TEST1

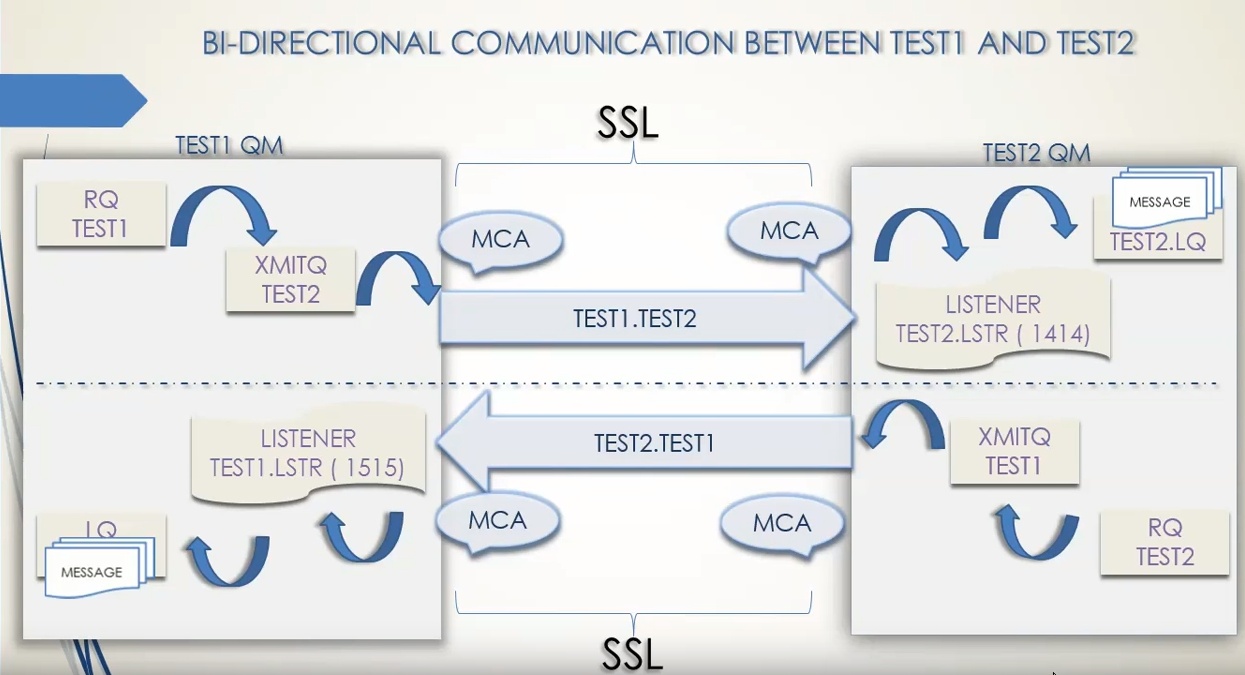
Amqsput TEST2 TEST2

Amqsget TEST1.LQ.TEST1



MQ Works : Enable SSL by design schematics

BI-DIRECTIONAL COMMUNICATIONS BETWEEN TEST1 AND TEST2

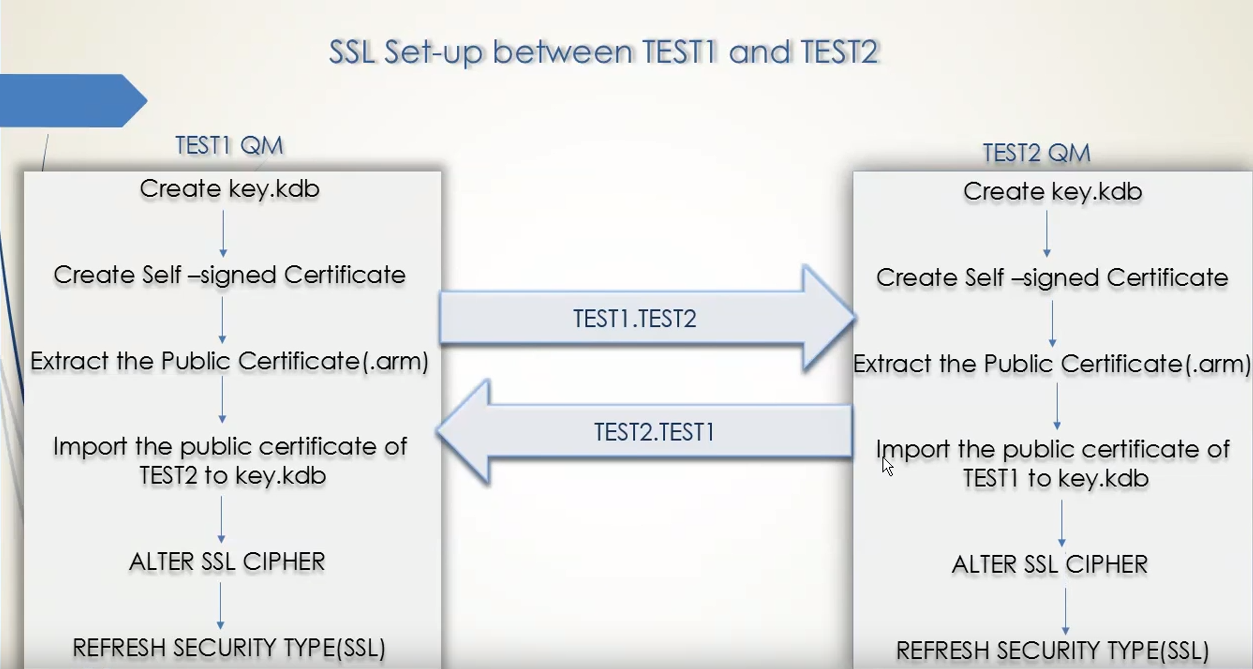


Configure SSL between the Queue Managers. For successful SSL handshake to be take place.

Requirements

* First we have to create key.kdb
* Second we have to create the self-signed certificate which is the private certificate of the queue manager.
* Extract the public certificate from key.kdb which would be test.arm and test2.arm
