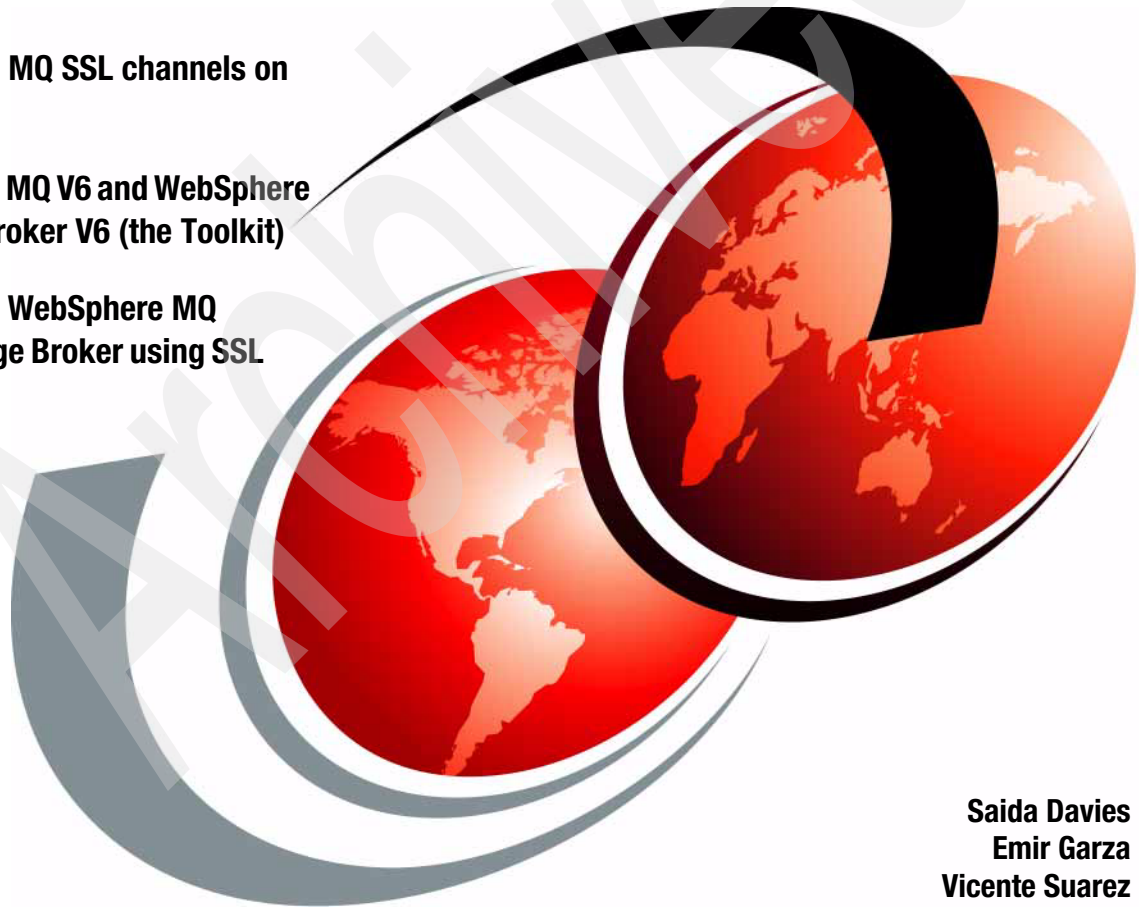


WebSphere MQ V6, WebSphere Message Broker V6, and SSL

WebSphere MQ SSL channels on
Windows

WebSphere MQ V6 and WebSphere
Message Broker V6 (the Toolkit)

Connecting WebSphere MQ
and Message Broker using SSL



Saida Davies
Emir Garza
Vicente Suarez



International Technical Support Organization

**WebSphere MQ V6, WebSphere Message Broker V6,
and SSL**

November 2006

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (November 2006)

This edition applies to Version 6 of IBM WebSphere MQ and Version 6 of IBM WebSphere Message Broker.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team that wrote this Redpaper	ix
Become a published author	x
Comments welcome	xi
Chapter 1. Connecting two Windows queue managers using SSL	1
1.1 Basic configuration	2
1.1.1 Creating the queue managers	2
1.1.2 Setting up the channels	3
1.1.3 Checking the channels	4
1.2 SSL: The very basics	5
1.3 Process overview	6
1.3.1 Creating a key repository for each queue manager	6
1.3.2 Obtaining a certificate for each queue manager	8
1.3.3 Installing the certificates in the key repositories	10
1.3.4 Setting up the channels for SSL authentication and testing	18
Chapter 2. WebSphere MQ V6 clients on Windows	21
2.1 Process overview	22
2.2 Setting up a non-SSL WebSphere MQ client	22
2.3 Verifying non-SSL client connectivity	23
2.3.1 Using WebSphere MQ server	23
2.3.2 Using channel tables	24
2.4 SSL server authentication	27
2.4.1 Creating a key repository for the queue manager	28
2.4.2 Creating a self-signed certificate	31
2.4.3 Installing the CA part in the client's key repository	33
2.4.4 Testing SSL server authentication	36
2.5 SSL client authentication	38
2.5.1 Creating a self-signed certificate for the client	39
2.5.2 Installing the CA part in the queue manager's key repository	41
2.5.3 Testing SSL client authentication	42
Chapter 3. WebSphere MQ SSL on z/OS, AIX 5L, and Windows	45
3.1 Introduction	46
3.2 Certification authority setup	47

3.2.1	Creating a root certificate	48
3.2.2	Checking if the certificate was created	53
3.3	z/OS to z/OS	55
3.3.1	Enabling SSL on the queue managers	57
3.3.2	Creating the queue manager certificate	63
3.3.3	Connecting the certificate to the key ring	66
3.3.4	Altering the channel attributes.	68
3.4	z/OS to Windows.	71
3.4.1	Creating a certificate for WIN1	74
3.4.2	Exporting the certificate from RACF	77
3.4.3	Downloading the certificate to Windows	78
3.4.4	Creating a key repository for WIN1	79
3.4.5	Importing the certificate.	79
3.4.6	Altering the channel attributes.	83
3.5	z/OS to AIX 5L.	84
3.5.1	Creating a certificate for the AIX 5L queue manager	86
3.5.2	Exporting the certificate.	89
3.5.3	Downloading the certificate to AIX 5L	91
3.5.4	Creating a key repository for AIX1	91
3.5.5	Importing the certificate for AIX1	95
3.5.6	Altering the channel attributes.	95
3.6	AIX 5L to Windows	96
3.6.1	Creating a certificate for the Windows queue manager	98
3.6.2	Exporting the certificate.	101
3.6.3	Downloading the certificate to Windows	102
3.6.4	Creating a key repository for WIN2.	103
3.6.5	Importing the certificate.	104
3.6.6	Altering the channel attributes.	105
3.7	Windows to Windows	105
3.7.1	Altering the channel attributes.	107
Chapter 4. Connecting the WebSphere Message Broker V6 Toolkit using SSL		109
4.1	Process overview	110
4.2	One-way (server) SSL authentication	110
4.2.1	Creating a self-signed certificate for the queue manager	110
4.2.2	Extracting the CA certificate	114
4.2.3	Installing the CA part in the Toolkit's key repository	116
4.2.4	Testing the one-way SSL connection	120
4.3	Two-way (mutual) SSL authentication.	124
4.3.1	Creating a certificate for the Toolkit	125
4.3.2	Extracting the CA certificate	126
4.3.3	Transferring the CA file	127

4.3.4	Installing the CA certificate in the queue manager	127
4.3.5	Setting up and testing two-way SSL	128
4.4	What could go wrong?	128
4.4.1	Unable to access stashed password	128
4.4.2	WebSphere MQ reason code 2397	129
4.4.3	WebSphere MQ reason code 2009	130
4.4.4	Configuration Manager proxy retry attempts	130
	Related publications	131
	IBM Redbooks	131
	Other publications	131
	Online resources	131
	How to get IBM Redbooks	132
	Help from IBM	132
	Index	133

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™
AIX®
ibm.com®
IBM®

Parallel Sysplex®
RACF®
Redbooks (logo) ™
Redbooks™

SupportPac™
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Internet Explorer, Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redpaper provides step-by-step guides to implement IBM WebSphere® MQ Secure Sockets Layer (SSL) channels in a variety of configurations:

- ▶ Microsoft® Windows® to and from Windows
- ▶ WebSphere MQ clients to WebSphere MQ queue managers (both on Windows)
- ▶ Any-to-any WebSphere MQ channel connections on IBM z/OS®, AIX® 5L™, and Windows, using RACF® as the certification authority
- ▶ WebSphere Message Broker Toolkit

The aim is for you to learn the basics of WebSphere MQ SSL using simple connectivity examples.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from IBM Hursley working with the International Technical Support Organization, Hursley Center.



Saida Davies is a Project Leader for the ITSO and is an experienced IBM Senior IT Specialist. She has published several Redbooks™ on WebSphere Business Integration topics for multiple platforms. Saida has experience in the architecture and design of WebSphere MQ solutions, extensive knowledge of z/OS operating system and a detailed working knowledge of both IBM and independent software vendors' operating system software. In a client-facing role as a senior IT specialist with IBM Global Services, her responsibilities included the development of services for WebSphere MQ within the z/OS and Windows platforms. This covered the architecture, scope, design, project management, and implementation of the software on stand-alone systems or on systems in a Parallel Sysplex® environment. She has received Bravo Awards for her project contributions. Saida has a degree in Computer Studies and her background includes z/OS systems programming. Saida supports Women in Technology activities and contributes and participates in the their meetings.



Emir Garza is an IT Specialist from IBM Hursley, in the U.K. He has seven years of experience in the business integration field. His areas of expertise include WebSphere MQ and WebSphere Message Broker.



Vicente Suarez is an IT Specialist from IBM Hursley, in the U.K. He has five years of experience in the business integration field. His areas of expertise include WebSphere MQ and WebSphere Message Broker.

The team thanks the following people for their invaluable technical advice to this Redpaper:

Morag Hughson
Software Engineer, IBM Software Group, Application and Integration Middleware Software, IBM Hursley

Hazel Fix
Software Engineer, IBM Software Group, Application and Integration Middleware Software, IBM Hursley

Ian Vanstone
WebSphere MQ Developer, IBM Software Group, Application and Integration Middleware Software, IBM Hursley

Don Graminske
Technical Sales, IBM Software Group, Application and Integration Middleware Software, IBM Phoenix
Don co-authored an earlier version of the chapter about clients.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Connecting two Windows queue managers using SSL

This section presents a step-by-step guide to configuring two WebSphere MQ Version 6 queue managers on Microsoft Windows for communication using Secure Sockets Layer (SSL) channels.

We 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 WebSphere MQ, refer to *WebSphere MQ V6 Security*, SC34-6588. Download the PDF from:

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

1.1 Basic configuration

Two queue managers are needed with a working connection (sender/receiver channel pairs in both directions). Table 1-1 shows the names and attributes used. You can create queue managers with the same names, or adjust the following instructions to match your configuration.

Table 1-1 Basic 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
WebSphere MQ installation directory (throughout this document, <MQdir>)	C:\MQV6	C:\MQV6

If you create the two queue managers as in Table 1-1, the only change you need to make is the IP address. You can create the two queue managers on the same, or on separate, Windows systems. Skip to 1.1.3, “Checking the channels” on page 4, if you already have two interconnected queue managers.

1.1.1 Creating the queue managers

Use Version 6 WebSphere MQ Explorer or a command script to create the queue managers. Example 1-1 shows how to create a queue manager called QM1 with a listener on port 11111.

Example 1-1 Create queue manager QM1

```
@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
```

Example 1-1 on page 2 shows how to create QM1. It can also be adapted to create QM2.

1.1.2 Setting up the channels

The following commands (Example 1-2), when run from a command prompt on the machine where QM1 is running, create the necessary WebSphere MQ objects for QM1 to communicate with QM2.

Example 1-2 Create objects for QM1

```
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, the commands shown in Example 1-3, when run from a command prompt on the machine where QM2 is running, create the objects that QM2 needs to communicate with QM1.

Example 1-3 Create objects for QM2

