1 Connecting two Windows Queue Managers using SSL

1.1 Abstract

This article presents a step-by-step guide to configuring two WebSphere MQ Version 6[1] queue managers on Windows for communication using SSL channels.

I assume that you are familiar with SSL in general and know how to set up non-SSL sender/receiver channels between two queue managers.

For more information about SSL with MQ, refer to the WebSphere MQ V6 Security manual, SC34-6588. You can download the PDF from:

http://www.ibm.com/software/integration/wmq/library/

1.2 Basic Configuration

We need two queue managers with a working connection (sender/receiver channel pairs in both directions). I will use the following names and attributes; you can create queue managers with the same names, or adjust the instructions below to match your configuration:

Queue Manager name	QM1	QM2
IP address	192.168.1.65	192.168.1.64
Listener port	11111	22222
Transmit queue	QM2	QM1
Sender channel	QM1.QM2	QM2.QM1
Receiver channel	QM2.QM1	QM1.QM2
Local queue (for testing)	Q1	Q2
Remote queue definition (for testing)	QM2.Q2	QM1.Q1
MQ Installation directory (throughout this document, <MQdir>)	C:\MQV6	C:\MQV6

[1] Referred to as "MQ" from now on.

If you choose to create two queue managers as above, the only thing you'll need to change is the IP address. You can create the two queue managers on the same, or on separate, Windows systems. Skip to Checking the channel, below, if you already have two interconnected queue managers.

Creating the Queue Managers

You can use the Version 6 MQ Explorer to create the queue managers, or use a command script. The following example creates a queue manager called `QM1` with a listener on port 11111:

```
@echo Create queue manager
crtmqm -u QM1.DLQ QM1


@echo Start queue manager and associated services
amqmdain qmgr start QM1


@echo Create and start listener
@echo def listener('LISTENER.TCP') trptype(tcp) port(11111)
control(qmgr) | runmqsc QM1
@echo START LISTENER('LISTENER.TCP') | runmqsc QM1


@echo Create dead letter queue
@echo def ql(QM1.DLQ) replace | runmqsc QM1
```

The example above shows how to create QM1; you can adapt it to create QM2.

Setting up the channels

The following commands, when run from a command prompt on the machine where QM1 *is running*, create the necessary MQ objects for QM1 to communicate with QM2:

```
echo def ql(QM2) replace usage(xmitq) trigger trigdata(QM1.QM2)
initq(SYSTEM.CHANNEL.INITQ) | runmqsc QM1


echo def chl(QM1.QM2) chltype(sdr) replace xmitq(QM2)
conname('192.168.1.64(22222)') | runmqsc QM1


echo def chl(QM2.QM1) chltype(rcvr) replace | runmqsc QM1
```

```
@rem Create queues for test
echo def ql(Q1) replace | runmqsc QM1
echo def qr(QM2.Q2) replace rname(Q2) rqmname(QM2) | runmqsc QM1
```

Similarly, these commands (from a command prompt on the machine where QM2 is running) create the objects that QM2 needs to communicate with QM1:

```
echo def ql(QM1) replace usage(xmitq) trigger trigdata(QM2.QM1)
initq(SYSTEM.CHANNEL.INITQ) | runmqsc QM2

echo def chl(QM2.QM1) chltype(sdr) replace xmitq(QM1)
conname('192.168.1.65(11111)') | runmqsc QM2

echo def chl(QM1.QM2) chltype(rcvr) replace | runmqsc QM2

@rem Create queues for test
echo def ql(Q2) replace | runmqsc QM2
echo def qr(QM1.Q1) replace rname(Q1) rqmname(QM1) | runmqsc QM2
```

1.3 Checking the channels

Before proceeding, open a command prompt and check that the channels you intend to use with SSL (in our example configuration QM1.QM2 and QM2.QM1) run correctly. The example below assumes that the channels are already running, or the transmission queue is triggered:

Test	Machine	Run

QM1 to QM2	Same as QM1	`C:\>`**`amqsput QM2.Q2 QM1`**
		`Sample AMQSPUT0 start`
		`target queue is QM2.Q2`
		`test msg 1`
		
		`Sample AMQSPUT0 end`
		
		`C:\>`

	Same as QM2	`C:\>`**`amqsget Q2 QM2`**
		`Sample AMQSGET0 start`
		`message <test msg 1>`
		`no more messages`
		`Sample AMQSGET0 end`
		
		`C:\>`

QM2 to QM1	Same as QM2	`C:\>`**`amqsput QM1.Q1 QM2`**
		`Sample AMQSPUT0 start`
		`target queue is QM1.Q1`
		`test msg 2`
		
		`Sample AMQSPUT0 end`
		
		`C:\>`

	Same as QM1	`C:\>`**`amqsget Q1 QM1`**
		`Sample AMQSGET0 start`
		`message <test msg 2>`
		`no more messages`
		`Sample AMQSGET0 end`
		
		`C:\>`

With both queue managers and their channels up and running, we are ready to set up an SSL connection.

1.4 SSL: the very basics

In the SSL protocol, the party that starts a conversation (in this case, the MQ sender channel) is the *SSL client*. The other party (MQ receiver channel) is the SSL server.

The SSL client (sender channel) authenticates the server by requesting the server's certificate. This is sometimes called *one-way authentication*. Optionally, the server (receiver channel) may require client authentication (this is mutual, or two-way, authentication).

In MQ, most customers using SSL channels will probably set them up to request mutual authentication. In this example we will set up one-way authentication first, and then mutual authentication.

Incidentally, one-way authentication is what happens when you shop online: your browser, an SSL client, receives a certificate from the online shop, so you know it is safe to give them your credit card, but the shop does not request a certificate from you.

When we start a sender channel (say, QM2.QM1), this is what happens (this is called SSL handshake):

QM2 starts the connection and requests a certificate.

QM1 sends its certificate. This is encrypted ("signed") using the Certification Authority certificate—more about this below) .

QM2 verifies QM1's digital signature in the certificate. QM2 now knows QM1 is who it claims to be.

If mutual authentication is required, QM2 sends its certificate to QM1.

The handshake continues with the selection of a secret key that both parties can use to sign and/or encrypt messages.

From the list above, it follows that:

1. The party being authenticated must have a certificate. This is called a "Personal Certificate".
2. The authenticating party must be able to decipher the certificate's signature: it must have the Certification Authority's certificate used to sign the other party's Personal certificate.

1.5 Process Overview

We will follow this process to establish an SSL connection between QM1 and QM2:

1. Create a key repository for each queue manager.
2. Obtain a certificate for each queue manager.
3. Install the certificates in the key repositories.
4. Set up the channels for SSL authentication and test.

1.6 Step-by-step instructions

Create a key repository for each queue manager

The instructions below show how to create a key repository for queue manager QM1; you need to repeat these steps for QM2.