* Exchanging and adding the public certificate of TEST1 to TEST2.kdb and vice-versa
* Once the certificate are imported, alter the SSL CIPHER on both Channels. We can use any one of the IBM Default provided CIPHER suites (Algorithms), but it should be same on both SDR and RCVR channels.
* Refresh the SSL security on both the Queue Manager which makes the SSL changes to take effect.

SSL Set-up between TEST1 and TEST2



Creating the key.kdb of TEST1 Queue Manager

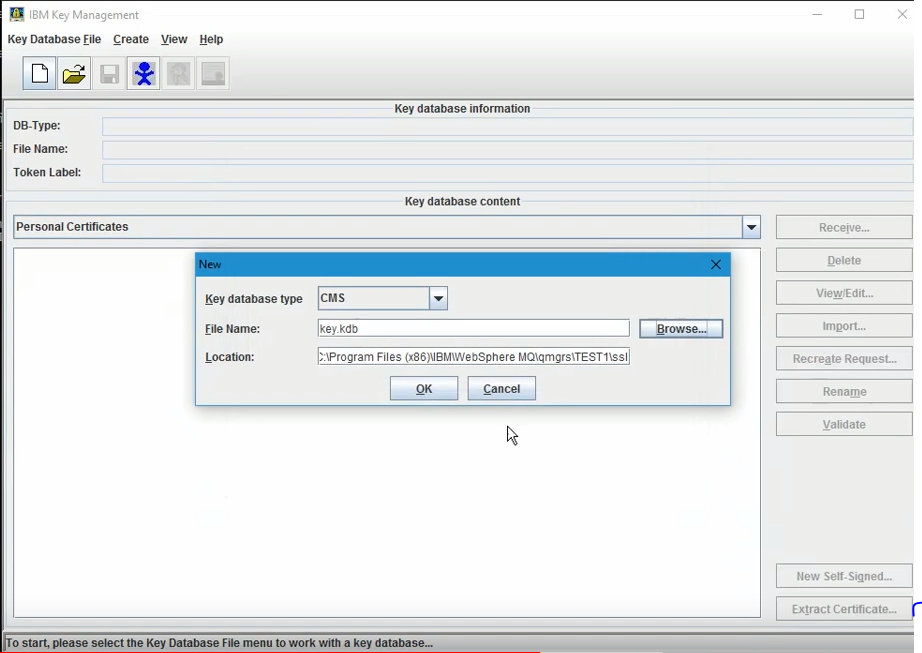
1. Create key.kdb of TEST1 Queue Manager
2. Create self-Signed Certificate of TEST1
3. Extract the public certificate TEST1.ARM