```
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 q1(Q2) replace | runmqsc QM2
echo def qr(QM1.Q1) replace rname(Q1) rqnname(QM1) | runmqsc QM2
```

1.1.3 Checking the channels

Before proceeding, open a command prompt and check that the channels you intend to use with SSL (in the configuration example QM1.QM2 and QM2.QM1) run correctly (Table 1-2). The following example assumes that the channels are already running, or the transmission queue is triggered.

Table 1-2 Channel test

Test	Machine	Run
QM1 to QM2	Same as QM1	C:\> amqspu t QM2.Q2 QM1 Sample AMQSPUT0 start target queue is QM2.Q2 test msg 1 [Press ENTER] Sample AMQSPUT0 end C:\>
	Same as QM2	C:\> amqsge t Q2 QM2 Sample AMQSGET0 start message <test msg 1> [wait 15 seconds] no more messages Sample AMQSGET0 end C:\>
QM2 to QM1	Same as QM2	C:\> amqspu t QM1.Q1 QM2 Sample AMQSPUT0 start target queue is QM1.Q1 test msg 2 Sample AMQSPUT0 end C:\>
	Same as QM1	C:\> amqsge t 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, you are ready to set up the SSL connection.

1.2 SSL: The very basics

In the SSL protocol, the party that starts a conversation (in this case, the WebSphere MQ sender channel) is the *SSL client*. The other party (WebSphere 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) might require client authentication (this is mutual, or two-way, authentication).

In WebSphere MQ, most customers using SSL channels set them up to request mutual authentication. In this example, one-way authentication is set up 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 a sender channel is started (for example, QM2.QM1), this is what happens; it is called the SSL handshake:

1. QM2 starts the connection and requests a certificate.
2. QM1 sends its certificate. This is encrypted (*signed*) using the certification authority certificate (we describe this more later).
3. QM2 verifies QM1's digital signature in the certificate. QM2 now knows QM1 is who it claims to be.
4. 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 encrypt messages.

From the previous steps, it follows that:

- ▶ The party being authenticated must have a certificate. This is called a *personal certificate*.
- ▶ The authenticating party must be able to decipher the certificate's signature: It must have the *certification authority certificate* used to sign the other party's personal certificate.

1.3 Process overview

To establish an SSL connection between QM1 and QM2, use the following process:

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.

1.3.1 Creating a key repository for each queue manager

To create a key repository for queue manager QM1, perform the following steps. Repeat these steps for QM2.

1. Open a Windows command prompt and enter `strmqikm`. This starts the IBM Key Management (iKeyman) GUI.
2. Create a key repository for the queue manager. Select **Key Database File** → **New**.
3. Create a repository as follows:
 - Key database type: **CMS** (Certificate Management System)
 - File name: `key.kdb`
 - Location: `<MQdir>\Qmgrs\QM1\ssl`
In this example: `C:\MQV6\Qmgrs\QM1\ssl`

Figure 1-1 illustrates these options. Click **OK**.

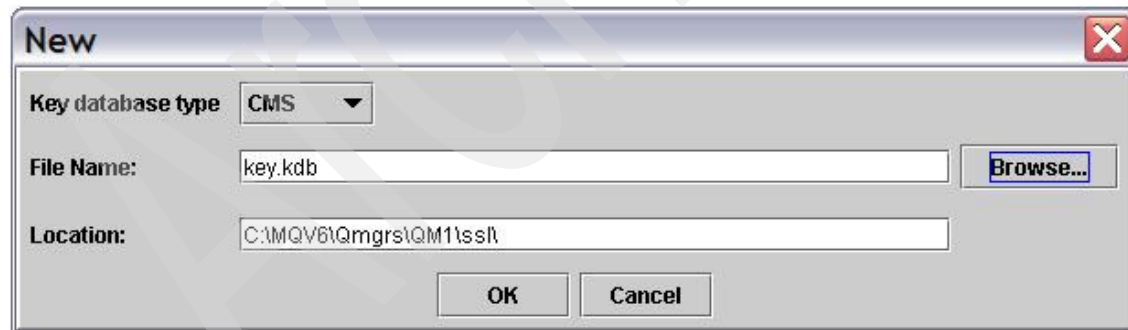


Figure 1-1 Create key repository

4. Enter a password (remember it because it is required later) and select **Stash the password to a file?**, as shown in Figure 1-2. Click **OK**.



Figure 1-2 Key repository password

5. The message shown in Figure 1-3 opens. Click **OK**.



Figure 1-3 Password confirmation

A key repository for queue manager QM1 is created.

6. After creating the key repository, the GUI shows the installed certification authority certificates provided with iKeyman. Use the drop-down menu (top right) to switch to viewing **Personal Certificates**, as shown in Example 1-4.

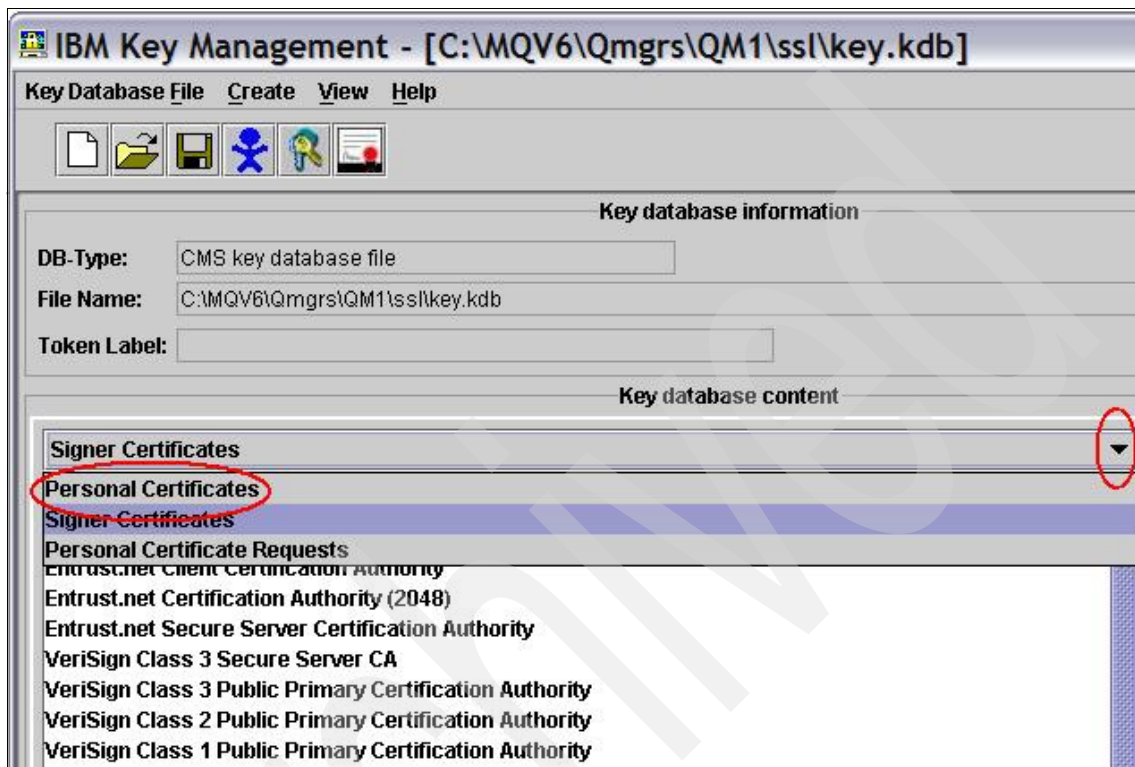


Figure 1-4 Switch to Personal Certificates view

Keep the iKeyman GUI open, because you need to come back to it shortly.

Repeat the previous steps for QM2 on the machine where *queue manager QM2* is running.

1.3.2 Obtaining a certificate for each queue manager

The following instructions show how to obtain a certificate for queue manager QM1. 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 following instructions are for obtaining a demo (valid for 30 days) personal certificate from GlobalSign. There are other sites for requesting certificates, which you can easily find by performing an Internet search.

GlobalSign is convenient because it does not require registration.

Note: Certificates for purposes other than a demo do incur a cost (dispensing certificates is what certification authorities do for a living).

To obtain a certificate:

1. Open Microsoft Internet Explorer and go to the following Web site:
<http://www.globalsign.com>
2. Select **Buy Certificates**. This opens a list. From the list, select **Personal Certificates**.
3. This opens:
http://www.globalsign.com/digital_certificate/personalsign/index.cfm
Select PersonalSign Demo Certificate (click the **Get Yours Now!** button).
4. This takes you to a page showing an 8-step process (see Table 1-3) for obtaining your certificate.

Table 1-3 Obtaining a personal certificate

Step ^a	Comments
Step 1. CHECK ROOT First, you need to install GlobalSign's Root Certificate.	This is installed already.
Step 2. SUBMIT YOUR E-MAIL ADDRESS Submit your e-mail address and provide a password.	Your Internet e-mail address is required (for example, emir_garza@uk.ibm.com) and a password that is used in step 4. After selecting Go to step 3 , GlobalSign sends 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 receive the e-mail, from ca@globalsign.net, within a minute. It contains a hyperlink. Click it (ensure that clicking the hyperlink invokes the <i>same</i> browser you were using before).
Step 4. ENTER YOUR PASSWORD Enter the password you provided in step 2.	Enter the password you chose in step 2.
Step 5. PROVIDE PERSONAL DATA Enter some personal information.	Click Go to step 6 without making any changes. In particular, leave Protect private key set to No .

Step ^a	Comments
Step 6. ACCEPT AGREEMENT Read the subscriber agreement.	Click 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 receive another e-mail within five minutes. It contains a hyperlink that downloads your certificate and opens a browser page with an "Install" button. Ensure it is the <i>same</i> browser as before.
Step 8. INSTALL CERTIFICATE When receiving our mail, click the hyperlink in order to install your certificate.	Click Install . Click OK to any browser warnings. You receive a message confirming that your certificate is installed. Click OK .

a. Steps reprinted from the GlobalSign Web site.

Verify that the certificate is installed. The following text is from the final confirmation window:

Note for Microsoft Internet Explorer® Users:

After having installed your certificate, 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 have opened the certificate manager, where you see a GlobalSign Certificate "issued to" your e-mail address.

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

1.3.3 Installing the certificates in the key repositories

The following instructions show how to install the certificate just obtained for queue manager QM1. Repeat these steps for QM2.

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

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

Exporting the certificate from Internet Explorer

To export the certificate from Internet Explorer:

1. Open Internet Explorer and select **Tools → Internet Options → Content → Certificates**.

2. You see the certificate you just obtained and installed, as shown in Figure 1-5. Select (click) the certificate and then click **Export**.

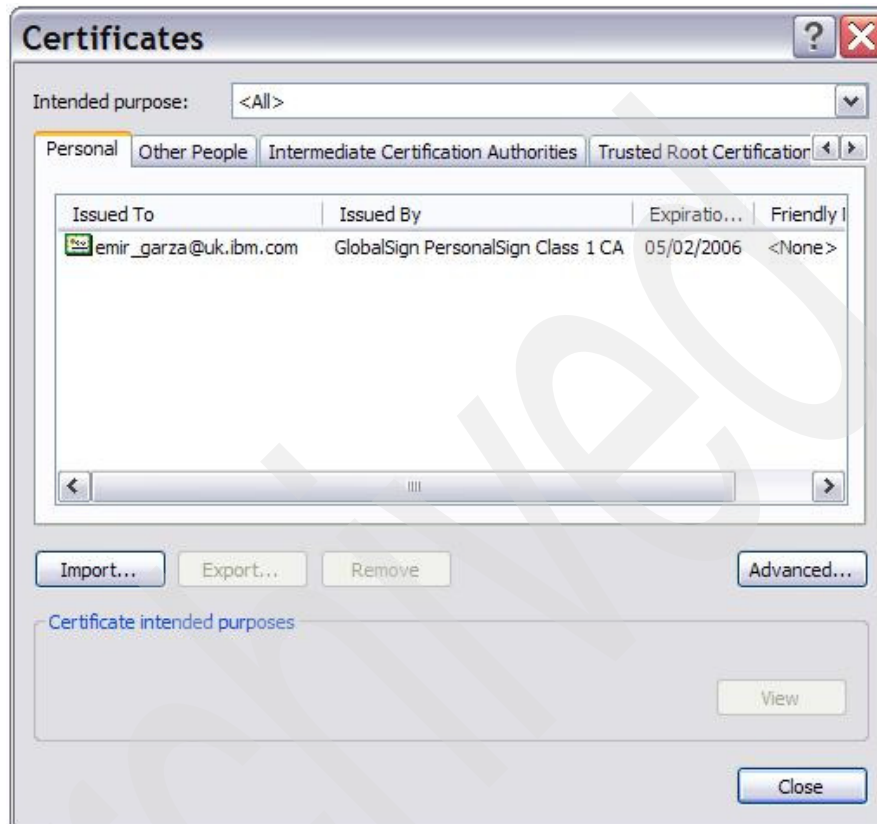


Figure 1-5 Internet Explorer Certificates

3. The Certificate Export Wizard opens. At the Welcome window, click **Next**.
4. At the Export Private Key dialog, select **Yes**.

5. At the Export File Format window, select **Include all certificates in the certification path if possible** and **Enable strong protection**, as shown in Figure 1-6. Click **Next**.



Figure 1-6 Export Certificate

6. At the Password window, enter a password to protect the exported certificate (remember it, because it is required when importing).
7. At the File to Export window, enter (or navigate to) `<MQdir>\Qmgrs\QM1\ssl\QM1.pfx` (in this example, `C:\MQV6\Qmgrs\QM1\ssl\QM1.pfx`). Click **Next**.
8. At the completion window, verify the settings. The settings *must* be as shown in Figure 1-7 on page 13:
 - File Name: `C:\MQV6\Qmgrs\QM1\ssl\QM1.pfx`
 - Export Keys: Yes
 - Include all certificates in the certification path: Yes
 - File Format: Personal Information Exchange (*.pfx)Click **Finish**.



Figure 1-7 Export Certificate settings

You see the message:

The export was successful.

The next step imports the certificate into QM1's key repository.

Importing the certificate

To import the certificate:

1. Switch to the iKeyman GUI, which was left open at the end of 1.3.1, "Creating a key repository for each queue manager" on page 6.

If iKeyman is closed:

- a. Enter **strmqikm** from a command prompt.
- b. Select **Key Database File** → **Open** → **<MQdir>\Qmgrs\QM1\ssl\key.kdb**.
- c. Enter the password.
- d. Select **Personal Certificates**.

2. The Personal Certificates pane is empty. Click **Import**, as shown in Figure 1-8.

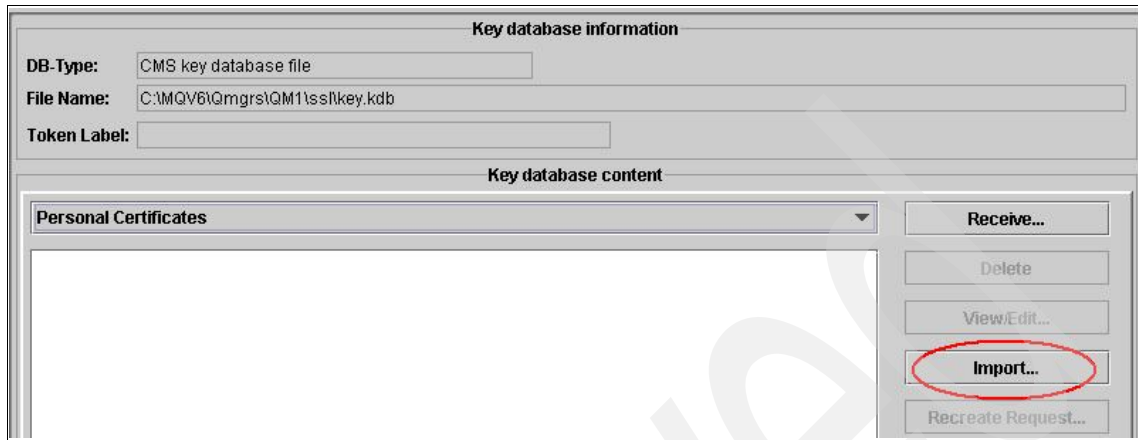


Figure 1-8 Import certificate

3. This opens the Import Key window (Figure 1-9). Select **PKCS12** for the Key file type.

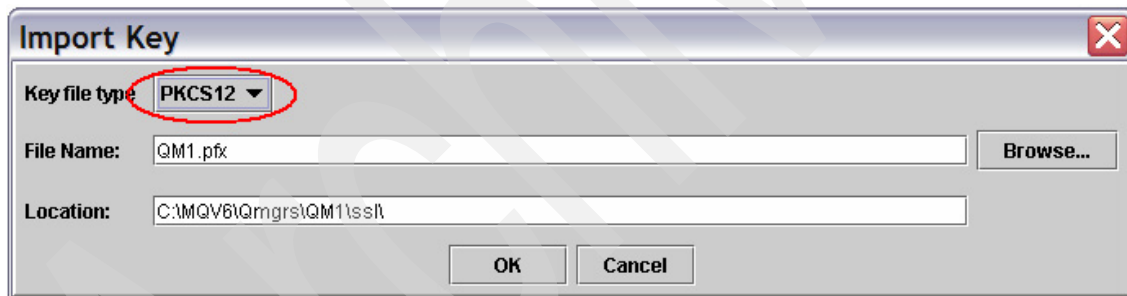


Figure 1-9 Import Key

Click **Browse**:

- a. Navigate to <MQdir>\Qmgrs\QM1\ssl\.
- b. Select **All Files** (see Figure 1-10).
- c. Select **QM1.pfx** (the certificate exported from Internet Explorer).

Click **Open**.

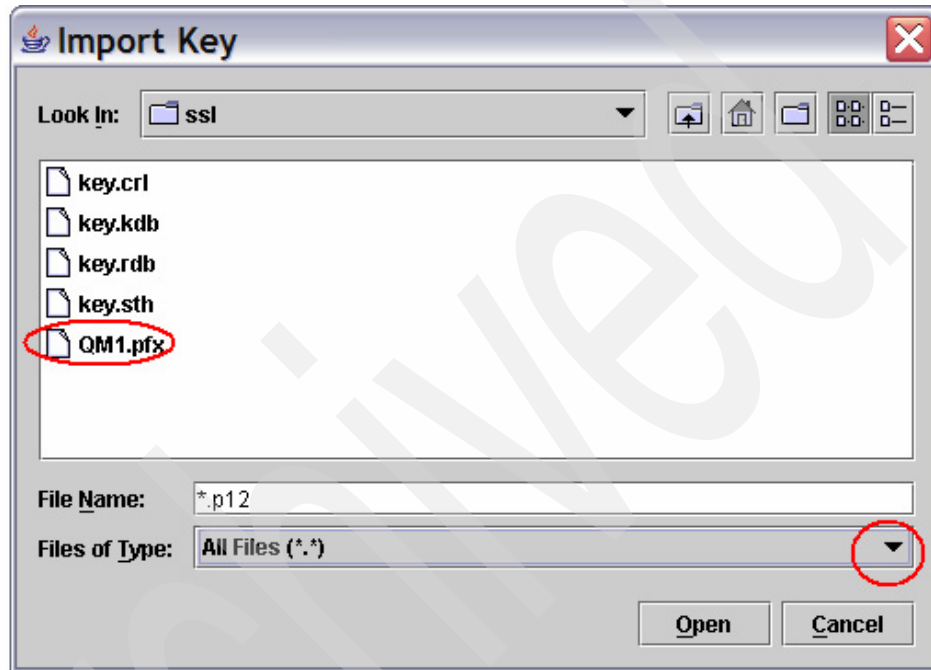


Figure 1-10 Select exported certificate

4. This returns you to the Import Key window. Click **OK**.
5. Enter the password you gave when exporting the certificate from Internet Explorer. Click **OK**.
6. The Change Labels window (Figure 1-11 on page 16) asks “Would you like to change any of these labels before completing the import process?” and shows four certificates¹: 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).
Select the personal certificate (this enables the new label field, at the bottom).

¹ 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 might work, but if it does not, repeat the export process, this time including all certificates.

Enter the label. This must be `ibmwebspheremq` followed by the queue manager name, all in lowercase, `ibmwebspheremqm1`. Click **Apply**. Click **OK**.

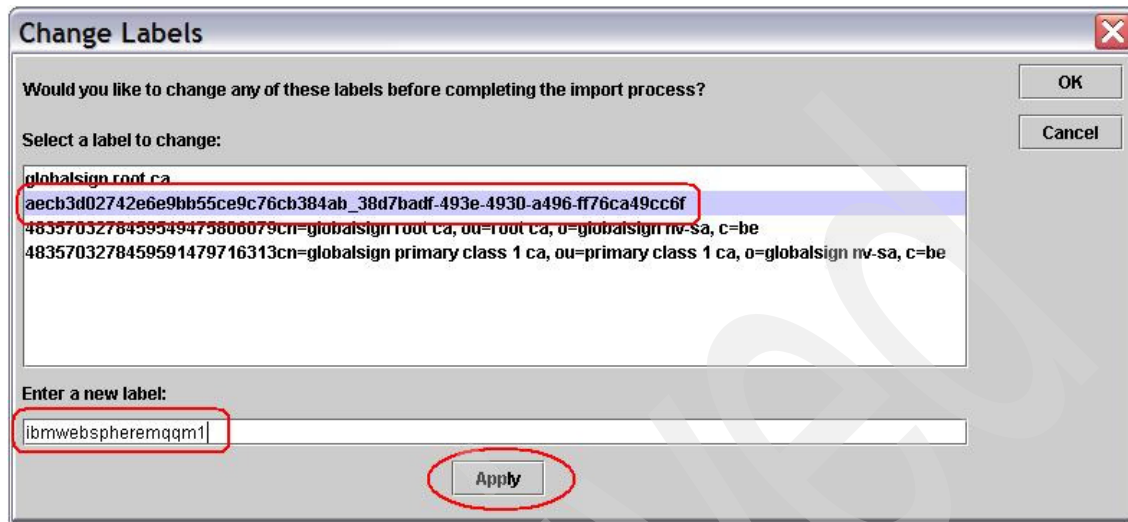


Figure 1-11 Change Labels

You see the certificate listed under Personal Certificates (Figure 1-12).

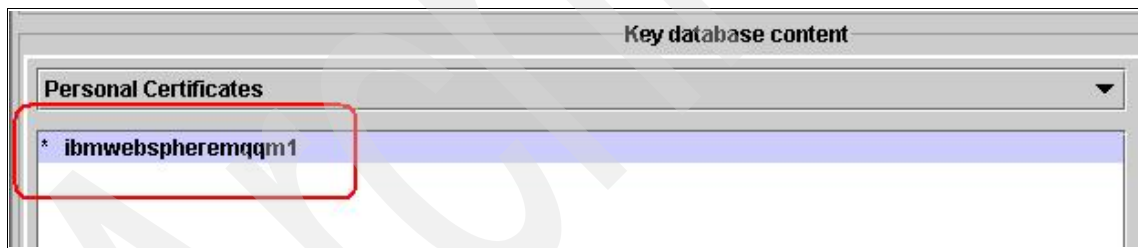


Figure 1-12 Queue manager personal certificate

You also see the certification authority certificates listed under Signer Certificates (Figure 1-13).

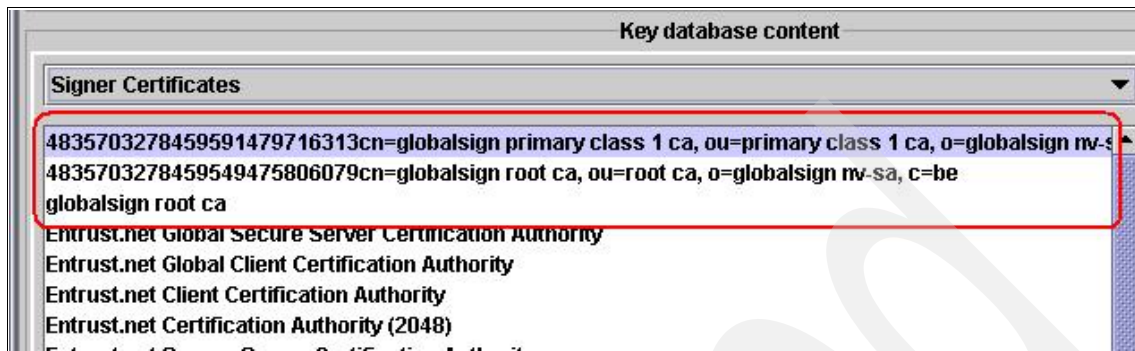


Figure 1-13 Certification authority certificates

7. Close the repository (select **Key Database File** → **Close**), as shown in Figure 1-14.

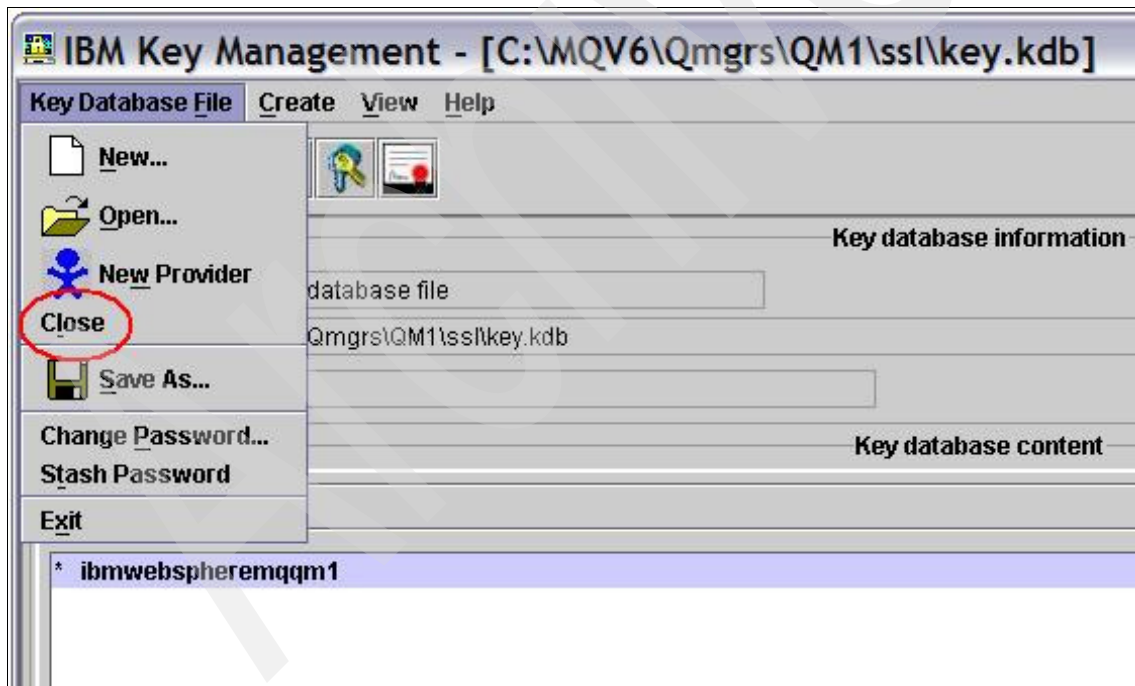


Figure 1-14 Close key repository

8. Close the iKeyman GUI.

Repeat these steps for queue manager QM2 (on the machine where *QM2* is running).

1.3.4 Setting up the channels for SSL authentication and testing

Open WebSphere MQ Explorer on both machines, and start the queue managers.

Setting up channels on QM1

To set up channels on QM1:

1. Select **Channels** (under Advanced).
2. Right-click **QM1.QM2** and select **Properties** → **SSL**.

Set the SSL CipherSpec to **NULL_MD5** (any other cipherspec works, as long as it matches that of the receiver channel in QM2), as shown in Figure 1-15. Click **Apply**. Click **OK**.

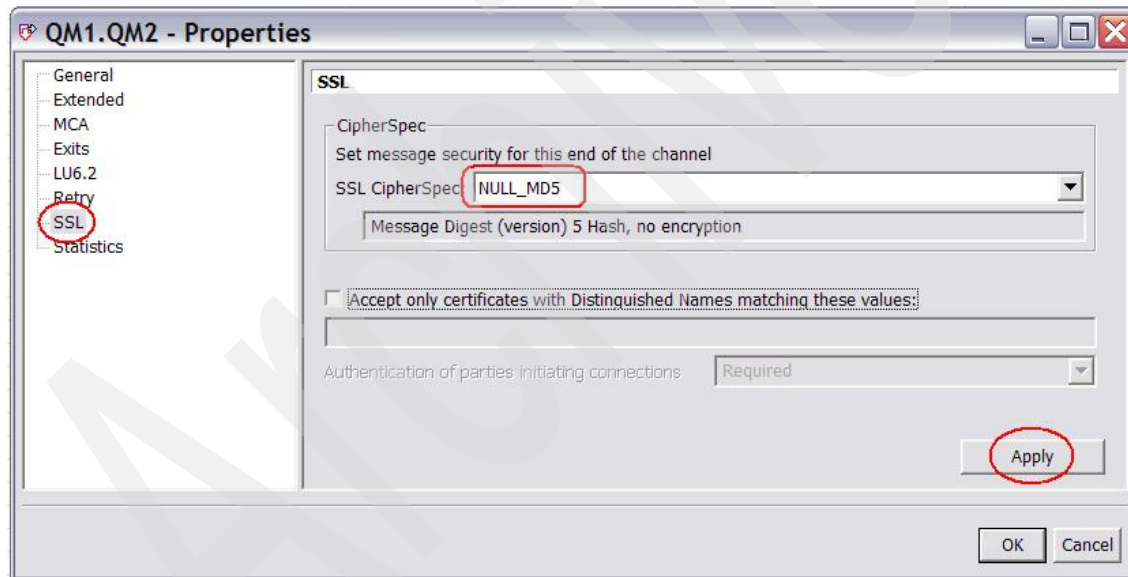


Figure 1-15 Sender channel SSL attributes

3. Right-click **QM2.QM1** and select **Properties** → **SSL**.

Set the SSL CipherSpec to **NULL_MD5** (again, any cipherspec works, as long as it matches that of the sender channel in QM2).

Leave Authentication of parties initiating connections as **Required**, as shown in Figure 1-16. Click **Apply**. Click **OK**.

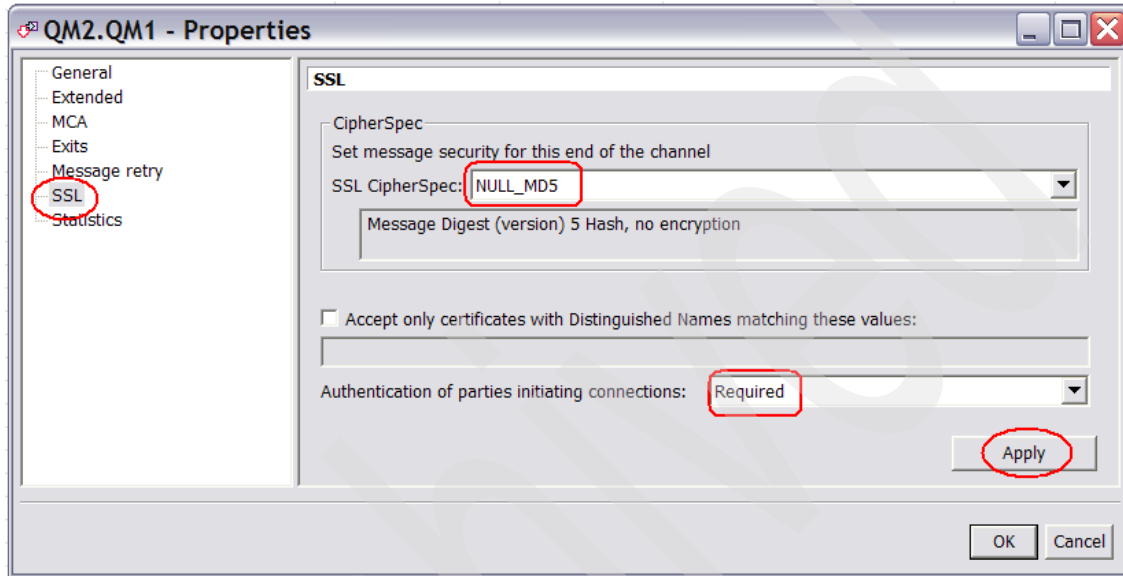


Figure 1-16 Receiver channel SSL properties

Setting up channels on QM2

To set up channels on QM2:

1. Select **Channels** (under Advanced).
2. Right-click **QM2.QM1** and select **Properties** → **SSL**.
Set the SSL Cipherspec to **NULL_MD5**. Click **Apply**. Click **OK**.
3. Right-click **QM1.QM2** and select **Properties** → **SSL**.
Set the SSL Cipherspec to **NULL_MD5**.
Leave Authentication of parties initiating connections as **Required**.
Click **Apply**. Click **OK**.

Verifying the key repository location

From WebSphere MQ Explorer, right-click queue manager **QM1** and select **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`; see Figure 1-17. Note that the key repository file extension, `.kdb`, *must* be omitted.

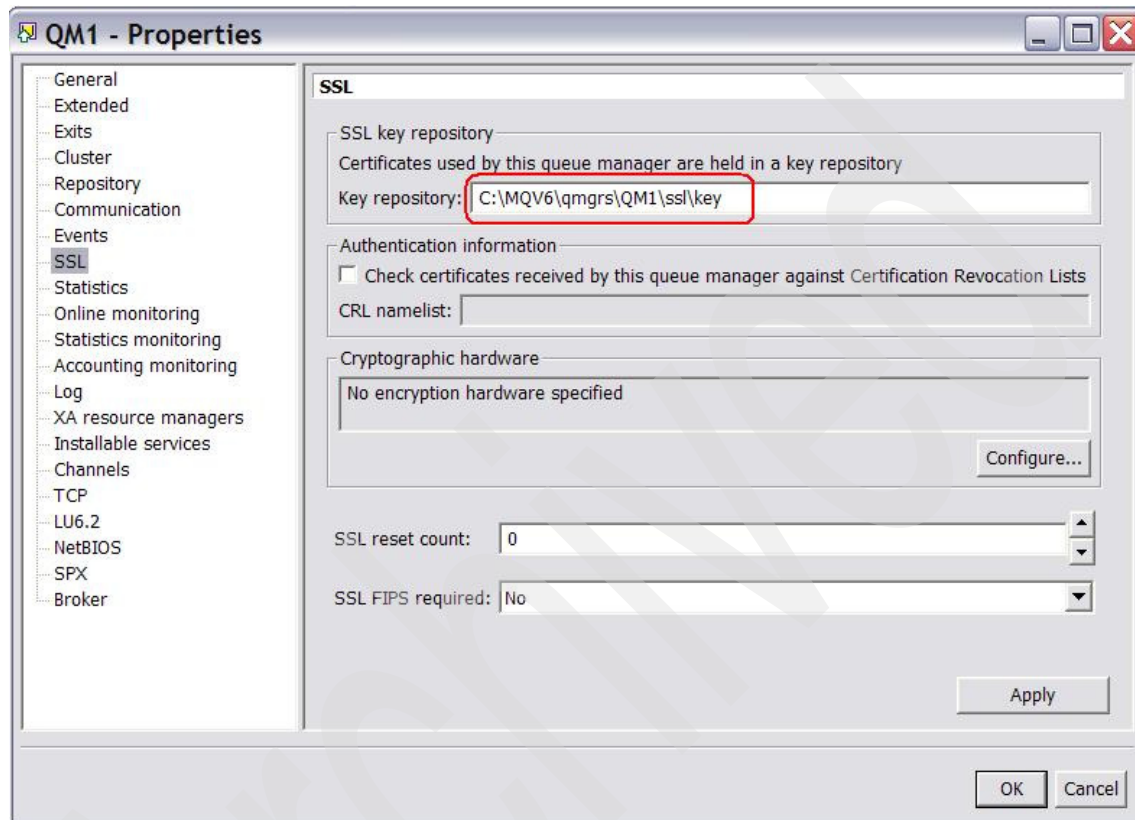


Figure 1-17 Verify the key repository location

Repeat the check for queue manager QM2.

Starting the channels

From the QM1 machine, start the sender channel QM1.QM2.

From the QM2 machine, start the sender channel QM2.QM1.

The channels start, meaning they have successfully exchanged their SSL certificates.

This concludes the SSL setup for two Windows queue managers using an external certification authority.

WebSphere MQ V6 clients on Windows

This chapter provides a step-by-step guide for configuring WebSphere MQ clients to use Secure Sockets Layer (SSL). We assume that you are familiar with the basic operation of WebSphere MQ V6 Explorer.

2.1 Process overview

Perform the following steps to set up a WebSphere MQ client to use SSL:

1. Set up a non-SSL WebSphere MQ client.
2. Verify the non-SSL client connectivity.
3. Set up a one-way SSL channel connection. One-way means that only the queue manager (in SSL terms, the server) presents a certificate, which the client authenticates.
4. Convert the SSL connection to two-way (that is, mutual authentication between the client and queue manager).

The instructions that follow assume that there are two machines, one for the queue manager and one for the client; the instructions also work if both the client and queue manager run in the same machine.

2.2 Setting up a non-SSL WebSphere MQ client

As an optional step, create and start a queue manager, called SSLQM, listening on port 33333. This is to ensure that we start with a “clean” queue manager, *with no certificates assigned*.

You can use an existing queue manager if you prefer. The instructions in this chapter assume that you have the configuration listed in Table 2-1.

Table 2-1 Queue manager configuration

Queue manager name	SSLQM
IP address	192.168.1.64
Listener port	33333
Server connection channel	SSL.CLIENTS
Local queue (for testing)	Q1
WebSphere MQ installation directory (throughout this document, <MQdir>)	C:\MQV6

To create a queue manager and associated objects as listed in Table 2-1 on page 22, either use WebSphere MQ Explorer, or open a Windows command prompt and enter the commands¹ shown in Example 2-1.

Example 2-1 Create queue manager