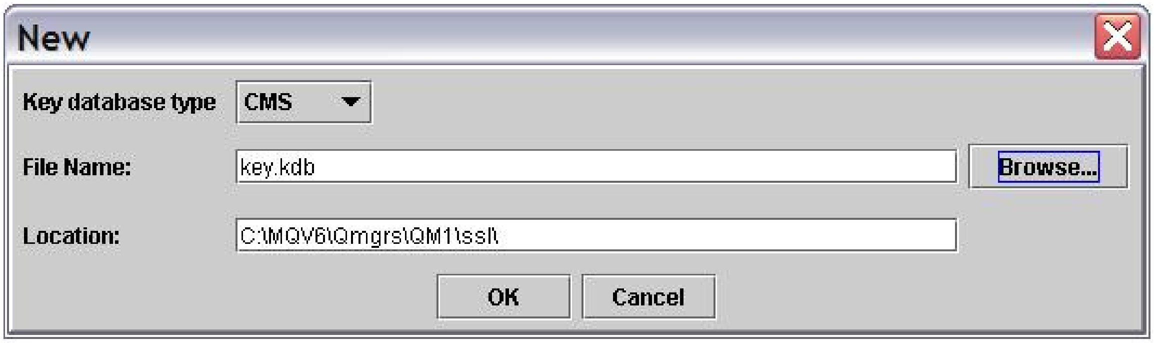
Open a Windows Command prompt and enter `strmqikm`. This starts the IBM Key Management (iKeyMan) GUI.

Create a key repository for the queue manager:

Select `Key Database File` ⇨ `New` and create a repository as follows:

- Key Database Type: `CMS`
- File name: `key.kdb`
- Location: `<MQdir>\Qmgrs\QM1\ssl`. In this example: `C:\MQV6\Qmgrs\QM1\ssl`

You should see this:



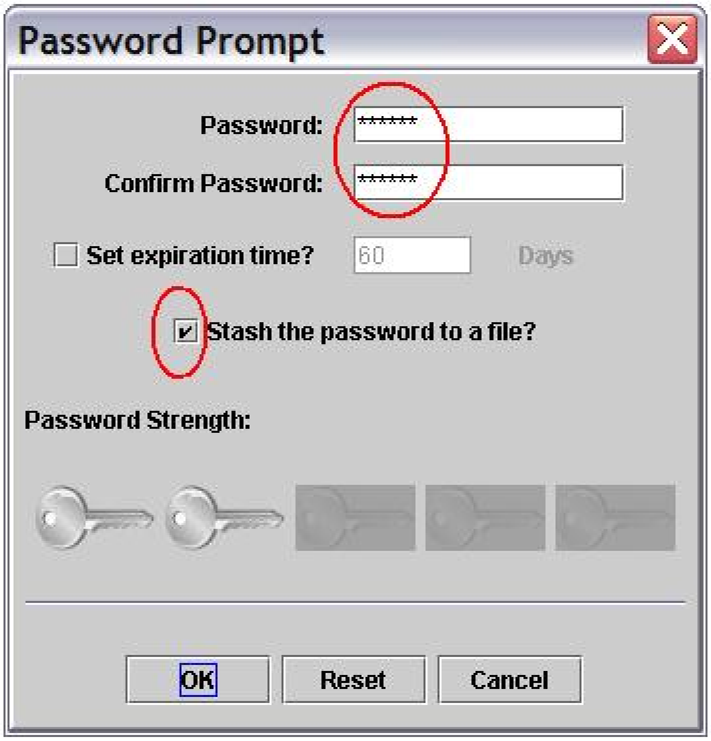
New

Key database type: CMS

File Name: key.kdb

Location: C:\MQV6\Qmgrs\QM1\ssl\

Click OK. Then **enter a password** (and remember it; you will need it!) and **tick** `Stash the password to a file?`



Password Prompt

Password: ******

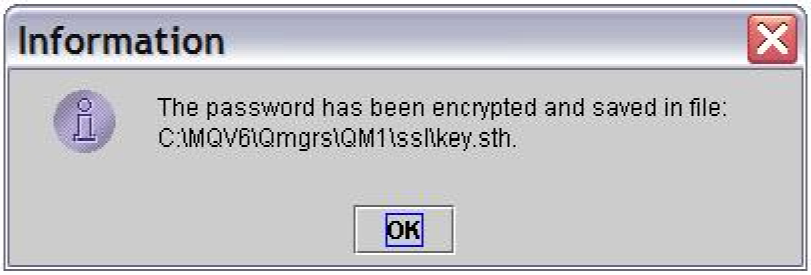
Confirm Password: ******

☐ Set expiration time? 60 Days

☑ Stash the password to a file?

Password Strength:

Click OK. You will see this dialog:

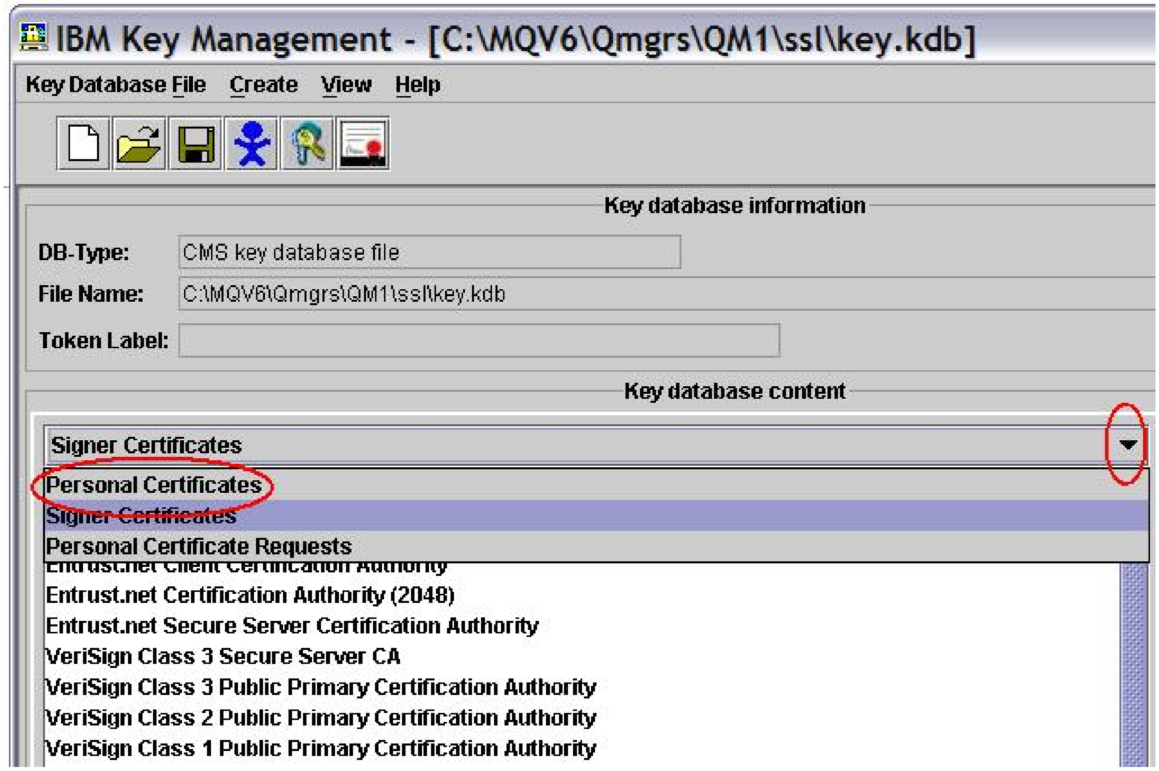


Information

The password has been encrypted and saved in file:
C:\MQV6\Qmgrs\QM1\ssl\key.sth.