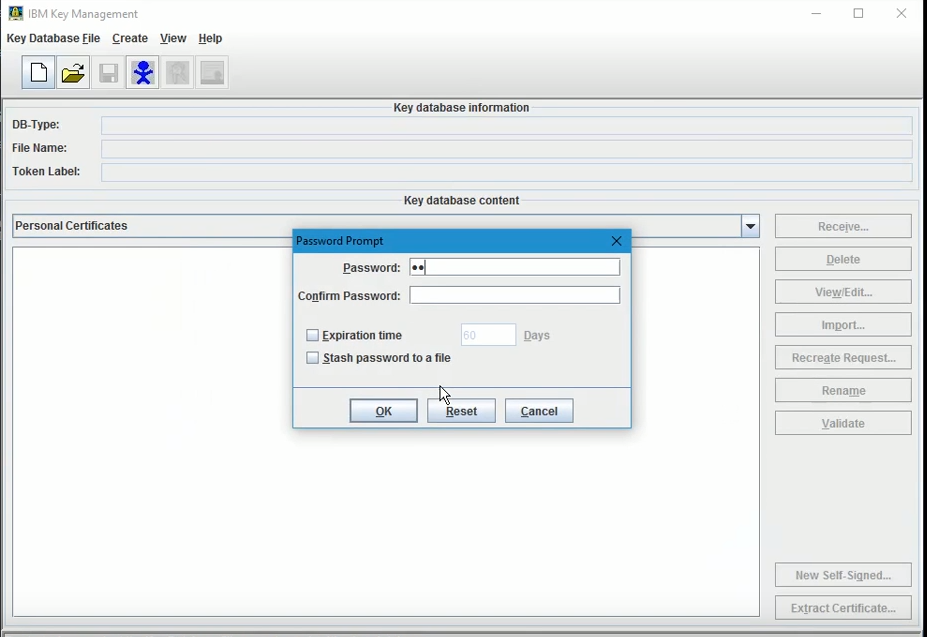
* Strmqikm 🡪 open the IBM key management