```
@echo Create queue manager
crtmqm -u SSLQM.DLQ SSLQM
@echo Start queue manager
amqmdain qmgr start SSLQM
@echo Create listener
@echo DEFINE LISTENER('LISTENER.TCP') TRPTYPE(TCP) PORT(33333)
CONTROL(QMGR) | runmqsc SSLQM
@echo def ql(SSLQM.DLQ) | runmqsc SSLQM
@echo def ql(Q1) | runmqsc SSLQM
@echo def chl(SSL.CLIENTS) chltype(SVRCONN) | runmqsc SSLQM
@echo START LISTENER('LISTENER.TCP') | runmqsc SSLQM
```

2.3 Verifying non-SSL client connectivity

This section explains how to use and test non-SSL connectivity.

2.3.1 Using WebSphere MQ server

First, check basic client connectivity using the MQSERVER environment variable:

1. If not already started, start the queue manager and its listener. Use WebSphere MQ Explorer, or the command:
`amqmdain qmgr start SSLQM`
2. On the client machine², open a command prompt and set the MQSERVER environment variable:
`set MQSERVER=SSL.CLIENTS/TCP/192.168.1.64(33333)`

¹ You can also place the commands in a Windows command (.cmd) script.

² You can use the queue manager machine if you do not have a separate machine for the client.

3. Run the sample programs **amqsputc** and **amqsgetc** to put and get messages from the test queue, Q1, as shown in Example 2-2.

Example 2-2 Test the client connection