Click OK. You have created a key repository for queue manager QM1.

After creating the key repository, the GUI shows the installed Certification Authority certificates provided with iKeyMan. **Use the pull-down (top right) to switch to viewing Personal Certificates**:



IBM Key Management - [C:\MQV6\Qmgrs\QM1\ssl\key.kdb]

Key database information

DB-Type: CMS key database file

File Name: C:\MQV6\Qmgrs\QM1\ssl\key.kdb

Token Label:

Key database content

Signer Certificates

Personal Certificates
Signer Certificates
Personal Certificate Requests
Entrust.net Client Certification Authority
Entrust.net Certification Authority (2048)
Entrust.net Secure Server Certification Authority
VeriSign Class 3 Secure Server CA
VeriSign Class 3 Public Primary Certification Authority
VeriSign Class 2 Public Primary Certification Authority
VeriSign Class 1 Public Primary Certification Authority

Keep the iKeyMan GUI open; we will come back to it shortly.

At the machine where queue manager QM2 runs, **repeat the steps above for QM2.**

Obtain a certificate for each queue manager

The instructions below show how to obtain a certificate for queue manager QM1; you need to repeat these steps for QM2.

There are a number of ways to obtain a certificate for your queue manager:

- You can create self-signed certificates.
- You can have an in-house Certification Authority.
- You can request a certificate from a Certification Authority.

The instructions below are for obtaining a demo (valid for 30 days) personal certificate from globalsign.com. There are other sites for requesting certificates (for example Thawte, or VeriSign); GlobalSign is convenient because it does not require registration. Note that certificates for purposes other than a demo will cost money (dispensing certificates is what Certification Authorities do for a living).

To obtain a certificate:

Open Internet Explorer and go to http://www.globalsign.com.

Select “Buy Certificates”; this opens a list. From the list, select “Personal Certificates”. This should open:

http://www.globalsign.com/digital_certificate/personalsign/index.cfm

Select **PersonalSign Demo** (click on the “Get Yours Now!” button). This takes you to a screen showing an 8-step process for obtaining your certificate:

Step	**Comments**
Step 1. CHECK ROOT First, you need to install GlobalSign´s Root Certificate.	This should already be installed.
Step 2. SUBMIT YOUR E-MAIL ADDRESS Submit your e-mail address and provide a password.	This asks your internet e-mail address and a password that you will need in step 4. After you press "go to step 3" GlobalSign will send you an e-mail.
Step 3. CHECK YOUR MAILBOX You will receive an e-mail from GlobalSign in your mailbox. You have to check your mailbox and click on the hyperlink.	You will receive and e-mail from "ca@globalsign.net" within one minute. The e-mail contains a hyperlink -- click on it (**make sure that clicking on the hyperlink invokes the same browser you were using before**).
Step 4. ENTER YOUR PASSWORD Enter the password you provided in step 2.	Enter the password you gave in step 2.
Step 5. PROVIDE PERSONAL DATA Enter some personal information.	Click on "Go to step 6" without making any changes. In particular, **leave “Protect private key” set to “No”.**
Step 6. ACCEPT AGREEMENT Read the subscriber agreement.	Click on "Agree (Go to step 7)"
Step 7. CHECK YOUR MAILBOX You will receive an e-mail from GlobalSign containing a hyperlink. Check your mailbox!	You will receive another e-mail within 5 minutes. It contains a hyperlink that downloads your certificate and opens a browser page with an "Install" button. **Make sure it is the same browser as before.**

Step 8. INSTALL CERTIFICATE When receiving our mail, click on the hyperlink in order to install your certificate.	Click on "Install". Click "OK" to any browser warnings. You should receive a message confirming that your certificate is installed. Click OK.

How do you know the certificate is installed? The following is extracted from the final confirmation screen:

> **Note for Microsoft Internet Explorer Users:**
>
> After having installed your certificate, you can now verify that you OWN a Globalsign Certificate.
>
> Go to the "Tools" menu, select "Internet options", click on the "Content" tab and finally click on "Certificates".
>
> By doing so, you will have opened the certificate manager and you will see a GlobalSign Certificate "issued to" your e-mail address.

Repeat these steps for queue manager QM2 (on the machine where QM2 runs).

Install the certificates in the key repositories

The instructions below show how to install the certificate you just obtained for queue manager QM1; you need to repeat these steps for QM2.

The certificate you have just obtained is accessible from Internet Explorer. To install it for QM1, you need to:

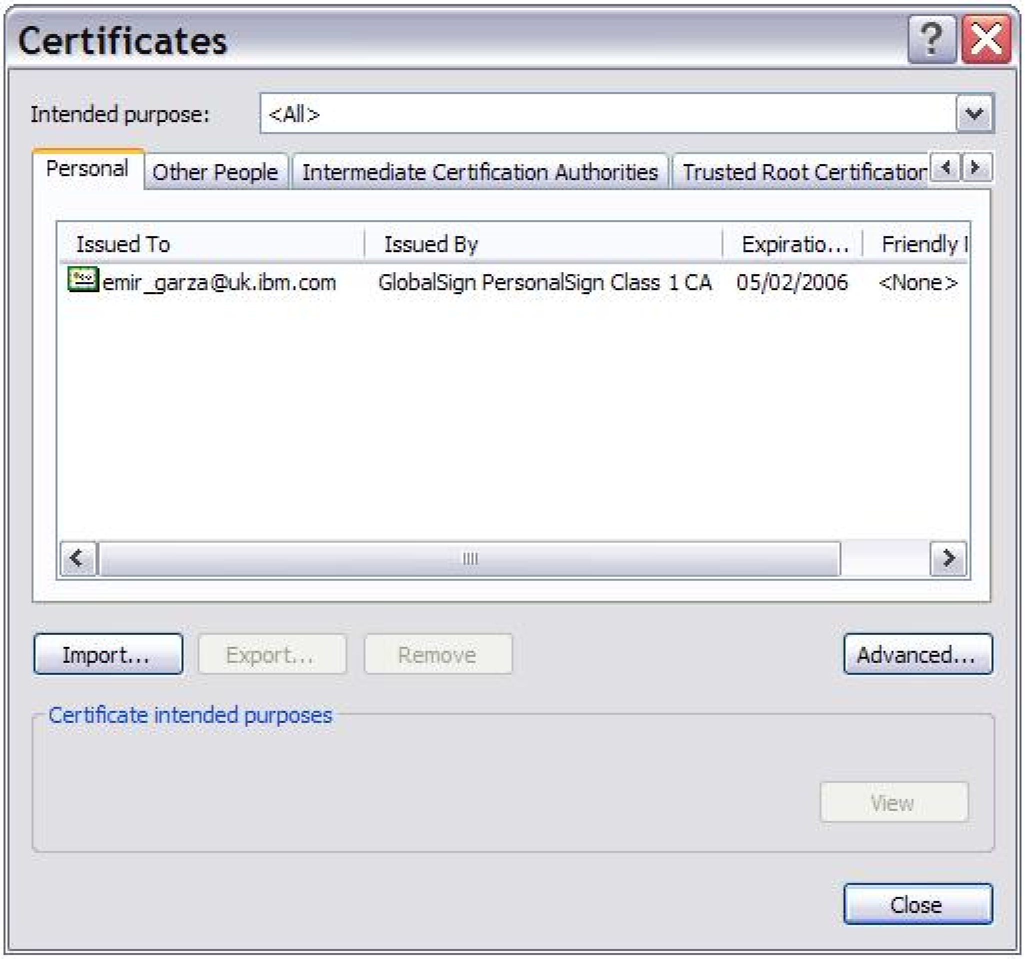
- Export the certificate from Internet Explorer.
- Import the certificate into QM1's key repository.