File New

Key.kdb

Browse 🡪 qmrs folder🡪 TEST1 -> ssl 🡪 key.kdb 🡪Save

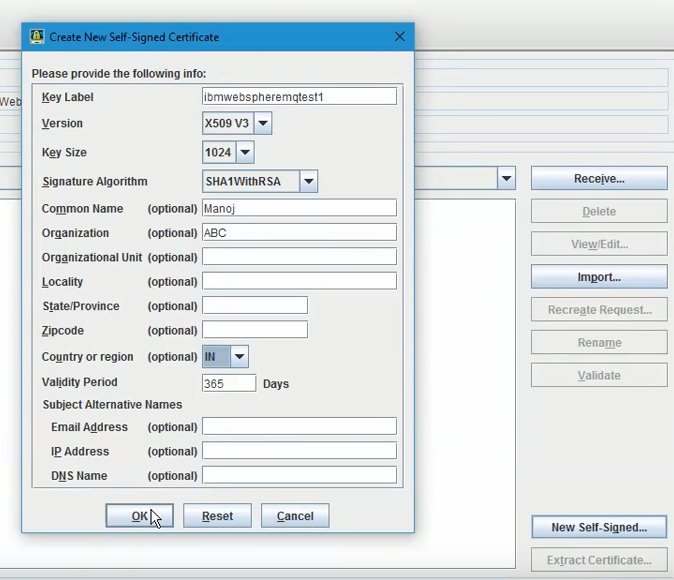




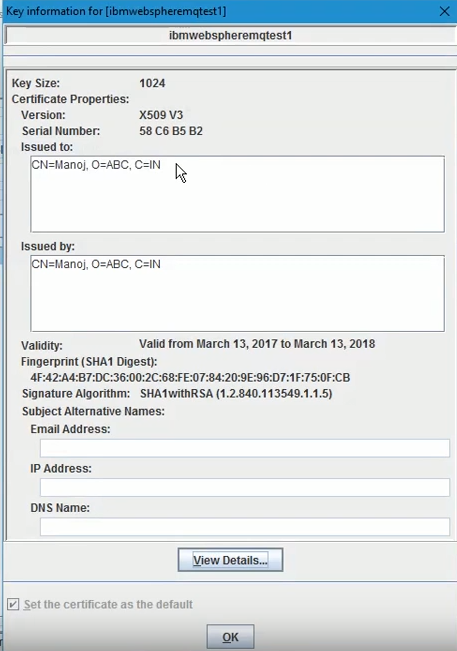


Next creating Self-Signed Certificate of TEST1

Click the New Signed

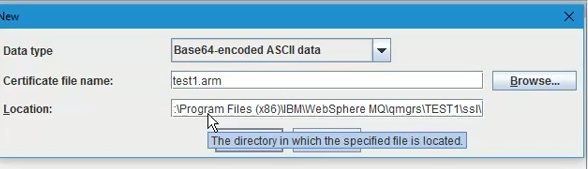


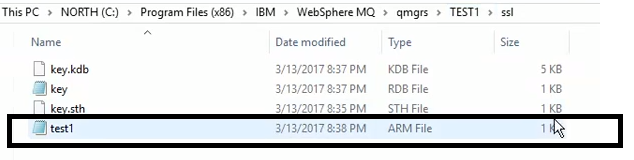
View Details



Extracting the public certificate

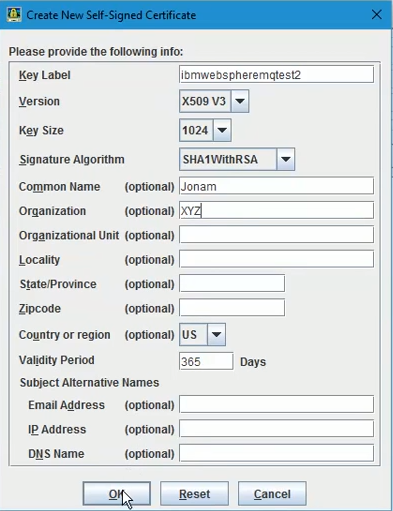
Click Extract Certificate





REPEAT THE 3 Steps for TEST2 QM

1. Create key.kdb of TEST2 Queue Manager
2. Create self-Signed Certificate of TEST2
3. Extract the public certificate TEST2.ARM

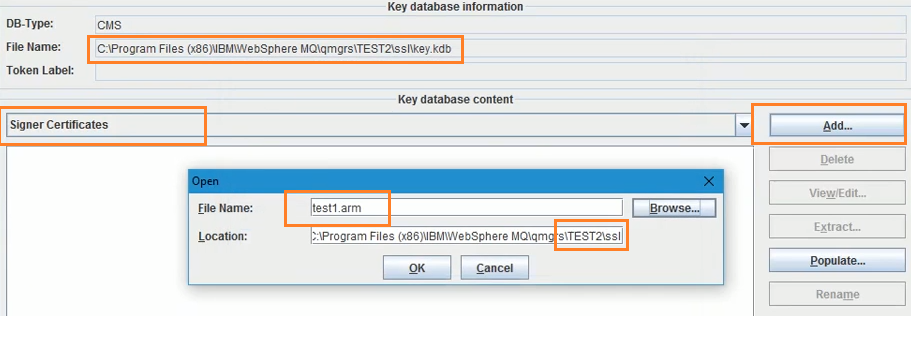


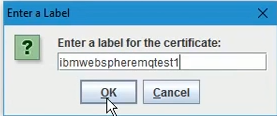


Exchanging the .ARM files with both QM’s

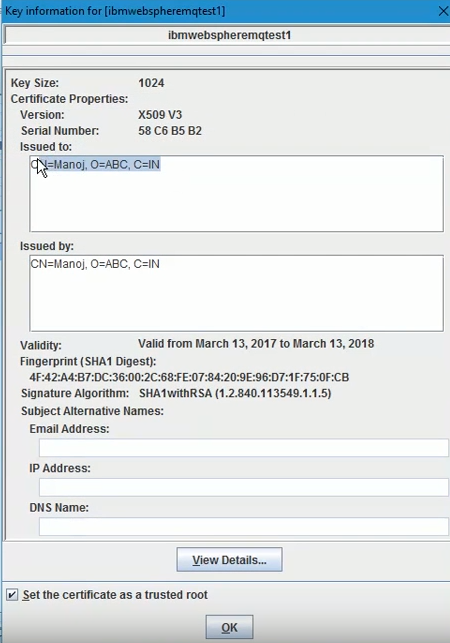
* (copy the test1.arm and place it in TEST2 QM’SSL folder and vice versa)
* Import the .ARM files to their partners key.kdb