```
C:\>set MQSERVER=SSL.CLIENTS/TCP/192.168.1.64(33333)
```

```
C:\>amqsputc Q1
```

```
Sample AMQSPUTO start
```

```
target queue is Q1
```

```
7777
```

```
[Press ENTER to end amqsputc]
```

```
Sample AMQSPUTO end
```

```
C:\>amqsgetc Q1
```

```
Sample AMQSGETO start
```

```
message <7777>
```

```
[Wait 15 seconds]
```

```
no more messages
```

```
Sample AMQSGETO end
```

```
C:\>
```

You cannot use the MQSERVER environment variable to set up SSL channels; you need to create a channel table, which will be pointed to by the environment variables MQCHLLIB and MQCHLTAB. The next section explains how to set up the channel table.

2.3.2 Using channel tables

The instructions that follow are for setting up a WebSphere MQ client to use a channel table. This is a summary of the necessary steps:

1. Create a client connection channel on the queue manager machine (this creates the channel table).
2. Transfer the channel table file to the client machine.
3. Set up the appropriate environment variables on the client machine.
4. Test the WebSphere MQ client with channel tables.

Creating a client connection channel

Perform the following steps on the queue manager machine:

1. First, check that you do not have the channel table environment variables set.
Open a command prompt. Enter **set mqchl**, as shown in Example 2-3. There should be no variables defined.

Example 2-3 Check the environment variables

```
C:\>set mqchl
Environment variable mqchl not defined
```

2. Define a client connection channel with a connection name that points to the queue manager's host and listener port. You can use the WebSphere MQ Explorer, or the MQSC command as shown in Example 2-4.

Example 2-4 Create a client connection channel

```
C:\>runmqsc SSLQM
5724-H72 (C) Copyright IBM Corp. 1994, 2004.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager SSLQM.
```

```
def chl(SSL.CLIENTS) chltype(CLNTCONN) replace
conname('192.168.1.64(33333)')
  1 : def chl(SSL.CLIENTS) chltype(CLNTCONN) replace
conname('192.168.1.64(33333)')
AMQ8014: WebSphere MQ channel created.
end
  2 : end
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

As a result, you see a file called AMQCLCHL.TAB in your queue manager's @ipcc directory, as shown in Example 2-5.

Example 2-5 Channel table

```
C:\<MQdir>\Qmgrs\SSLQM\@ipcc>dir
Volume in drive C is IBM_PRELOAD
Volume Serial Number is DCC1-463F

Directory of C:\ <MQdir>\Qmgrs\SSLQM\@ipcc

27/06/2006  09:40    <DIR>          .
```

```
27/06/2006 09:40 <DIR> ..
27/06/2006 10:03 4,008 AMQCLCHL.TAB
```

Next, you need to transfer the channel table file to the client machine.

Transferring the channel table file

You need a directory in the client machine to place the channel table. In this example, we use C:\MQCLIENT. To transfer the channel table file:

1. On the client machine, create a directory called C:\MQCLIENT.
2. Copy (or FTP binary) the AMQCLCHL.TAB file from the queue manager's @ipcc directory to the C:\MQCLIENT directory on the client machine.

Setting up the environment variables

Open a command prompt on the client machine. Set the MQCHLLIB and MQCHLTAB environment variables, as shown in Example 2-6.

Example 2-6 Set the environment variables

```
C:\MQCLIENT>set mqchllib=C:\MQCLIENT
C:\MQCLIENT>set mqchltab=AMQCLCHL.TAB
C:\MQCLIENT>set mqchl
mqchllib=C:\MQCLIENT
mqchltab=AMQCLCHL.TAB
C:\MQCLIENT>set mqserver
Environment variable mqserver not defined
```

Where:

- ▶ C:\MQCLIENT is the directory to contain the client channel table.
- ▶ AMQCLCHL.TAB is the default file name for the channel table.

The last command is to ensure that the MQSERVER variable is not set by accident.

Note: You can also set the environment variable for the whole system, but, if you do so, this might interfere with other WebSphere MQ client work on your machine. In particular, you do not set these environment variables for the whole system if you use a single machine for both client and server.

Keep the command prompt window open. If you close it, remember to set the environment variables again.

Testing the WebSphere MQ client with channel tables

At this point, you are ready to put and get messages from the client machine, using the **amqsputc** and **amqsgetc** commands (remember to use the same command prompt window, because a new window will not have the necessary environment variables), as shown in Example 2-7.

Example 2-7 Test the client connection with channel tables

```
C:\MQCLIENT>amqsputc Q1
Sample AMQSPUTO start
target queue is Q1
7777
```

```
Sample AMQSPUTO end
```

```
C:\MQCLIENT>amqsgetc Q1
Sample AMQSGETO start
message <7777>
no more messages
Sample AMQSGETO end
```

After we know that the WebSphere MQ client works, we are ready to add SSL support. First, we implement server authentication only (one-way), and then mutual authentication.

2.4 SSL server authentication

With SSL server authentication, the queue manager must have a certificate, which it presents to the client, so that the client knows it is connecting to the right server. This is what happens when you shop online and your URL starts with `https:`. The browser asks for a certificate, so you know you are giving your credit card details to the right company, but the server (the online shop) does not ask for a certificate from you.

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.

In this example, we create self-signed certificates³.

³ Chapter 1, “Connecting two Windows queue managers using SSL” on page 1, shows how to obtain a certificate from an external authority. Chapter 3, “WebSphere MQ SSL on z/OS, AIX 5L, and Windows” on page 45, shows how to act as an in-house certification authority.

The steps to implement SSL server authentication are:

1. Create a key repository for the queue manager.
2. Create a self-signed certificate for the queue manager.
3. Add the certification authority certificate to the client. (The client needs this to decode the queue manager's certificate.)
4. Test the SSL server authentication.

2.4.1 Creating a key repository for the queue manager

To create a key repository for queue manager SSLQM:

1. On the queue manager machine, open a Windows command prompt and enter `strmqikm`. This starts the IBM Key Management (iKeyman) GUI.
2. Create a key repository for the queue manager. Select **Key Database File** → **New**.
3. Create a repository as shown in Figure 2-1:
 - Key database type: **CMS**
 - File Name: `key.kdb`
 - Location: `<MQdir>\Qmgrs\SSLQM\ssl`
In this example: `C:\MQV6\Qmgrs\SSLQM\ssl`

Click **OK**.

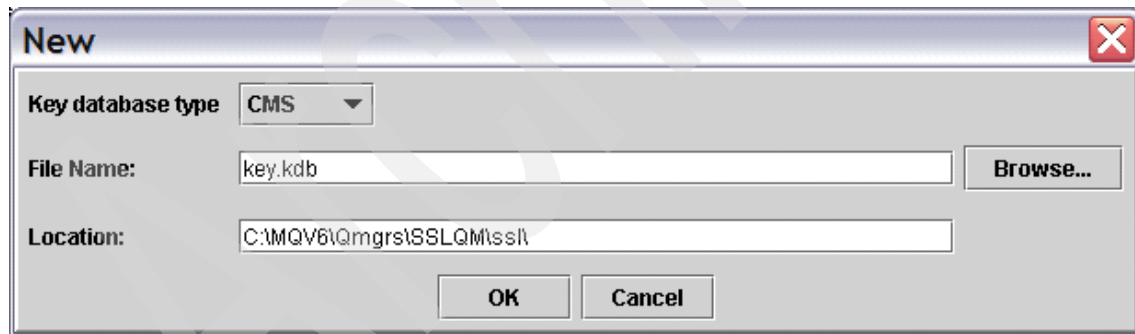


Figure 2-1 Create a key repository

4. Enter a password (remember it, because you need it to open this key repository). Select **Stash the password to a file?**, as shown in Figure 2-2. Click **OK**.



Figure 2-2 Key Repository password

5. The message shown in Figure 2-3 opens. Click **OK**.

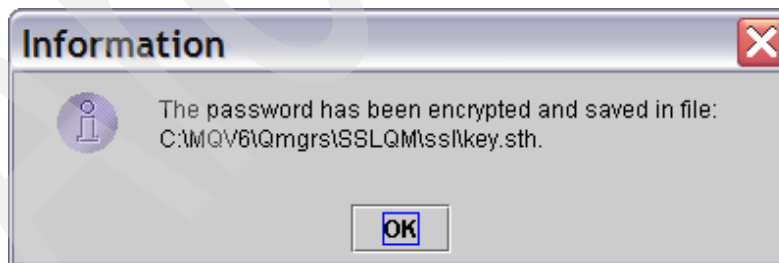


Figure 2-3 Key repository password confirmation

You have created a key repository for queue manager SSLQM.

- After creating the key repository, the GUI shows the installed certification authority certificates provided with iKeyman. Use the menu (top right) to switch to viewing **Personal Certificates**, as shown in Figure 2-4.

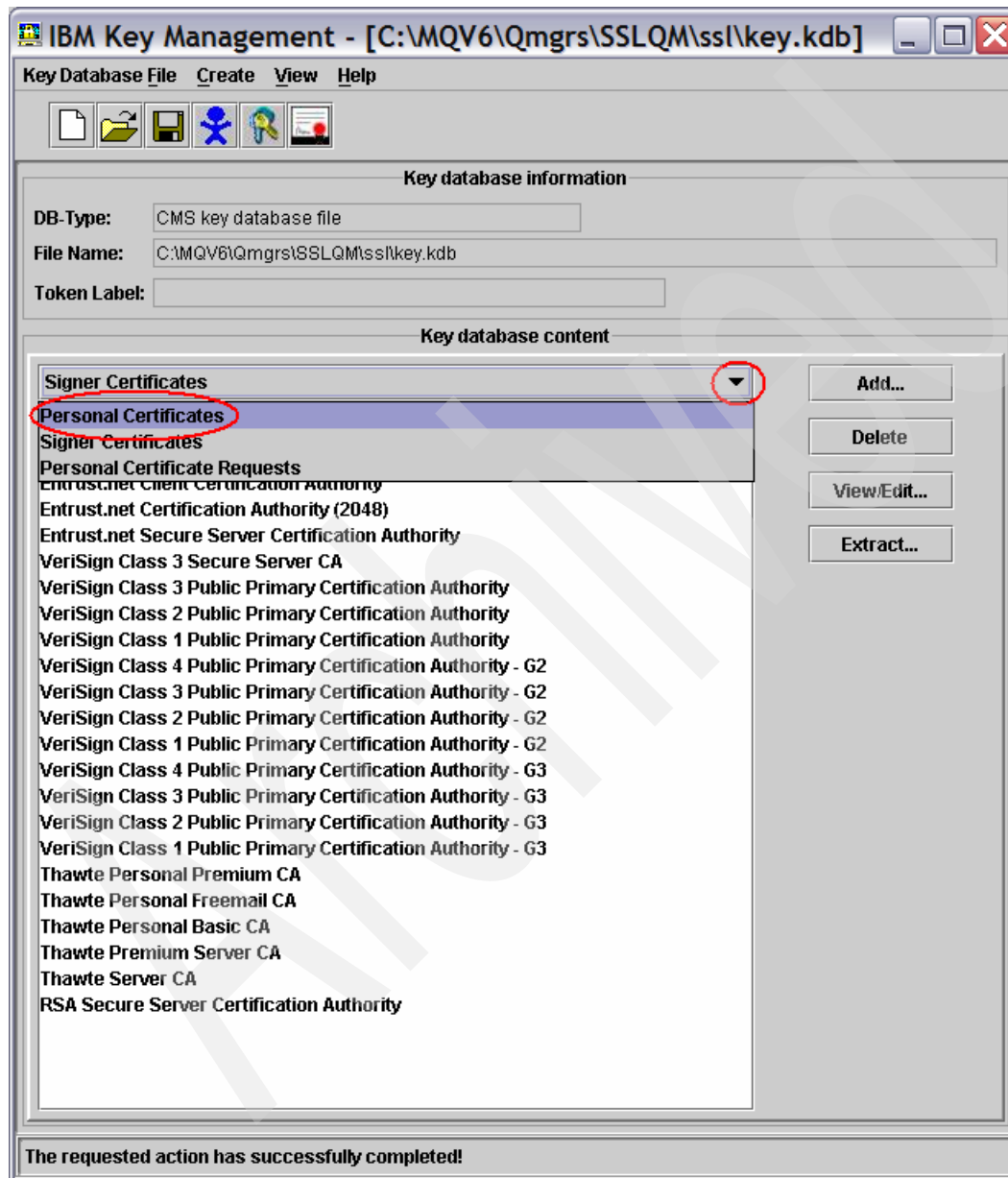


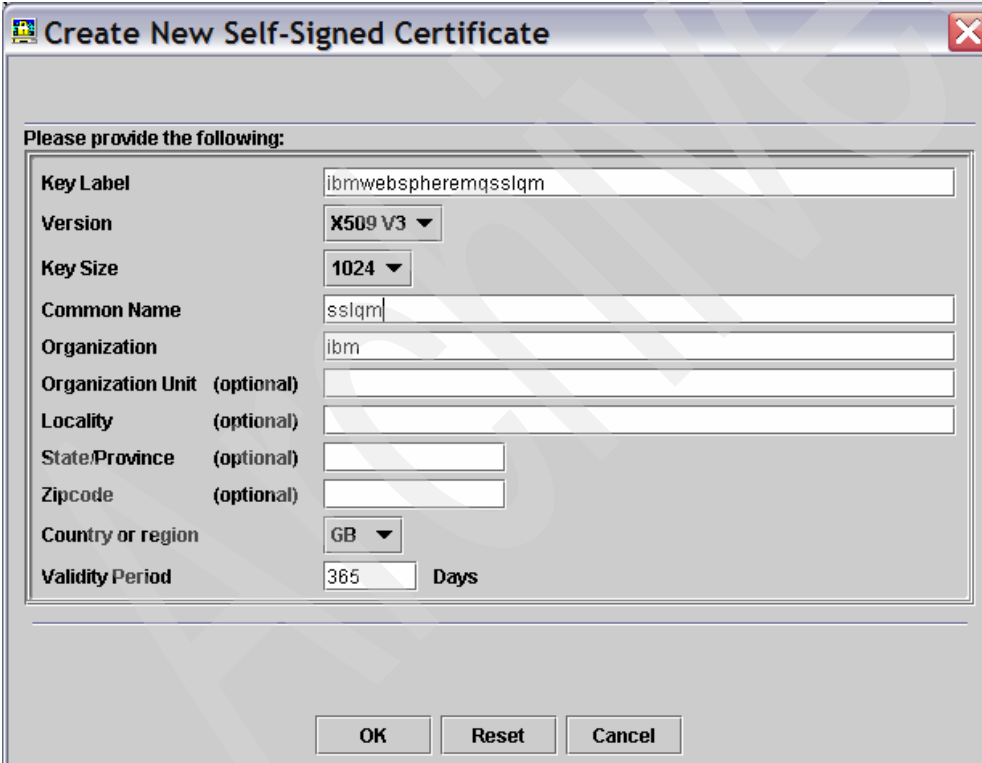
Figure 2-4 Personal Certificates

2.4.2 Creating a self-signed certificate

To create a self-signed certificate:

1. Click **New Self-Signed Certificate** (bottom-right corner⁴).
2. Fill in the certificate attributes, as shown in Figure 2-5:
 - Key Label (must be `ibmwebsphermqsslqm` followed by the queue manager name in *lowercase*): `ibmwebsphermqsslqm`
 - Common Name: `sslqm`
(You can have a different naming convention for the common name; feel free to enter any other value.)
 - Organization: `ibm` (or your company's name)

Leave all other fields unchanged. Click **OK**.