Export the certificate from Internet Explorer

Open Internet Explorer and **select**:

```
Tools ⇨
      Internet Options … ⇨
            Content ⇨
                  Certificates …
```

You will see the certificate you just obtained and installed:



Certificates
Personal
Other People
Intermediate Certification Authorities
Trusted Root Certification
Import...
Export...
Remove
Advanced...
Certificate intended purposes
View
Close

Intended purpose: <All>

Issued To	Issued By	Expiratio...	Friendly N
emir_garza@uk.ibm.com	GlobalSign PersonalSign Class 1 CA	05/02/2006	<None>

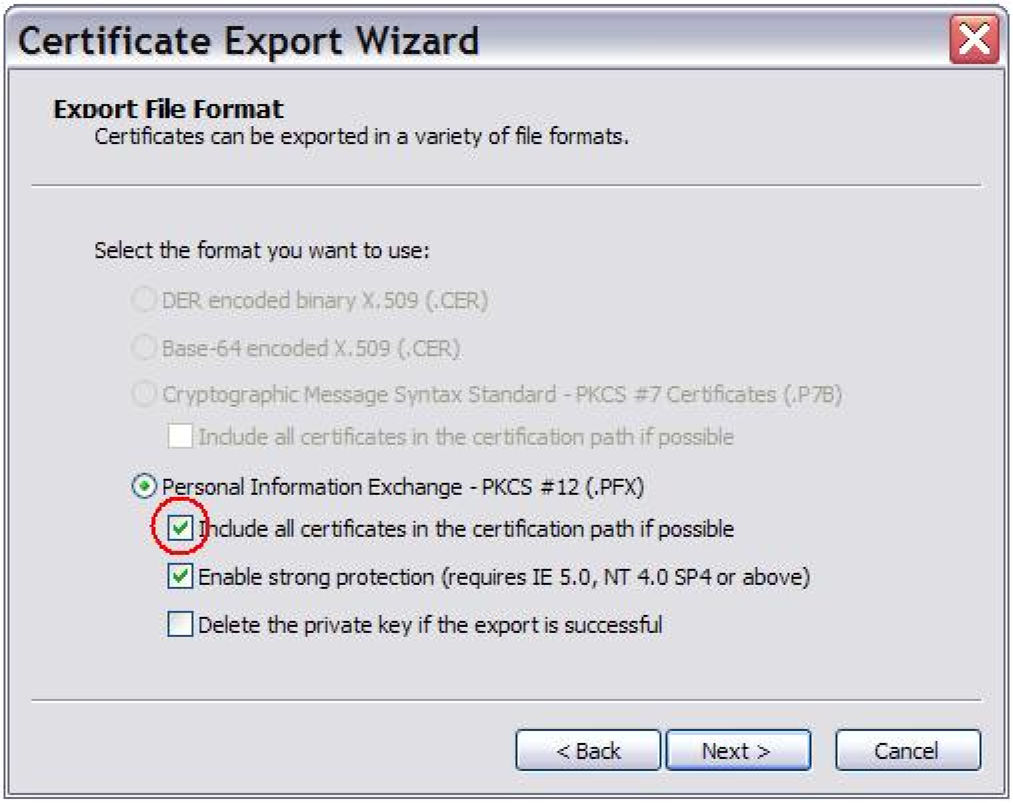
Select (click on) the certificate; then **select Export ...**

This opens the Export Certificate Wizard:

At the Welcome screen, **click** `Next`.

At the Export Private Key dialog, **select** `Yes`.

At the Export File Format dialog, select `Include all certificates` …



Certificate Export Wizard

Export File Format
Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)
○ Base-64 encoded X.509 (.CER)
○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
□ Include all certificates in the certification path if possible
● Personal Information Exchange - PKCS #12 (.PFX)
☑ Include all certificates in the certification path if possible
☑ Enable strong protection (requires IE 5.0, NT 4.0 SP4 or above)
□ Delete the private key if the export is successful

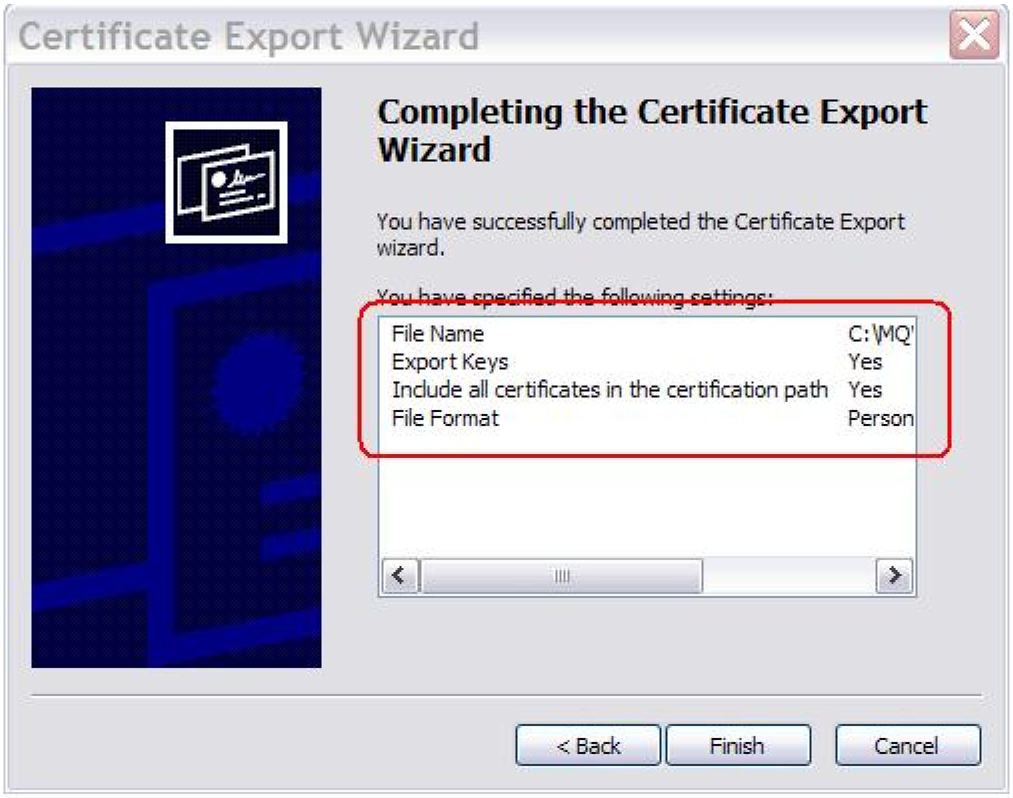
Click `Next`.