TEST2 first

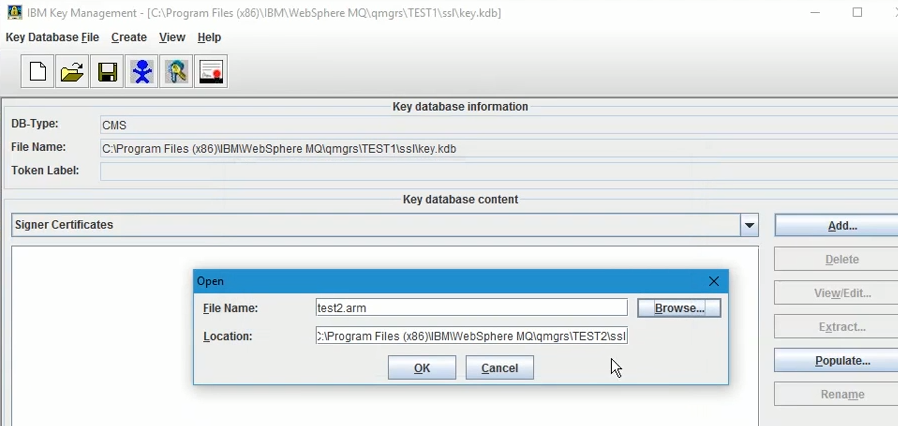


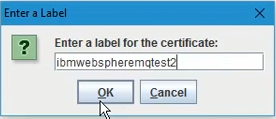


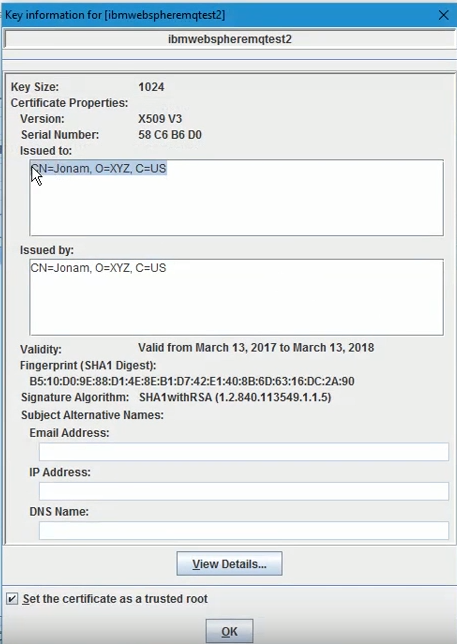
View



TEST1 Open test1\ssl\key.kdb







After completed follow the command

TEST1 :

ALTER CHANNEL (TEST1.TEST2) CHLTYPE(SDR) SSLCIPH(TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA)

ALTER CHANNEL (TEST2.TEST1) CHLTYPE(RCVR) SSLCIPH(TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA)

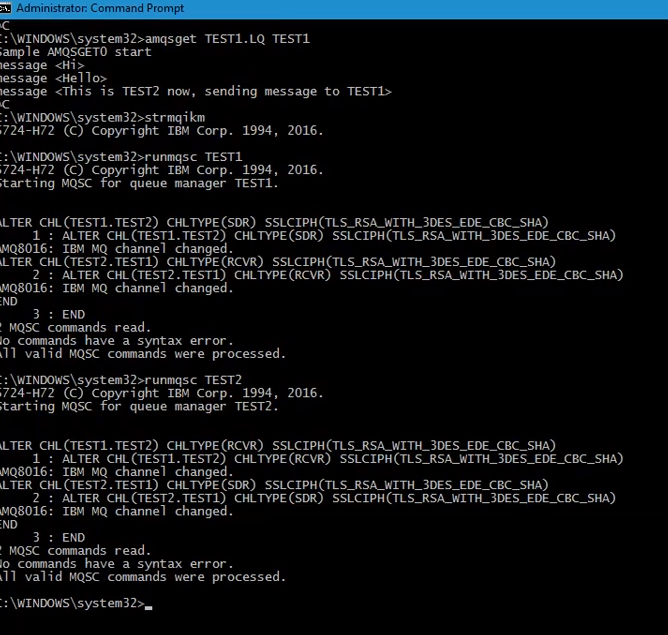
TEST2 :

ALTER CHANNEL (TEST1.TEST2) CHLTYPE(RCVR) SSLCIPH(TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA)

ALTER CHANNEL (TEST2.TEST1) CHLTYPE(SDR) SSLCIPH(TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA)

REFRESH the security on both MQ

REFRESH SECURITY TYPE(SSL)



REFRESH SECURITY TYPE (SSL)

Once the security is refreshed check the channel status on both sides which would be in running state

DIS CHS(TEST1.TEST2)

DIS CHS(TEST2.TEST)