Create New Self-Signed Certificate

Please provide the following:

Key Label	ibmwebsphermqsslqm
Version	X509 V3 ▼
Key Size	1024 ▼
Common Name	sslqm
Organization	ibm
Organization Unit (optional)	
Locality (optional)	
State/Province (optional)	
Zipcode (optional)	
Country or region	GB ▼
Validity Period	365 Days

OK Reset Cancel

Figure 2-5 Create New Self-Signed Certificate

⁴ If you cannot see the New Self-Signed button on the bottom-right corner, it is because you have not yet switched to the Personal Certificates view (see the previous step).

You see the certificate listed, with an asterisk to the left of the name (the asterisk means that this is the default certificate for the key repository).

Now the queue manager has a certificate. The queue manager presents this certificate to the WebSphere MQ client when the client connects. To validate the queue manager's certificate, the client needs the certification authority (CA) certificate.

To extract the CA certificate:

1. Click **Extract Certificate** (bottom-right corner).
 2. Enter the following values:
 - File Name: sslqm.arm
 - Location: C:\<MQDir>\Qmgrs\SSLQM\ssl\
- Click **OK**.

This creates a file called sslqm.arm in C:\<MQDir>\Qmgrs\SSLQM\ssl. If you open the file with Notepad, you see something like Example 2-8.

Example 2-8 Exported certificate

```
-----BEGIN CERTIFICATE-----
MIIByzCCATSgAwIBAgIEQ1kcEDANBgkqhkiG9w0BAQQFADAQMswCQYDVQQGE...
ChMDaWJtMQ0wCwYDVQQDEwRtYnY2MB4XDTA1MTAyMDE2NDkyMFoXDTA2MTAyMT...
MAkGA1UEBhMCR0IxDDAKBgNVBAoTA21ibTENMAAGA1UEAxMEbWJ2NjCBnzANBg...
AA0BjQAwwYkCgYEAOKzAGeurDcg6J7kTCZcm5M3xtka1NZ06Kq2KLPqaQH8cKC...
E29GBAQ4rW9FmVu6iUJUzi4Z8p7oaXw5Y6Y6JGdJNBSONaEwdKE3hIHNnoygpJ...
k1kbgTUDmVSG8fKsutWftbqJmKDXPDaBjRkCAwEAATANBgkqhkiG9w0BAQQFAA...
oGkiAT0I9TxkP31qZV6ZQmoIZ/UCm9FT7V8gMuux7C5sV/Cgh1zsBmf6vVBjrn...
in5Y/Zq7ySx4Y1H2FNV133cvyRXroujE9bInUSwzmicCg71tys4vEnN11Dm/md...
KKVCZ4oHYg==
-----END CERTIFICATE-----
```

You now must transfer the CA Certificate file to the client's key repository. Before that, close the repository with which you were working, as shown in Figure 2-6.

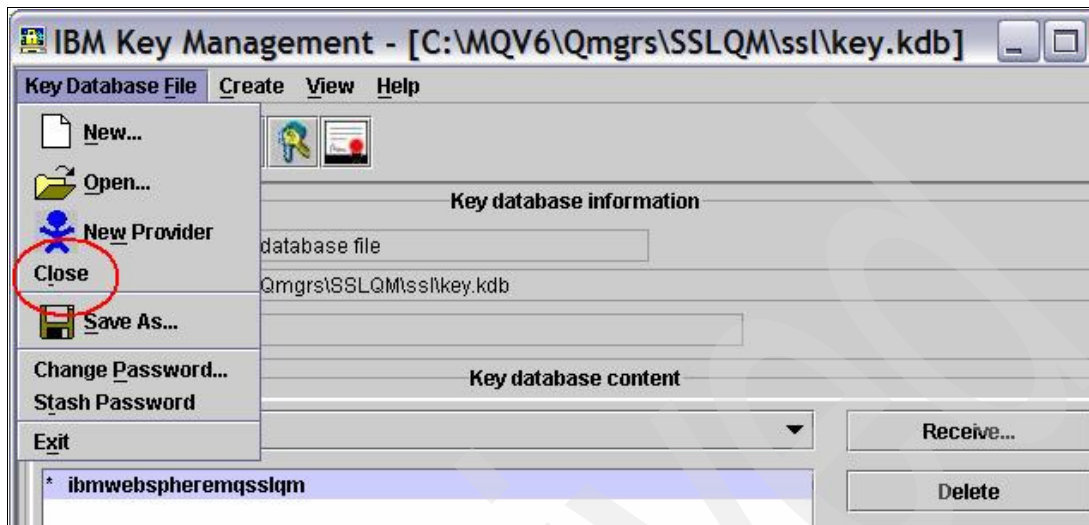


Figure 2-6 Close key repository

Close the iKeyman GUI.

2.4.3 Installing the CA part in the client's key repository

This task consists of the following steps, executed on the client machine (again, this can be the same as the queue manager machine):

1. Create a key repository for the client.
2. Copy or transfer the CA (sslqm.arm) file to the client.
3. Add the CA file to the client's key repository.

We place the client's key repository in C:\MQCLIENT, the same directory we use for the channel table.

Perform the following steps:

1. Copy (or FTP ASCII) the CA certificate file (sslqm.arm) from the queue manager's ssl directory to the C:\MQCLIENT directory on the client machine.
2. On the client machine, switch to the command prompt window you opened earlier ⁵.

Enter **strmqikm**.

3. Create a key repository for the client. This is the same as creating a key repository for the queue manager; the only difference is the key repository location. Select **Key Database File** → **New**.
4. Create a repository as follows:
 - Key database type: **CMS**
 - File Name: `key.kdb`
 - Location: `C:\MQCLIENT`Click **OK**.
5. At the password prompt, enter the password for this repository (twice), and then click **OK**.
6. As before, you see a list of CA certificates provided by default.
Click **Add** (top-right corner).
7. In the next window (Figure 2-7), enter:
 - Certificate file name: `sslqm.arm`
 - Location: `C:\MQCLIENT\`Click **OK**.

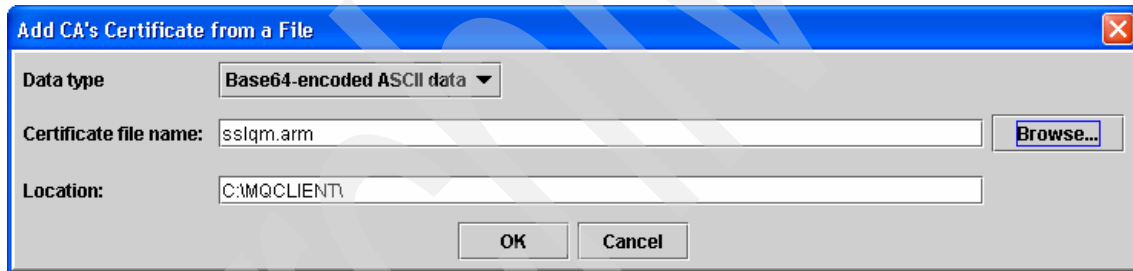


Figure 2-7 Add CA's Certificate

⁵ If you closed it, remember to reenter these commands before trying to connect to the queue manager:

```
set mqchl1lib=C:\MQCLIENT
set mqchl1tab=AMQCLCHL.TAB
```


8. When prompted to enter a label for the certificate, enter `sslqm`. (You can enter any name you want; just make sure it lets you identify the CA certificate as belonging to the queue manager.) Click **OK**.

The certificate now appears in the Signer Certificates repository, as shown in Figure 2-8.

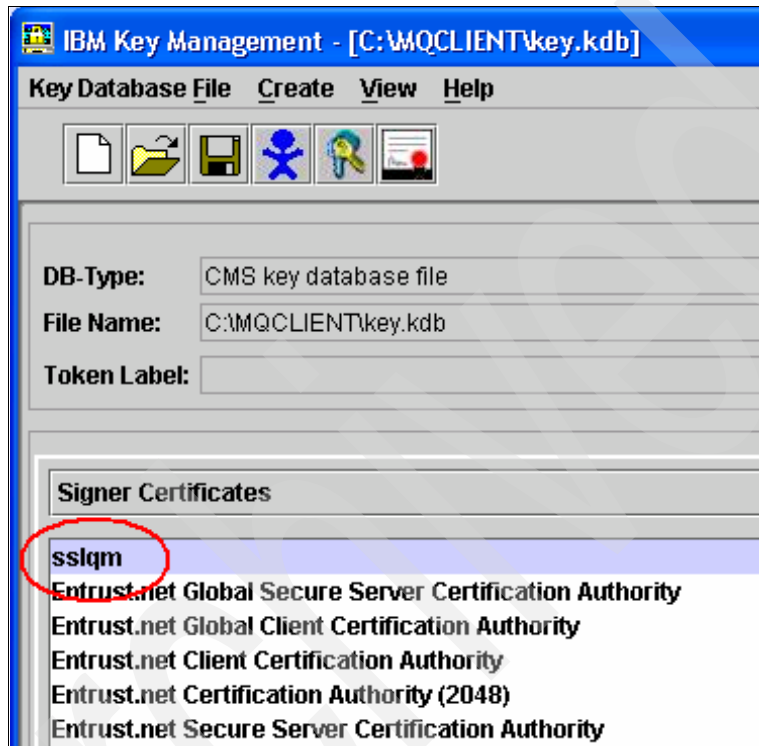


Figure 2-8 Certification authority certificate

9. Close the file (**Key Database File** → **Close**).
10. Close the iKeyman GUI.
11. Return to the command prompt window.

We created a certificate store for the WebSphere MQ client in C:\MQCLIENT, but we have not *told* the client where to find it. To do this, set the MQSSLKEYR environment variable, as shown in Example 2-9.

Example 2-9 Key repository environment variable

```
C:\MQCLIENT>set mqsslkeyr=C:\MQCLIENT\key
C:\MQCLIENT>set mq
mqchllib=C:\MQCLIENT
mqchltab=AMQCLCHL.TAB
mqsslkeyr=C:\MQCLIENT\key
```

Note: You must not put the file extension (.kdb) in the MQSSLKEYR environment variable. This is often overlooked and is the source of most problems.

Do *not* close the command prompt window.

You might want to write a command file (called, for example SSL.cmd) to reestablish the environment variables when you open a new command prompt window, as shown in Example 2-10.

Example 2-10 Resetting environment variables

```
set mqchllib=C:\MQCLIENT
set mqchltab=AMQCLCHL.TAB
set mqsslkeyr=C:\MQCLIENT\key
set mq
```

At this point, the queue manager has a certificate and the client has the corresponding CA certificate. We are ready to set up server SSL (one-way) authentication.

2.4.4 Testing SSL server authentication

Change the client connection and server connection channel definitions for SSL.

Both channels need to specify the same cipherspec. In this example, we use TRIPLE_DES_SHA_US, although anything other than none will do.

Because we are only interested in server authentication (one-way SSL), the server connection channel must specify SSLCAUTH(OPTIONAL).

To alter the client and server connection definitions, use the **runmqsc** command (or WebSphere MQ Explorer) on the queue manager machine, as shown in Example 2-11.

Example 2-11 Setting channels for SSL server authentication

```
C:\>runmqsc SSLQM
5724-H72 (C) Copyright IBM Corp. 1994, 2004.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager SSLQM.
alter chl(SSL.CLIENTS) chltype(svrconn) sslciph(triple_des_sha_us)
sslcauth(optional)
      1 : alter chl(SSL.CLIENTS) chltype(svrconn)
sslciph(triple_des_sha_us) sslcauth(optional)
AMQ8016: WebSphere MQ channel changed.
alter chl(SSL.CLIENTS) chltype(clntconn) sslciph(triple_des_sha_us)
      2 : alter chl(SSL.CLIENTS) chltype(clntconn)
sslciph(triple_des_sha_us)
AMQ8016: WebSphere MQ channel changed.
end
      3 : end
2 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

Note that the **runmqsc** command folds to uppercase any string entered without quotation marks (it can be confusing that it does not show the echoed command folded). This is what we want in this case. If you use quotation marks for the cipherspec, you must enter the string in uppercase: `sslciph(TRIPLE_DES_SHA_US)`.

If using WebSphere MQ Explorer instead of the **runmqsc** command, remember to set Authentication of parties initiating connections to **Optional** in the server connection channel's SSL tab.

The change to the client connection channel updates the channel table (file `AMQCLCHL.TAB`) in your queue manager's `@ipcc` directory.

Copy the channel table (or FTP binary) to the client directory (`C:\MQCLIENT`) in the client machine.

We are ready to test the client with SSL server authentication. Perform the following steps:

1. On the client machine, switch to the command prompt window. Verify that the necessary environment variables are set, as shown in Example 2-12.

Example 2-12 Checking the environment variables

```
C:\MQCLIENT>set mq
mqchl1ib=C:\MQCLIENT
mqchl1tab=AMQCLCHL.TAB
mqsslkeyr=C:\MQCLIENT\key
```

2. Run **amqsputc** and **amqsgetc** to test the SSL connection, as shown in Example 2-13.

Example 2-13 Testing the SSL server authentication

```
C:\MQCLIENT>amqsputc Q1
Sample AMQSPUTO start
target queue is Q1
test msg 1
```

```
Sample AMQSPUTO end
```

```
C:\MQCLIENT>amqsgetc Q1
Sample AMQSGETO start
message <test msg 1>
no more messages
Sample AMQSGETO end
```

If the programs run successfully, it means that the client channel works with SSL.

We are now ready to set up client authentication (two-way SSL).

2.5 SSL client authentication

SSL client authentication means that the client has a certificate assigned, which it presents to the server when the channel starts.

To set up client authentication, we need to perform the following steps:

1. Create a self-signed certificate for the client.
2. Add the certification authority certificate to the queue manager.
3. Change the server connection channel to request client authentication.
4. Re-test.

2.5.1 Creating a self-signed certificate for the client

This is the same as creating a self-signed certificate for the queue manager. The only difference is that we run it from the client machine. To create a self-signed certificate for the client:

1. On the client machine, open a Windows command prompt and enter **strmqikm**.
2. Click the **Open** icon (or select **Key Database File** → **Open**).
3. Provide the following values (Figure 2-9):
 - Key database type: **CMS**
 - File Name: **key.kdb**
 - Location: **C:\MQCLIENT**

Click **OK**.

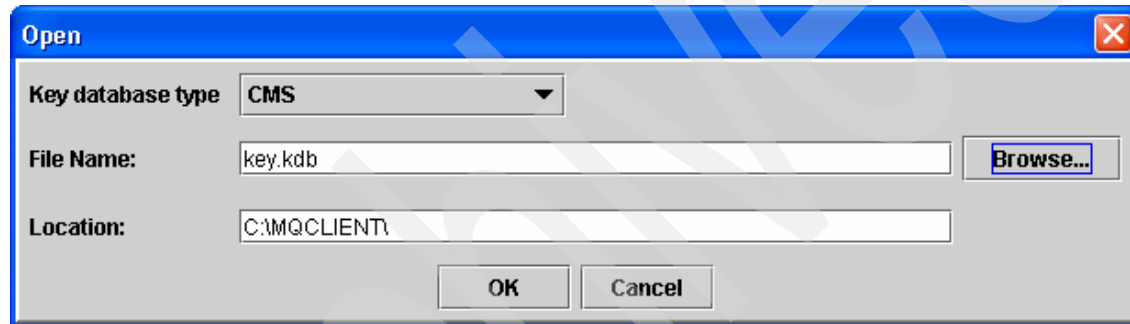


Figure 2-9 Open key repository

4. Enter the password when prompted. Click **OK**.
5. If the GUI shows the certification authority certificates, use the top-right menu to switch to viewing **Personal Certificates**. Click **New Self-Signed Certificate** (bottom-right corner).
6. Fill in the certificate attributes:
 - Key Label: **ibmwebsphermq** followed by the *lowercase* user ID
In our case: **ibmwebsphermqemir**
 - Common Name: **emir**
(You might have a different naming convention for the common name; feel free to enter any other value.)
 - Organization: **ibm** (or your company's name)Leave all other fields unchanged. Click **OK**.

You see the certificate listed, with an asterisk to the left of the name (the asterisk means that this is the default certificate for the key repository).

Now the WebSphere MQ client has a certificate, which it presents to the queue manager when the client connects. To validate the client's certificate, the queue manager needs the certification authority (CA) certificate.

To extract the CA certificate:

1. Click **Extract Certificate** (bottom-right corner).
2. Enter the following values:

- File Name: `<user ID>.arm`

In our case: `emir.arm`

- Location: `C:\MQCLIENT\`

Click **OK**.

This creates a file called `<user ID>.arm` in `C:\MQCLIENT`, as shown in Example 2-14.

Example 2-14 Extracted certification authority certificate