At the Password dialog, **enter a password** to protect the exported certificate (you will need it when importing).

At the File to Export dialog,

> **Enter (or navigate to)**:
>
> `<MQdir>\Qmgrs\QM1\ssl\QM1.pfx.`
>
> (In this example: `C:\MQV6\Qmgrs\QM1\ssl\QM1.pfx.`)
>
> **Click** `Next`.

At the completion dialog, **verify the settings:**



Completing the Certificate Export Wizard

You have successfully completed the Certificate Export wizard.

You have specified the following settings:

File Name	C:\MQ'
Export Keys	Yes
Include all certificates in the certification path	Yes
File Format	Person

The settings must be:

File Name:	C:\MQV6\Qmgrs\QM1\ssl\QM1.pfx
Export Keys:	Yes
Include all …:	Yes
File Format:	Personal Information Exchange (*.pfx)

C**lick** Finish. You should see the message "The export was successful".

The next step is to import the certificate into QM1's key repository.

Import the certificate

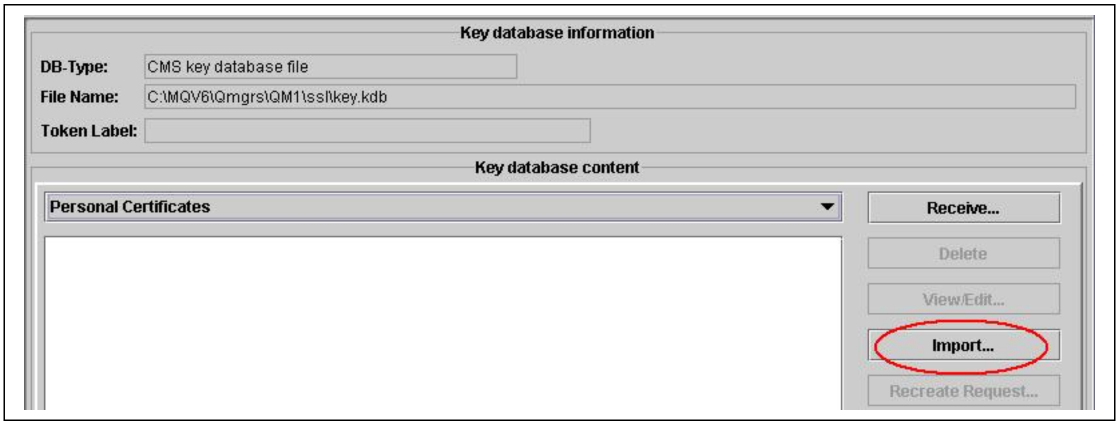
Switch to the iKeyMan GUI, which you left open at the end of step (Create a key repository for each queue manager).

If iKeyMan is closed:

- Enter strmqikm from a command prompt
- Key Database File ⇨
Open ⇨
`<MQdir>\Qmgrs\QM1\ssl\key.kdb`

- Enter the password
- From pull-down, select Personal Certificates

The Personal Certificates pane is empty. **Click on the** `Import` **… button:**



Key database information

DB-Type: CMS key database file

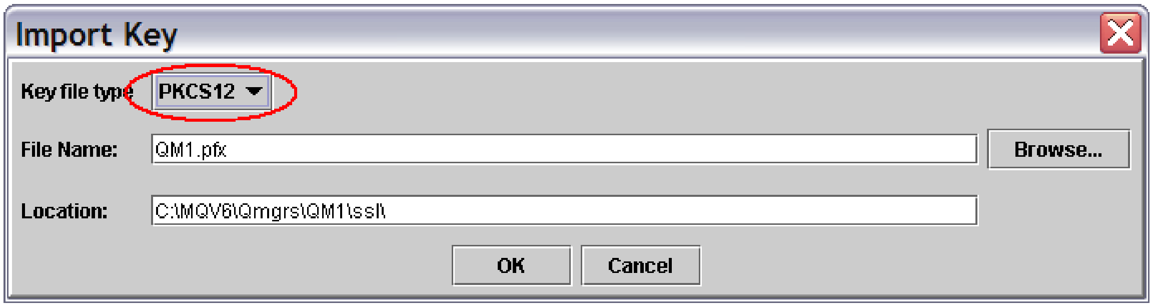
File Name: C:\MQV6\Qmgrs\QM1\ssl\key.kdb

Token Label:

Key database content

Personal Certificates

This opens the Import Key dialog. **Select PKCS12** from the `Key file type` pull-down:



Import Key

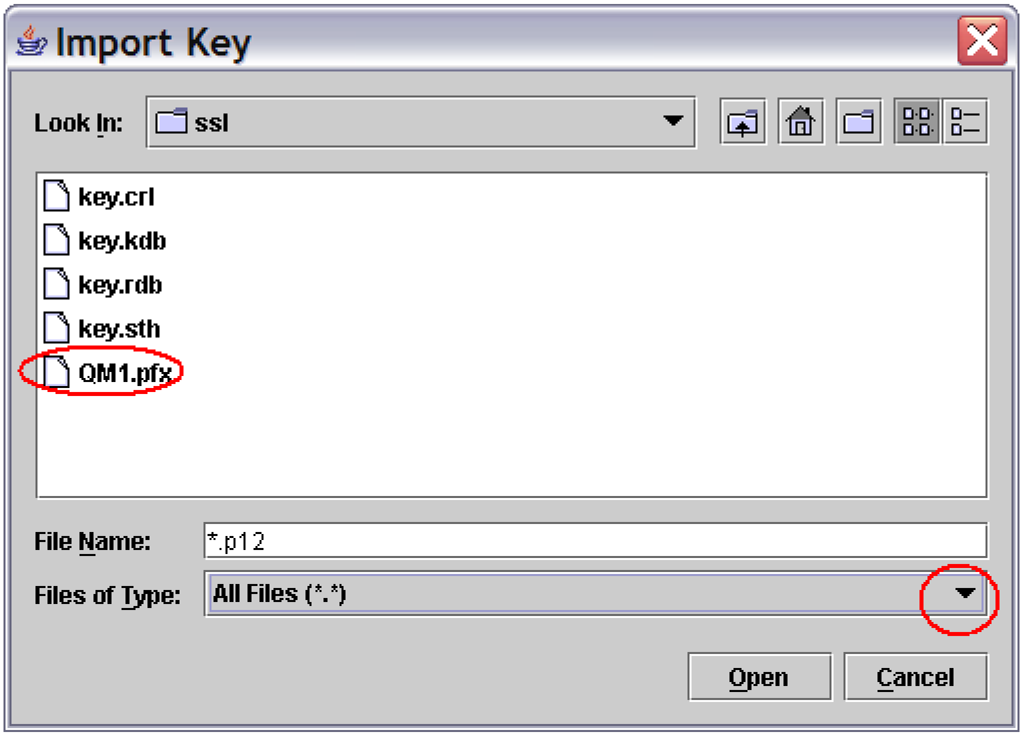
Key file type: PKCS12

File Name: QM1.pfx

Location: C:\MQV6\Qmgrs\QM1\ssl\

Click on the `Browse` **… button**

- **Navigate to** `<MQdir>\Qmgrs\QM1\ssl\`
- **Select** `All files` from the pull-down (see picture, below)
- **Select** `QM1.pfx` (the certificate exported from Internet Explorer)



Look In: ssl

File Name: *.p12

Files of Type: All Files (*.*)

Click Open. This returns to the Import Key dialog.

Click OK.

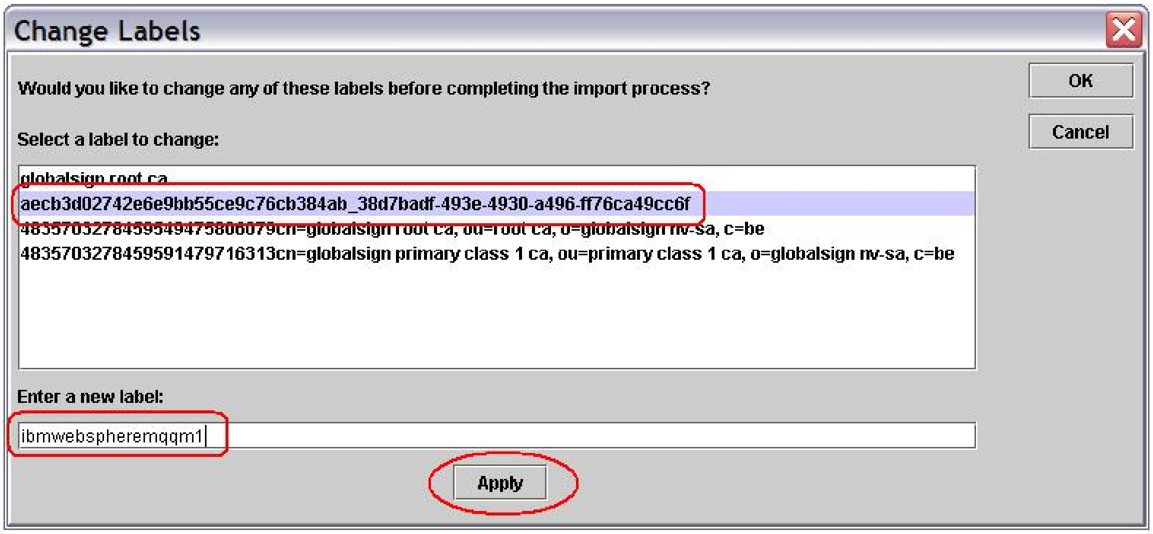
Enter the password you gave when exporting the certificate from Internet Explorer.

The Change labels dialog asks "Would you like to change any of these labels…?" and shows four certificates[2]: three are for the Certification Authority (they all have "globalsign" somewhere in the label) and one is the personal certificate (the label is a hexadecimal string).

Click on the personal certificate (this enables the new label field, at the bottom).

[2] If you see only one certificate, it is because you did not select `Include all certificates` when exporting the certificate from Internet Explorer. The import may work, but if it doesn't, repeat the export process, this time including all certificates.

Enter the label. This must be `ibmwebspheremq` followed by queue manager name, *all in lowercase*: `ibmwebspheremqqm1`.



Would you like to change any of these labels before completing the import process?

Select a label to change:

globalsign root ca
aecb3d02742e6e9bb55ce9c76cb384ab_38d7badf-493e-4930-a496-ff76ca49cc6f
4835703278459549475806079cn=globalsign root ca, ou=root ca, o=globalsign nv-sa, c=be
4835703278459591479716313cn=globalsign primary class 1 ca, ou=primary class 1 ca, o=globalsign nv-sa, c=be

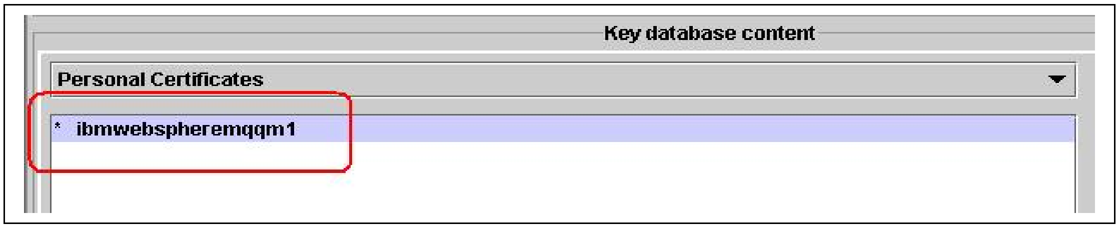
Enter a new label:

ibmwebspheremqqm1

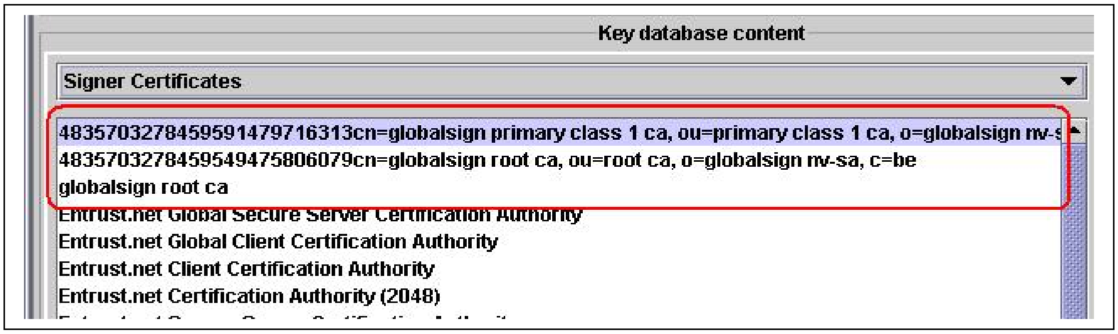
Click on `Apply`**.**

Click OK.