```
C:\MQCLIENT>dir
Volume in drive C is IBM_PRELOAD
Volume Serial Number is F48A-0639

Directory of C:\MQCLIENT

27/06/2006  15:56    <DIR>          .
27/06/2006  15:56    <DIR>          ..
27/06/2006  15:12             4,008 AMQCLCHL.TAB
27/06/2006  15:56             694 emir.arm
27/06/2006  15:54             80 key.crl
27/06/2006  15:54          125,080 key.kdb
27/06/2006  15:54             80 key.rdb
27/06/2006  13:58          129 key.sth
27/06/2006  14:31             92 SSL.cmd
27/06/2006  13:24          694 sslqm.arm
```

3. Close the repository (**Key Database File** → **Close**).
4. Close the iKeyman GUI.

Now, you must transfer the CA Certificate file to the queue manager's key repository.

2.5.2 Installing the CA part in the queue manager's key repository

This task consists of the following steps, executed on the queue manager machine:

1. Copy or transfer the CA (<user ID>.arm) file to the client.
2. Add the CA file to the queue manager's key repository.

Perform the following steps:

1. Copy (or FTP ASCII) the CA certificate file (<user ID>.arm) from the client's C:\MQCLIENT directory to the queue manager's ssl directory. This is <MQdir>\Qmgrs\SSLQM\ssl.
2. On the queue manager machine, open a command prompt window.
Enter **strmqikm**.
3. Click the **Open** icon (or select **Key Database File** → **Open**).
4. Open the key.kdb file in <MQdir>\Qmgrs\SSLQM\ssl. Click **OK**.
5. Enter the password when prompted. Click **OK**.
6. After opening the key repository, the GUI shows the Personal Certificates. Switch to viewing **Signer Certificates**.
Click **Add** (top-right corner).
7. In the next window, enter:
 - Certificate file name: <user ID>.arm
 - Location: <MQdir>\Qmgrs\SSLQM\sslClick **OK**.

8. When prompted to enter a label for the certificate, enter the client's user ID. (You can enter any name you want; just make sure it lets you identify the CA certificate as belonging to the client.) Click **OK**.

The certificate now appears in the Signer Certificates repository, as shown in Figure 2-10.

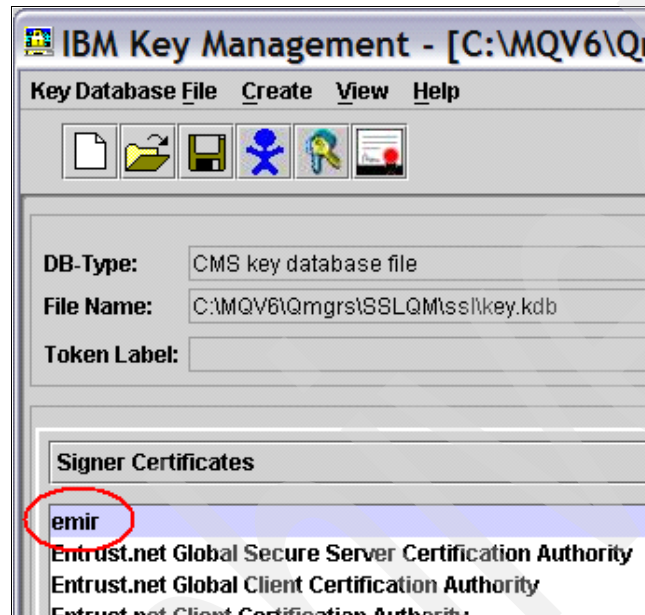


Figure 2-10 Certification authority certificate

9. Close the file (**Key Database File** → **Close**).
10. Close the iKeyman GUI.

At this point, the client has a certificate and the queue manager has the corresponding CA certificate. We are ready to set up client SSL (two-way) authentication.

2.5.3 Testing SSL client authentication

To enable client authentication, the server connection channel must specify SSLCAUTH(REQUIRED).

Start **runmqsc** (or the WebSphere MQ Explorer) on the queue manager machine, as shown in Example 2-15.

Example 2-15 Setting channels for SSL client authentication

```
C:\>runmqsc SSLQM
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2004.  ALL RIGHTS RESERVED.  
Starting MQSC for queue manager SSLQM.
```

```
alter chl(SSL.CLIENTS) chltype(svrconn) sslcauth(required)
```

```
1 : alter chl(SSL.CLIENTS) chltype(svrconn) sslcauth(required)  
AMQ8016: WebSphere MQ channel changed.  
end
```

```
2 : end
```

```
One MQSC command read.
```

```
No commands have a syntax error.
```

```
All valid MQSC commands were processed.
```

We are ready to test the client again:

1. On the client machine, switch to the command prompt window.
2. Verify that the necessary environment variables are set, as shown in Example 2-16.

Example 2-16 Checking the environment variables

```
C:\MQCLIENT>set mq  
mqchl1lib=C:\MQCLIENT  
mqchl1tab=AMQCLCHL.TAB  
mqsslkeyr=C:\MQCLIENT\key
```

3. Run **amqsputc** and **amqsgetc** to test the SSL connection, as shown in Example 2-17.

If **amqsputc** fails with MQCONN ended with reason code 2393 or MQCONN ended with reason code 2059, stop and start the queue manager.

Example 2-17 Testing the SSL client authentication

```
C:\MQCLIENT>amqsputc Q1
```

```
Sample AMQSPUTO start
```

```
target queue is Q1
```

```
test two-way auth
```

```
Sample AMQSPUTO end
```

```
C:\MQCLIENT>amqsgetc Q1
```

```
Sample AMQSGET0 start  
message <test two-way auth>  
no more messages  
Sample AMQSGET0 end
```

If the programs run successfully, it means that the client channel works with SSL.

This concludes the setup of a WebSphere MQ SSL Windows client with mutual authentication.

WebSphere MQ SSL on z/OS, AIX 5L, and Windows

When users set up WebSphere MQ SSL channels that do not cross company boundaries, the company can act as its own certification authority and create and dispense certificates for its queue managers.

z/OS installations that act as certificate authorities typically use RACF to generate and sign certificates.

This chapter provides a step-by-step guide to setting up a WebSphere MQ SSL configuration that uses RACF on z/OS to create certificates for z/OS, AIX 5L, and Windows queue managers.

3.1 Introduction

Figure 3-1 shows a diagram of the WebSphere MQ SSL configuration we want to create.

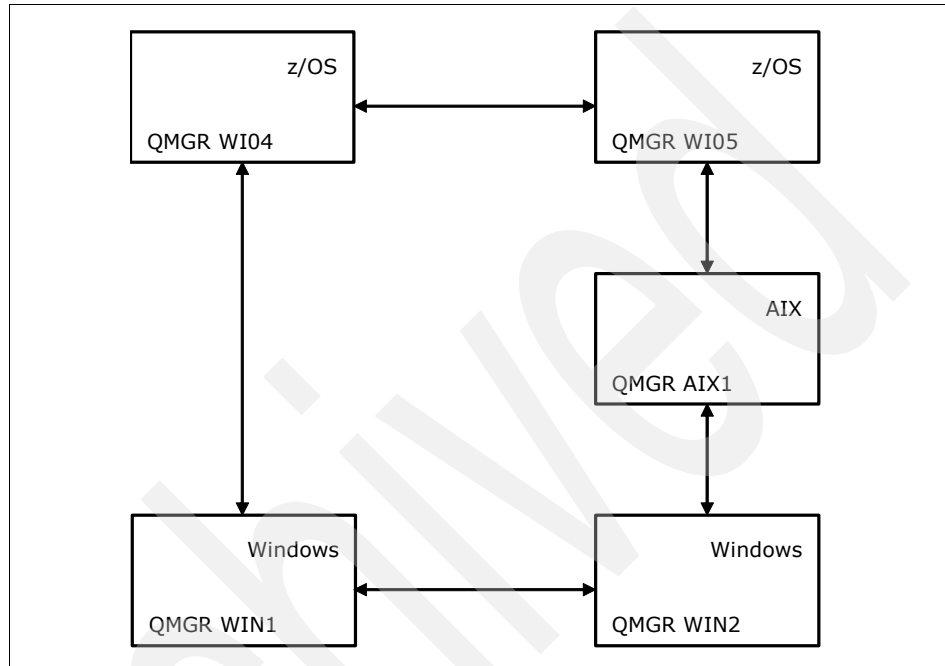


Figure 3-1 Queue manager configuration

We build the configuration in the following order:

1. Create certification authority certificates. See 3.2, “Certification authority setup” on page 47.
2. Set up SSL channels between two z/OS queue managers. See 3.3, “z/OS to z/OS” on page 55.
3. Set up SSL channels between a z/OS queue manager and a Windows queue manager. See 3.4, “z/OS to Windows” on page 71.
4. Set up SSL channels between a z/OS queue manager and an AIX 5L queue manager. See 3.5, “z/OS to AIX 5L” on page 84.
5. Set up SSL channels between an AIX 5L queue manager and a Windows queue manager. See 3.6, “AIX 5L to Windows” on page 96.
6. Set up SSL channels between two Windows queue managers. See 3.7, “Windows to Windows” on page 105.

3.2 Certification authority setup

This section explains how to set up a certification authority using RACF.

Certification paths and naming conventions

Before we can create certificates for a queue manager, we must have a certification authority certificate that we use to sign, that is, authenticate, the queue manager certificates.

Each certificate has a unique label. In this example, we use the certificate labels listed in Table 3-1. (You probably want to use labels that better suit your organization.)

Table 3-1 Certificate naming convention

Type	Label	Description
Root	MQ Root CA	Self-signed certificate. All other certificates are signed with this.
Personal (for z/OS queue managers)	ibmWebSphereMQqmgr_name	Certificate for a z/OS queue manager. Note that the label must follow the format shown.
Personal (for Windows and UNIX® queue managers)	ibmwebspheremqmgr_name	Certificate for Windows and UNIX queue managers. The label is similar to that of z/OS, but must be all lowercase.

Each certificate has an associated distinguished name: an X.509-format name that uniquely identifies the certificate owner.

An example of an X.509 distinguished name is:

- ▶ Common name: Emir Garza
- ▶ Title: IT Specialist
- ▶ Organizational unit: Software Group Services
- ▶ Organization: IBM
- ▶ Locality: Hursley
- ▶ State or province: Hampshire
- ▶ Country: UK

Only the common name and organization are required for SSL certificates; all other fields are optional. Our naming convention for distinguished name is listed in Table 3-2. (Again, you might prefer to follow conventions that better suit your organization.)

Table 3-2 Naming convention used here

	Root	Personal
Common name	SWGS WMQ	<i>queue manager name</i>
Organization	IBM	IBM

3.2.1 Creating a root certificate

To create a root certificate:

1. Log in to z/OS. Go to **ISPF option 6** (TSO commands).
2. Enter **racdcert** (without arguments) to check whether you are authorized to work with certificates. You get one of these two replies:

No certificate information was found for user userid

Digital certificate information for user <your TSO userid>:
[list of certificates follows]

If you are not authorized, you get the following message:

You are not authorized to issue the RACDCERT command.

RACDCERT also gives the following information on SYSLOG:

```
ICH408I USER(userid) GROUP(TSOUSER ) NAME(GARZA, E (EMIR) )
FULL VIOLATION ON COMMAND RACDCERT
```

Ask your systems programmer to give you authority to **racdcert**. If that is not possible, ask your system programmer to create the certificates for you.

3. Go to the RACF ISPF panel. (How you do that is different in each system. Ask your system programmer if it is not obvious from the ISPF main menu.) You get the panel shown in Example 3-1. What you need to type is in **bold** in the examples. Type 7. Press Enter.

Example 3-1 Main RACF menu

```

                                RACF - SERVICES OPTION MENU

OPTION ==> 7

SELECT ONE OF THE FOLLOWING:

    1  DATA SET PROFILES
```

- 2 GENERAL RESOURCE PROFILES
 - 3 GROUP PROFILES AND USER-TO-GROUP CONNECTIONS
 - 4 USER PROFILES AND YOUR OWN PASSWORD
 - 5 SYSTEM OPTIONS
 - 6 REMOTE SHARING FACILITY
 - 7 DIGITAL CERTIFICATES AND KEY RINGS
 - 99 EXIT
-

4. You see the panel shown in Example 3-2. Type 1. Press Enter.

Example 3-2 Digital Certificates and Related Services

RACF - Digital Certificates and Related Services
OPTION ==> 1

Select one of the following:

Digital Certificate Services

1. **Generate a certificate and a public/private key pair.**
2. Create a certificate request.
3. Write a certificate to a data set.
4. Add, Alter, Delete, or List certificates or check whether a digital certificate has been added to the RACF database and associated with a user ID.
5. Renew a certificate.

Key Ring Services

6. Create, List, or Delete an entire key ring or Connect or Remove a certificate to/from a key ring.

Certificate Name Filtering Services

7. Add, Alter, Delete, or List certificate name filters associated with a user ID.
-

5. Fill in the values to generate a digital certificate.

Fill in the values shown in Example 3-3. Press PF8 (down) to scroll to the next panel.

Example 3-3 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	_____	=> _	=> X

Enter the name of the data set which contains the PKCS10 request data:

More: +

Enter the following information about the Signing Authority:
(Required if a PKCS10 data set name was entered above)

	(default) Personal	or Site	Certificate or Authority
Signing Certificate Type:	_	=> _	=> _
and			
Label(in quotes):	_____		

Enter the Subject's X.509 Distinguished Name:

Common Name(in quotes): (ex: 'John Q. Public')
'SWGS WMQ'

Title(in quotes): (ex: 'Systems Programmer')

Fill in the values shown in Example 3-4. Press PF8 to scroll down.

Example 3-4 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	_____	=> _	=> X

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Organizational Unit(in quotes): (ex: 'S390','MVS')

Organization(in quotes): (ex: 'IBM')
'IBM'

Locality(in quotes): (ex: 'Poughkeepsie')

State/Province(in quotes): (ex: 'New York')

There is nothing to enter in the next panel, as shown in Example 3-5. Press PF8 (down) to scroll to the next panel.

Example 3-5 Generate a Digital Certificate

RACF - Generate a Digital Certificate
COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	_____	=> _	=> X

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Country(in quotes): (ex: 'US')

Enter the decimal size of the private key:

_____ (Default is 1024)

Enter any character to use IBM's crypto service provider:

_ Integrated Cryptographic Support Facility (ICSF).

Enter when the certificate is valid:

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)

Fill in the values shown in Example 3-6.

By default, certificates are valid for one year from creation. We gave the CA certificates an arbitrarily long life. Usage is Certsign; we only use this certificate to sign others.

Press Enter to create the certificate.

Example 3-6 Generate a Digital Certificate

```
RACF - Generate a Digital Certificate
COMMAND ==>

                Personal      Certificate
                (user ID)    or Site    or Authority
Certificate Type => _____ => _    => X

Enter the name of the data set which contains the PKCS10 request data:
_____

More:      - +
NOTBEFORE Date: YYYY-MM-DD (Default is current local date)
NOTBEFORE Time: HH:MM:SS   (Default is 00:00:00)

NOTAFTER Date: 2025-12-31 (Default is 1 year after NOTBEFORE)
NOTAFTER Time: 23:59:59   (Default is 23:59:59)

Enter the label name that is to be assigned to this certificate:

Label(in quotes): 'MQ Root CA'

Enter the key usage (more than one may be chosen):

_ Handshake - identification during security handshakes
_ Data encrypt - encrypting data
_ Docsign - legally binding signatures
x Certsign - signs other certificates and CRLs
```

Note: Note that you do not get positive feedback if it worked. You only get a message if there is an error.

3.2.2 Checking if the certificate was created

To check if the certificate was created:

1. From the RACF main certificates panel, select option 4, as shown in Example 3-7. Press Enter.

Example 3-7 Digital Certificates and Related Services

```
      RACF - Digital Certificates and Related Services
OPTION ==> 4

Select one of the following:

Digital Certificate Services
  1. Generate a certificate and a public/private key pair.
  2. Create a certificate request.
  3. Write a certificate to a data set.
  4. Add, Alter, Delete, or List certificates or
    check whether a digital certificate has been added to
    the RACF database and associated with a user ID.
  5. Renew a certificate.