You will see the certificate listed under Personal Certificates, and the Certification Authority certificates listed under Signer Certificates:

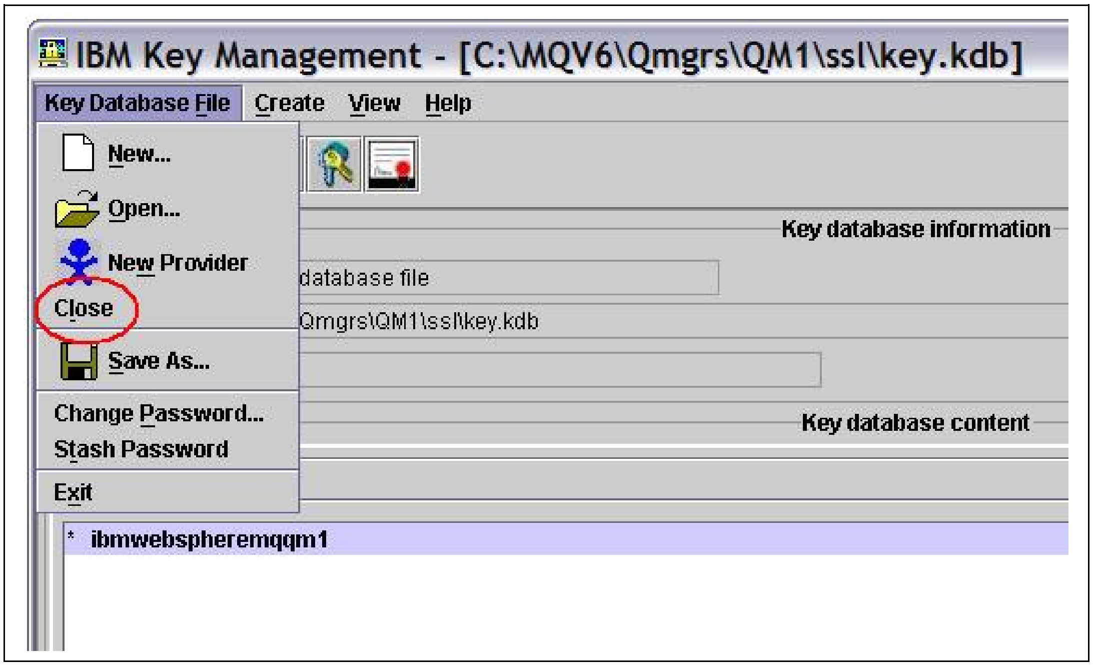


* ibmwebspheremqqm1



4835703278459591479716313cn=globalsign primary class 1 ca, ou=primary class 1 ca, o=globalsign nv-s
4835703278459549475806079cn=globalsign root ca, ou=root ca, o=globalsign nv-sa, c=be
globalsign root ca
Entrust.net Global Secure Server Certification Authority
Entrust.net Global Client Certification Authority
Entrust.net Client Certification Authority
Entrust.net Certification Authority (2048)

Close the repository:



Close the iKeyMan GUI.

Repeat these steps for queue manager QM2 (on the machine where QM2 runs).

Set up the channels for SSL authentication and test

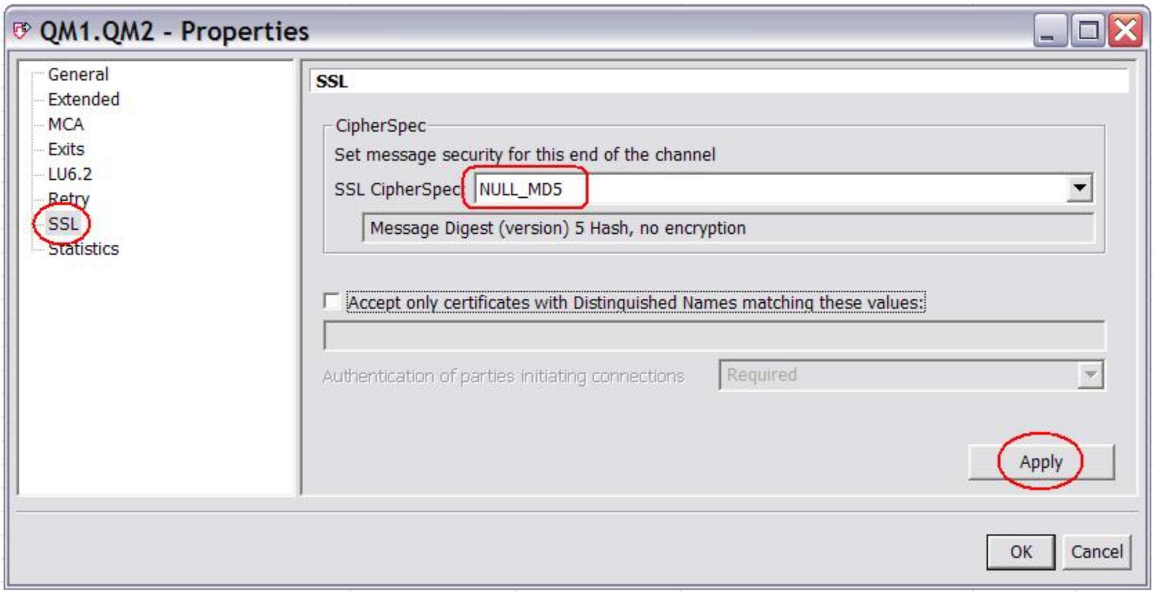
Open MQ Explorer and **start the queue managers.**

Set up channels on QM1

Select Channels (under Advanced).

Right-click on `QM1.QM2` ⇨ Properties ⇨ SSL

Set the `SSL Cipherspec` **to NULL_MD5** (any other cipherspec will do, as long as it matches that of the receiver channel in QM2):


QM1.QM2 - Properties

SSL

CipherSpec

Set message security for this end of the channel

SSL CipherSpec: NULL_MD5

Message Digest (version) 5 Hash, no encryption

□ Accept only certificates with Distinguished Names matching these values:

Authentication of parties initiating connections: Required

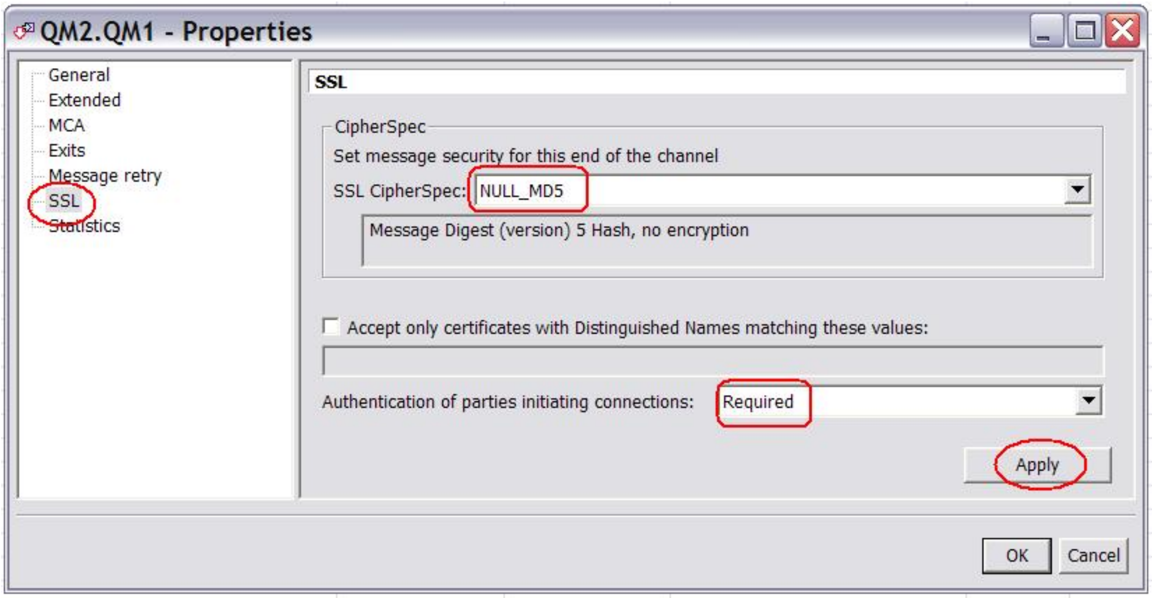
Click `Apply`.

Click `OK`.

Right-click on `QM2.QM1` ⇨ Properties ⇨ SSL

Set the `SSL Cipherspec` **to NULL_MD5** (again, any cipherspec will do, as long as it matches that of the sender channel in QM2).

Leave `Authentication of partner` (…) as `Required`:


QM2.QM1 - Properties

SSL

CipherSpec

Set message security for this end of the channel

SSL CipherSpec: NULL_MD5

Message Digest (version) 5 Hash, no encryption

□ Accept only certificates with Distinguished Names matching these values:

Authentication of parties initiating connections: Required

Click `Apply`.

Click `OK`.

Set up channels on QM2

Select Channels (under Advanced).

Right-click on QM2.QM1 ⇨ Properties ⇨ SSL

Set the SSL Cipherspec **to NULL_MD5.**

Click Apply.

Click OK.

Right-click on QM1.QM2 ⇨ Properties ⇨ SSL

Set the SSL Cipherspec **to NULL_MD5.**

Leave Authentication of partner (…) as Required.

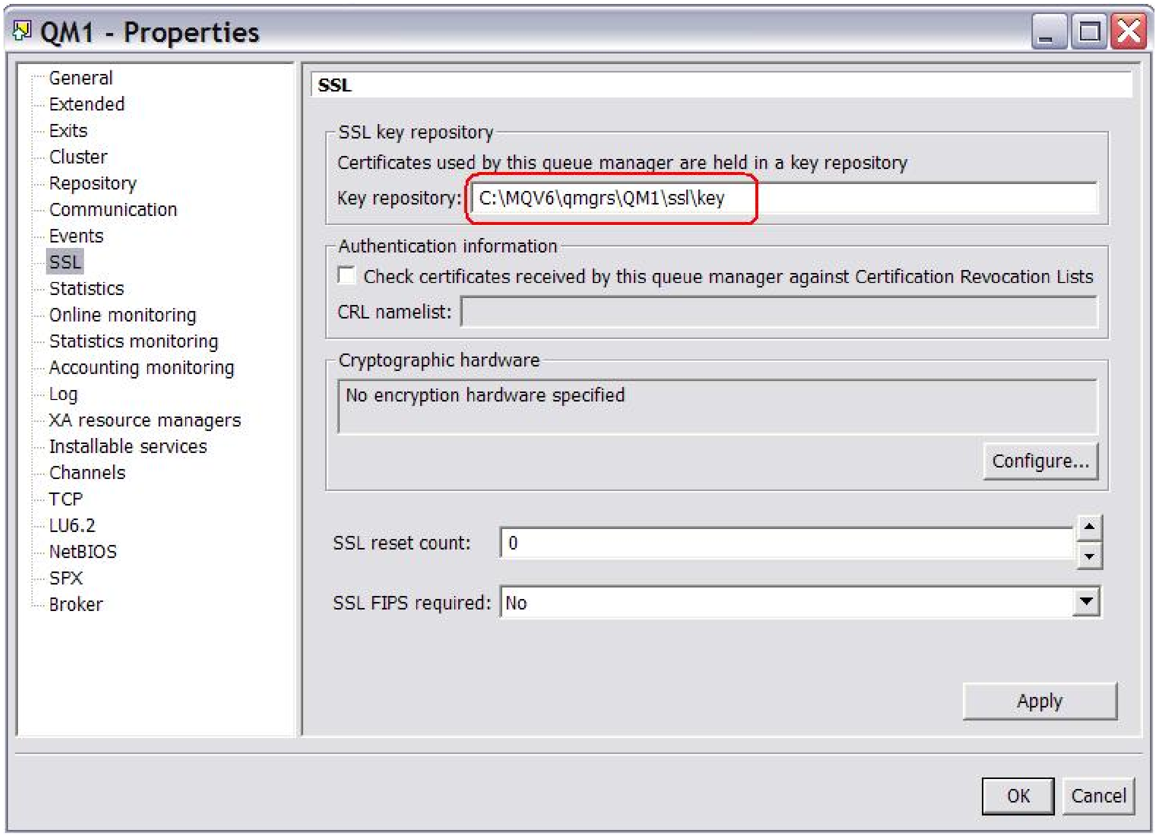
Click Apply.

Click OK.

Verify the key repository location

From MQ Explorer: **right-click on queue manager** QM1 ⇨ Properties ⇨ SSL.

Check that the key repository matches the location and name of the key repository you created. In our example, this is <MQdir>\qmgrs\QM1\ssl\key (note that the key repository file extension, .kdb, is omitted):



SSL key repository

Certificates used by this queue manager are held in a key repository

Key repository: C:\MQV6\qmgrs\QM1\ssl\key

Authentication information

□ Check certificates received by this queue manager against Certification Revocation Lists

CRL namelist:

Cryptographic hardware

No encryption hardware specified

SSL reset count: 0

SSL FIPS required: No

Repeat the check for queue manager `QM2`.

Start the channels

Start the sender channel `QM1.QM2`. You should see the channel status change to Running.

Switch to the QM2 machine, and **start the sender channel** `QM2.QM1`.

This concludes the SSL setup for two Windows queue managers using an external Certification Authority.