Key Ring Services
  6. Create, List, or Delete an entire key ring or
    Connect or Remove a certificate to/from a key ring.

Certificate Name Filtering Services
  7. Add, Alter, Delete, or List certificate name filters
    associated with a user ID.
```

2. In the panel shown in Example 3-8, select 3. Press Enter.

Example 3-8 Digital Certificate Services Main Panel

```
      RACF - Digital Certificate Services Main Panel
OPTION ==> 3

      Personal      Certificate
      (user ID)      Authority
For Certificate Type _____ or _ or x

More:      +

Select one of the following options:

1. Add a digital certificate to the RACF database.

2. List information for certificates and be given the
   opportunity to change or delete them.
```

3. List a certificate using a filter and be given the opportunity to change the trust status, label, or delete it.
4. Check whether a digital certificate has been added to the RACF database and associated with a user ID, by entering a data set name: _____

Password for PKCS12 format data set(in quotes):

=>	<=
=>	<=
=>	<=
=>	<=

-
3. In the next panel (Example 3-9), enter the label of the certificate you just created. Press Enter.

Example 3-9 Digital Certificate List Filtering Panel

RACF - Digital Certificate List Filtering Panel
 COMMAND ==> 1

For: CERTAUTH

Select one of the following to filter the certificates

1. List by Label(in quotes):==> 'MQ Root CA' <==
2. List by Serial Number:==>

<==

Issuers Distinguished Name:(in quotes) ==>

<==

You see a panel similar to the one shown in Example 3-10.

Example 3-10 Change Status/Delete Digital Certificate

```
RACF - Change Status/Delete Digital Certificate
COMMAND ==>

Digital certificate information for: CERTAUTH

Enter A to Alter, D to Delete, next to the Label field,
then press PF3 to process.

More:      +

Action Certificate information
-----

- Label:MQ Root CA
  Certificate ID:2QiJmZmDhZmjgdTYQNmWlqNAw8FA
  Status:TRUST
  Start Date:2006/06/28 00:00:00
  End Date: 2025/12/31 23:59:59
  Serial Number:00

Issuer's Name:CN=SWGS WMQ.0=IBM

Subject's Name:CN=SWGS WMQ.0=IBM
```

This concludes the creation of the root CA certificate.

3.3 z/OS to z/OS

This section explains how to set up SSL channels between two z/OS queue managers.

Assumptions

The instructions that follow assume the configuration listed in Table 3-3.

Table 3-3 Queue manager configuration

Queue manager name	WI04	WI05
Queue manager user ID	EMIR	EMIR
CHINIT user ID	EMIR	EMIR
Host name	winmvs60	winmvs60
Listener port	20004	20005
Key ring name	WI04	WI05

Ensure that you have a working sender/receiver channel pair between the two queue managers.

Example CSQUTIL JCL to set up channels

This section provides example CSQUTIL JCL to set up the channels.

Example 3-11 provides an example for queue manager WI04.

Example 3-11 JCL for WI04

```
//CSQUTIL EXEC PGM=CSQUTIL,PARM=WI04
//STEPLIB DD DISP=SHR,DSN=MQM.V600.SCSQANLE
//          DD DISP=SHR,DSN=MQM.V600.SCSQAUTH
//          DD DISP=SHR,DSN=MQM.V600.SCSQLOAD
//CSQUCMD DD *
DEF QL(WI05) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WI04.WI05)
DEF CHL(WI04.WI05) CHLTYPE(SDR) REPLACE +
    CONNAME('WINMVS60(20005)') XMITQ(WI05)
DEF CHL(WI05.WI04) CHLTYPE(RCVR) REPLACE
DEF QL(WI04.Q1) REPLACE
DEF QR(WI05.Q1) REPLACE +
    RNAME(WI05.Q1) RQMNAME(WI05)
//SYSIN DD *
    COMMAND TGTQMGR(WI04) DDNAME(CSQUCMD)
//SYSPRINT DD SYSOUT=*
```

Example 3-12 provides an example for queue manager WI05.

Example 3-12 JCL for WI05

```
//CSQUTIL EXEC PGM=CSQUTIL,PARM=WI05
//STEPLIB DD DISP=SHR,DSN=MQM.V600.SCSQANLE
//          DD DISP=SHR,DSN=MQM.V600.SCSQAUTH
//          DD DISP=SHR,DSN=MQM.V600.SCSQLOAD
//CSQUCMD DD *
DEF QL(WI04) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WI05.WI04)
DEF CHL(WI05.WI04) CHLTYPE(SDR) REPLACE +
    CONNAME('WINMVS60(20004)') XMITQ(WI04)
DEF CHL(WI04.WI05) CHLTYPE(RCVR) REPLACE
DEF QL(WI05.Q1) REPLACE
DEF QR(WI04.Q1) REPLACE +
    RNAME(WI04.Q1) RQMNAME(WI04)
//SYSIN DD *
    COMMAND TGTQMGR(WI05) DDNAME(CSQUCMD)
//SYSPRINT DD SYSOUT=*
```

Task summary

To set up SSL channels between two z/OS queue managers:

1. Enable SSL on the queue managers.
2. Create certificates and assign them to each queue manager.
3. Alter the channel definitions to activate SSL.

The following instructions apply to both queue managers. You might prefer to do the complete setup for one queue manager and then come back to this point and repeat these instructions for the other.

3.3.1 Enabling SSL on the queue managers

This section describes how to enable SSL on the queue managers.

Creating a key repository (“key ring”) for the queue manager

Key rings contain the certificates that queue managers use. Perform the following steps:

1. Using the RACF panels, select option 7 (DIGITAL CERTIFICATES AND KEY RINGS), and then option 6 (Create, List, or Delete an entire key ring), as shown in Example 3-13. Press Enter.

Example 3-13 Digital Certificates and Related Services

```
RACF - Digital Certificates and Related Services
OPTION ==> 6
```

Select one of the following:

Digital Certificate Services

1. Generate a certificate and a public/private key pair.
2. Create a certificate request.
3. Write a certificate to a data set.
4. Add, Alter, Delete, or List certificates or check whether a digital certificate has been added to the RACF database and associated with a user ID.
5. Renew a certificate.

Key Ring Services

- 6. Create, List, or Delete an entire key ring or**
Connect or Remove a certificate to/from a key ring.

...

2. You see the panel shown in Example 3-14. Select 1. EMIR is the user ID of the channel initiator address space. Press Enter.

Example 3-14 Digital Certificate Key Ring Services

```
RACF - Digital Certificate Key Ring Services
OPTION ==> 1
```

For user: **EMIR**

Enter one of the following at the OPTION line:

- 1 Create a new key ring**
 - 2 Delete an existing key ring
 - 3 List existing key ring(s)
 - 4 Connect a digital certificate to a key ring
 - 5 Remove a digital certificate from a key ring
-

3. Enter the key ring name in the next panel, as shown in Example 3-15. We use the queue manager name as the key ring name; you can use a different convention. Press Enter.

Example 3-15 Digital Certificate Key Ring Name

RACF - Digital Certificate Key Ring Name
COMMAND ==>

Enter a ring name:

WI04 _____

A ring name may not be a single asterisk * and
blanks are not allowed.

You see this ISPF message:

Key Ring WI04 has been successfully added

Connecting the CA certificate to the key ring

To connect the CA certificate to the key ring:

1. From the Key Ring Services panel (Example 3-16), select 4 (Connect a digital certificate to a key ring). Press Enter.

Example 3-16 Digital Certificate Key Ring Services

RACF - Digital Certificate Key Ring Services
OPTION ==> **4**

For user: **EMIR**

Enter one of the following at the OPTION line:

- 1 Create a new key ring
 - 2 Delete an existing key ring
 - 3 List existing key ring(s)
 - 4 Connect a digital certificate to a key ring**
 - 5 Remove a digital certificate from a key ring
-

2. In the next panel (Example 3-17), enter the Ring Name. Press Enter.

Example 3-17 Connect a Digital Certificate to a Key Ring

RACF - Connect a Digital Certificate to a Key Ring

COMMAND ==>

Ring Owner: EMIR

Ring Name: **WI04**_____

	Personal (user ID)	or Site	or Certificate Authority
Certificate Type =>	_____	=> _	=> x
Label name: 'MQ Root CA'	_____		(in quotes)
Usage	=> _	=> _	=> x
Default	=> _	(blank defaults to NO)	

You get this message:

Certificate successfully connected to key ring.

Assigning the key ring and enabling the SSL tasks

To assign the key ring and enable SSL tasks:

1. Using the WebSphere MQ ISPF panels, select action 3 (Alter), object type manager. Press Enter.
2. Then press PF8 until you see the panel shown in Example 3-18. Enter the values shown. Press Enter to make the change.

Example 3-18 Alter a Queue Manager

Alter a Queue Manager - 3

Command ==>

Press F7 or F8 to see other fields, or Enter to alter queue manager.

Queue manager name : WI04

SSL authentication namelist
name
SSL key repository
. **WI04**

SSL server tasks **5** 0 - 9999

Important: These changes are not permanent. Depending on your CSQINP2 setup, your queue manager might revert to no key ring and zero SSL tasks at the next restart.

Making these changes permanent

To make these changes permanent:

- 1. From the SDSF DA panel, enter ? next to the queue manager job, as shown in Example 3-19.

Example 3-19 SDSF display active

SDSF	DA	MV60	MV60	PAG	0	SIO	3	CPU	2
COMMAND INPUT ==>									
NP	JOBNAME	StepName	ProcStep	JobID	Owner				
?	WI04MSTR	WI04MSTR	PROCSTEP	STC01815	EMIR				
	WI04CHIN	WI04CHIN	PROCSTEP	STC01816	EMIR				

- 2. Press Enter and select JES2JCL, as shown in Example 3-20.

Example 3-20 Queue manager JES2 files

SDSF	JOB	DATA	SET	DISPLAY	-	JOB	WI04MSTR	(STC01815)
COMMAND INPUT ==>								
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	
	JESMSGLG	JES2		2	EMIR		C	
s	JESJCL	JES2		3	EMIR		C	
	JESYSMSG	JES2		4	EMIR		C	
	CSQOUT1	WI04MSTR		101	EMIR		A	
	CSQOUT2	WI04MSTR		102	EMIR		A	

3. Locate the xxxxINYG member of the CSQINP2 concatenation, as shown in Example 3-21.

Example 3-21 Queue manager JES2 JCL

```
XXCSQINP2 DD DSN=WI04BRK.MQSI.JCL(WI04INSG),DISP=SHR
XX        DD DSN=WI04BRK.MQSI.JCL(WI04INSX),DISP=SHR
XX        DD DSN=WI04BRK.MQSI.JCL(WI04INSS),DISP=SHR
XX        DD DSN=WI04BRK.MQSI.JCL(WI04INYG),DISP=SHR
XX*       DD DSN=WI04BRK.MQSI.JCL(WI04INYC),DISP=SHR
XX*       DD DSN=WI04BRK.MQSI.JCL(WI04INYD),DISP=SHR
XX        DD DSN=WI04BRK.MQSI.JCL(WI04DISP),DISP=SHR
```

4. Edit the INYG member. It contains the ALTER QMGR statement shown in Example 3-22.

Example 3-22 INYG member

```
ALTER QMGR +
...
SSLKEYR( ' ' ) +
...
SSLTASKS( 0 ) +
...
```

5. Change the SSLKEYR and SSLTASKS attributes as shown in Example 3-23.

Example 3-23 INYG member (updated)

```
ALTER QMGR +
...
SSLKEYR( 'WI04' ) +
...
SSLTASKS( 5 ) +
...
```

6. Give the CHINIT user ID RACF access to the key ring.

You *must* do this (or ask your system programmer to do it for you), even if your CHINIT user ID has the RACF SPECIAL attribute. See Example 3-2 on page 49.

Example 3-24 RACF access to key ring

```
rdef FACILITY IRR.DIGTCERT.** UACC(NONE)
PE IRR.DIGTCERT.** CL(FACILITY) ID(EMIR) ACC(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

7. If *necessary*, update the CHINIT JCL. The channel initiator needs access to the z/OS Cryptographic Services. These are in the SGSKLOAD library (before z/OS 1.6) or SIEALNKE, usually SYS1.SIEALNKE, from z/OS 1.6. Ask your systems programmer if SGSKLOAD (or SIEALNKE) is in LPA. If it is, there is nothing to do. If it is *not*, add it to STEPLIB in the channel initiator JCL.
8. Stop and start the channel initiator. You see the following message when the channel initiator restarts:

```
CSQX151I =WI05 CSQXSSLI 5 SSL server subtasks started, 0 failed
```

This completes the initial queue manager SSL setup. Repeat these steps for the other queue manager.

3.3.2 Creating the queue manager certificate

To create the queue manager certificate:

1. From the RACF certificates panel, select 1 (Generate a certificate).
2. Fill in the Generate a Digital Certificate panel.

Enter the values shown in Example 3-25. Press PF8.

Example 3-25 Generate a Digital Certificate

```

                                RACF - Generate a Digital Certificate
COMMAND ==>

                                Personal      Certificate
                                (user ID)    or Site    or Authority
Certificate Type => emir_____ => _      => _

Enter the name of the data set which contains the PKCS10 request data:
_____

More:      +

Enter the following information about the Signing Authority:
(Required if a PKCS10 data set name was entered above)

                                (default)
Signing      Personal    or Site    Certificate
Certificate Type:  _      => _      => x
and
Label(in quotes): 'MQ Root CA'_____

Enter the Subject's X.509 Distinguished Name:

```

Common Name(in quotes): (ex: 'John Q. Public')
'WI04'

Title(in quotes): (ex: 'Systems Programmer')

We are creating a certificate signed with the root certificate we created earlier (that is, the certificate is encrypted with the certification authority private key). Fill in the values shown in Example 3-26 and press PF8.

Example 3-26 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Organizational Unit(in quotes): (ex: 'S390','MVS')

Organization(in quotes): (ex: 'IBM')

'IBM'

Locality(in quotes): (ex: 'Poughkeepsie')

State/Province(in quotes): (ex: 'New York')

There is nothing to enter on the next panel, as shown in Example 3-27. Press PF8

Example 3-27 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR_____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Country(in quotes): (ex: 'US')

Enter the decimal size of the private key:

_____ (Default is 1024)

Enter any character to use IBM's crypto service provider:

_ Integrated Cryptographic Support Facility (ICSF).

Enter when the certificate is valid:

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)

Notice the label. We must use a label of the format `ibmWebSphereMQ` followed by the queue manager name. The case is significant. Enter the label exactly as shown in Example 3-28 (uppercase W, S, MQ, and uppercase queue manager name). Press Enter to create the certificate.

Example 3-28 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR_____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +
NOTBEFORE Date: YYYY-MM-DD (Default is current local date)
NOTBEFORE Time: HH:MM:SS (Default is 00:00:00)

NOTAFTER Date: **2025-12-31** (Default is 1 year after NOTBEFORE)
NOTAFTER Time: HH:MM:SS (Default is 23:59:59)

Enter the label name that is to be assigned to this certificate:

Label(in quotes): 'ibmWebSphereMQWI04' _____

Enter the key usage (more than one may be chosen):

☒ Handshake - identification during security handshakes
☐ Data encrypt - encrypting data
☐ Docsign - legally binding signatures
☐ Certsign - signs other certificates and CRLs

You only receive a response if the command fails.

3.3.3 Connecting the certificate to the key ring

Connect the newly created certificate to the queue manager's key ring:

1. From the RACF Certificates panel, select option 6 (Key ring services).
2. Fill in the panel as shown in Example 3-29.

Example 3-29 Digital Certificate Key Ring Services

RACF - Digital Certificate Key Ring Services
OPTION ==> 4
For user: **emir**_____

Enter one of the following at the OPTION line:

- 1 Create a new key ring
 - 2 Delete an existing key ring
 - 3 List existing key ring(s)
 - 4 Connect a digital certificate to a key ring
 - 5 Remove a digital certificate from a key ring
-

3. Emir is the user ID the queue manager and channel initiator run under.
Provide the values shown in Example 3-30. Press Enter.

Example 3-30 Connect a Digital Certificate to a Key Ring

```

RACF - Connect a Digital Certificate to a Key Ring
COMMAND ==>

Ring Owner: EMIR

Ring Name:
WI04_____
_____
_____
_____

Certificate Type => emir_____
                        Personal (user ID) or Site or Certificate Authority
                        => _ => _

Label name: 'ibmWebSphereMQWI04'_____ (in quotes)

Usage => x_____
                        Personal or Site or Certificate Authority
                        => _ => _

Default => _ (blank defaults to NO)

```

You see this message:

Certificate successfully connected to key ring.

4. Restart the channel initiator.
Check that the messages shown in Example 3-31 appears in the job log.

Example 3-31 Restart channel initiator

```

+CSQX080I =WI04 CSQXGIP SSLTASKS=5, SSLKEYC=0
+CSQX081I =WI04 CSQXGIP SSLKEYR=WI04
...
+CSQX151I =WI04 CSQXSSLI 5 SSL server subtasks started, 0 failed

```

This concludes the creation and assignment of the certificate to the queue manager. Remember to repeat these instructions for the other queue manager.

3.3.4 Altering the channel attributes

From the WebSphere MQ ISPF panels (Example 3-32), alter the sender channel attributes (see “Using WebSphere MQ Explorer” on page 70 for doing this using in the WebSphere MQ Explorer).

Example 3-32 Alter a Sender Channel

Alter a Sender Channel - 4

Command ==>

Press F7 or F8 to see other fields, or Enter to alter channel.

More: - +

Channel name : WI04.WI05
Disposition : QMGR WI04

SSL cipher specification . . NULL_MD5
SSL peer name
.

Header compression N and N=None, S=System
Message compression N and N=None, R=RLE,
and and F=ZlibFast, H=ZlibHigh

When the SSL cipher specification (called *cipherspec*) is set to anything other than blank, the receiver channel must present a certificate to the sender (so that the sender authenticates the receiver's identity). Optionally, the receiver channel can request that the sender also presents a certificate (thus achieving mutual authentication).

When the channels establish the SSL session, they exchange a randomly generated key that they use to sign and (optionally) to encrypt the messages.

A NULL_MD5 cipherspec means no encryption (NULL) and signing with the MD5 algorithm.

Alter the corresponding receiver channel on the other queue manager, as shown in Example 3-33.

Example 3-33 Alter a Receiver Channel

```
Alter a Receiver Channel - 3
Command ==>

Press F7 or F8 to see other fields, or Enter to alter channel.

Channel name . . . . . : WI04.WI05
Disposition . . . . . : QMGR    WI05

SSL cipher specification . . NULL_MD5
SSL peer name
. . . . .

SSL certificate required . . Y  Y=Yes, N=No

Header compression . . . . . N and . . . . . N=None, S=System
Message compression . . . . . N and . . . . . A=Any, N=None, R=RLE,
and . . . . . and . . . . . F=ZlibFast, H=ZlibHigh
and . . . . .
```

The SSL certificate required option is Yes by default. That is, the receiver wants to authenticate the sender's identity.

Repeat for the other sender/receiver channel pair (in this example, channel WI05.WI04).

Using WebSphere MQ Explorer

You can use WebSphere MQ Explorer instead of ISPF panels if you prefer. To use WebSphere MQ Explorer:

1. Connect your WebSphere MQ Explorer to the two queue managers and set the sender channel SSL properties as shown in Figure 3-2.

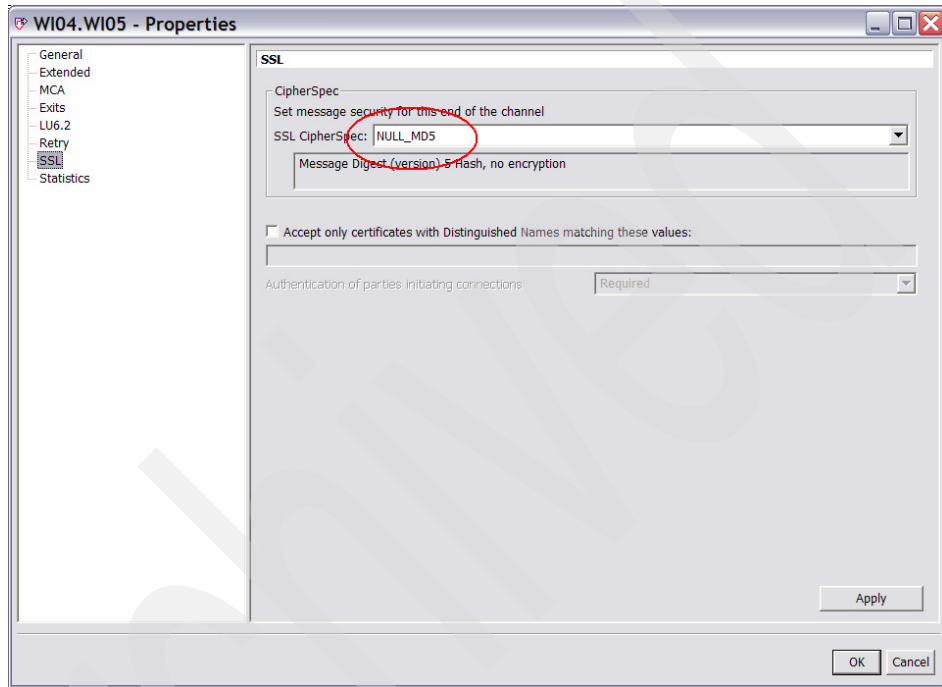


Figure 3-2 Sender channel on WI04

2. Alter the receiver channel SSL attributes as shown in Figure 3-3.

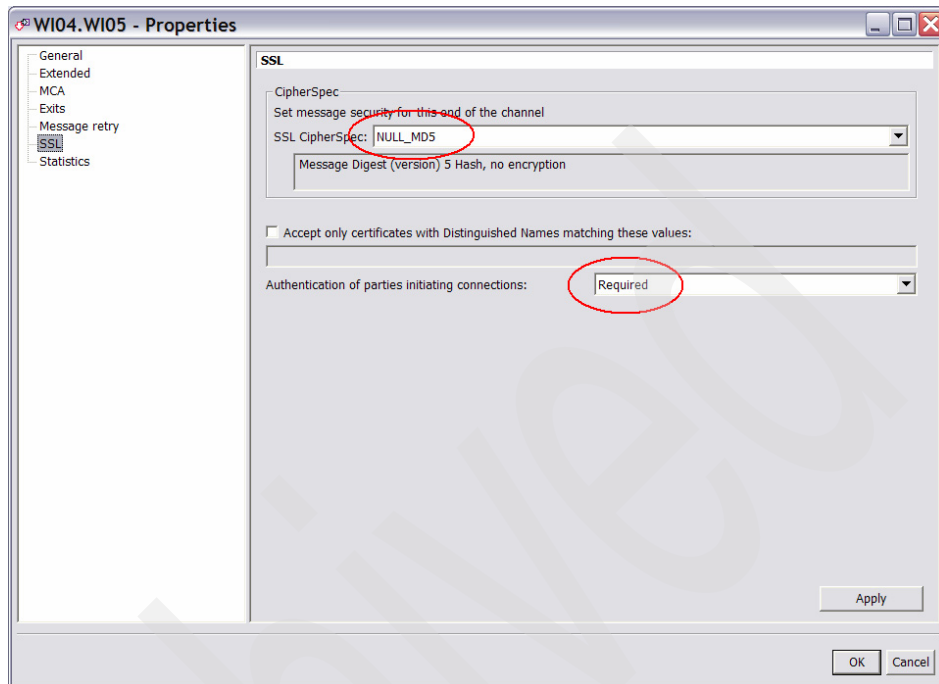


Figure 3-3 Receiver channel on WI05

3. Repeat the prior step for the channel WI05.WI04.
4. Stop and start the channels.
The channels should start, meaning they have successfully exchanged the SSL certificates.

This concludes the setup of SSL channels between two z/OS queue managers.

3.4 z/OS to Windows

This section describes how to set up SSL channels between a z/OS queue manager and a Windows queue manager.

Assumptions

The instructions that follow assume the configuration listed in Table 3-4.

Table 3-4 Queue manager configuration

Platform	z/OS	Windows
Queue manager name	WI04	WIN1
Queue manager user ID	EMIR	EMIR
CHINIT user ID	EMIR	N/A
Host name	winmvs60	garzae
Listener port	20004	20001

Ensure that you have a working sender/receiver channel pair between the two queue managers.

Example CSQUTIL JCL for WI04

Example 3-34 provides example CSQUTIL JCL for WI04.

Example 3-34 JCL for WI04

```
//CSQUTIL EXEC PGM=CSQUTIL,PARM=WI04
//STEPLIB DD DISP=SHR,DSN=MQM.V600.SCSQANLE
//          DD DISP=SHR,DSN=MQM.V600.SCSQAUTH
//          DD DISP=SHR,DSN=MQM.V600.SCSQLOAD
//CSQUCMD DD *
DEF QL(WIN1) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WI04.WIN1)
DEF CHL(WI04.WIN1) CHLTYPE(SDR) REPLACE +
    CONNAME('GARZAE(20001)') XMITQ(WIN1)
DEF CHL(WIN1.WI04) CHLTYPE(RCVR) REPLACE
DEF QL(WI04.Q1) REPLACE
DEF QR(WIN1.Q1) REPLACE +
    RNAME(WIN1.Q1) RQMNAME(WIN1)
//SYSIN DD *
    COMMAND TGTQMGR(WI04) DDNAME(CSQUCMD)
//SYSPRINT DD SYSOUT=*
```

Example MQSC for WIN1

Example 3-35 provides example MQSC for WIN1.

Example 3-35 MQSC for WIN1

```
DEF QL(WI04) REPLACE USAGE(XMITQ) TRIGGER +  
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WIN1.WI04)  
DEF CHL(WIN1.WI04) CHLTYPE(SDR) REPLACE +  
    CONNAME('WINMVS60(20004)') XMITQ(WI04)  
DEF CHL(WI04.WIN1) CHLTYPE(RCVR) REPLACE  
DEF QL(WIN1.Q1) REPLACE  
DEF QR(WI04.Q1) REPLACE +  
    RNAME(WI04.Q1) RQMNAME(WI04)
```

Task summary

To set up SSL channels between a z/OS and a Windows queue manager, we perform the tasks listed in Table 3-5.

Table 3-5 Task summary

Task	Comments
1. Enable SSL on the z/OS queue manager.	Already done when setting up SSL between two z/OS queue managers. See 3.3.1, “Enabling SSL on the queue managers” on page 57.
2. Create a certificate and assign it to the z/OS queue manager.	Already done when setting up SSL between two z/OS queue managers. See: ► 3.3.2, “Creating the queue manager certificate” on page 63 ► 3.3.3, “Connecting the certificate to the key ring” on page 66
3. Use RACF to create a certificate for the Windows queue manager.	See 3.4.1, “Creating a certificate for WIN1” on page 74.
4. Export the certificate from RACF.	See 3.4.2, “Exporting the certificate from RACF” on page 77.
5. Download (FTP) the certificate to Windows.	See 3.4.3, “Downloading the certificate to Windows” on page 78.
6. Create a key repository for the Windows queue manager.	See 3.4.4, “Creating a key repository for WIN1” on page 79.
7. Import the certificate in Windows.	See 3.4.5, “Importing the certificate” on page 79.
8. Alter the channel definitions to start using SSL.	See 3.4.6, “Altering the channel attributes” on page 83.

3.4.1 Creating a certificate for WIN1

To create a certificate for WIN1:

1. From the RACF certificates panel, select 1 (Generate a certificate).
2. Fill in the Generate a Digital Certificate panel.

Enter the values shown in Example 3-36. Press PF8.

Example 3-36 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	emir	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: +

Enter the following information about the Signing Authority:
(Required if a PKCS10 data set name was entered above)

	(default) Personal	or Site	Certificate or Authority
Signing Certificate Type:	_	=> _	=> x

and

Label(in quotes): **'MQ Root CA'** _____

Enter the Subject's X.509 Distinguished Name:

Common Name(in quotes): (ex: 'John Q. Public')

'WIN1' _____

Title(in quotes): (ex: 'Systems Programmer')

Enter the values shown in Example 3-37. Press PF8.

Example 3-37 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
--	-----------------------	---------	-----------------------------

Certificate Type => **EMIR**_____ => _ => _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Organizational Unit(in quotes): (ex: 'S390','MVS')

Organization(in quotes): (ex: 'IBM')

'IBM'_____

Locality(in quotes): (ex: 'Poughkeepsie')

State/Province(in quotes): (ex: 'New York')

There is nothing to enter in the next panel, as shown in Example 3-38. Press PF8.

Example 3-38 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR _____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Country(in quotes): (ex: 'US')

Enter the decimal size of the private key:

_____ (Default is 1024)

Enter any character to use IBM's crypto service provider:

_ Integrated Cryptographic Support Facility (ICSF).

Enter when the certificate is valid:

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)

Enter the values shown in Example 3-39. Press Enter.

Example 3-39 Generate a Digital Certificate

```
RACF - Generate a Digital Certificate
COMMAND ==>

                Personal      Certificate
                (user ID) or Site or Authority
Certificate Type => EMIR_____ => _ => _

Enter the name of the data set which contains the PKCS10 request data:
_____

More:      - +
NOTBEFORE Date: YYYY-MM-DD (Default is current local date)
NOTBEFORE Time: HH:MM:SS   (Default is 00:00:00)

NOTAFTER Date:  2025-12-31 (Default is 1 year after NOTBEFORE)
NOTAFTER Time:  HH:MM:SS   (Default is 23:59:59)

Enter the label name that is to be assigned to this certificate:

Label(in quotes): 'ibmwebspheremqwin1'_____

Enter the key usage (more than one may be chosen):

x Handshake    - identification during security handshakes
_ Data encrypt - encrypting data
_ Docsign      - legally binding signatures
_ Certsign     - signs other certificates and CRLs
```

You only receive a response if the command fails.

3.4.2 Exporting the certificate from RACF

To export the certificate from RACF:

1. From the RACF certificates panel, select option 3 (Write a certificate to a data set).
2. Fill in the Write a Certificate to a Data Set panel.
Enter the values shown in Example 3-40. RACF exports the key to a data set called userid.SSLKEY.WIN1. Press PF8.

Example 3-40 Export a certificate

RACF - Write a Certificate to a Data Set

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	emir _____	=> _	=> _

More: +

Enter the label name of the certificate:
'ibmwebspheremqw1'_____ (in quotes)

Enter the data set name to which the certificate is to be written:
sslkey.win1_____

Enter any character to indicate the Certificate Format Type:

- _ DER encoded X.509
- _ Base64 encoded X.509 (default)
- x** DER encoded PKCS12
- _ Base64 encoded PKCS12
- _ DER encoded PKCS7
- _ Base64 encoded PKCS7

Fill in the values shown in Example 3-41. Note that you must enter/reenter a password (in quotation marks). You need the password later when you import the certificate. Press Enter.

Example 3-41 Export a certificate

RACF - Write a Certificate to a Data Set

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR _____	=> _	=> _

- _ DER encoded X.509
- _ Base64 encoded X.509 (default)
- x** DER encoded PKCS12
- _ Base64 encoded PKCS12
- _ DER encoded PKCS7
- _ Base64 encoded PKCS7

Password for PKCS12 format data set (in quotes):

=> **'password' [not visible]** <=

Re-enter password to verify (in quotes):

=> **'password' [not visible]** <=

If the command worked, you see (ISPF option 3.4) the data set userid.SSLKEY.WIN1.

3.4.3 Downloading the certificate to Windows

Use FTP to download the exported certificate to Windows (Example 3-42). We place the exported certificate in the queue manager's ssl directory (<MQDir>\Qmgrs\WIN1\ssl).

Example 3-42 Download the certificate

```
C:\MQV6\Qmgrs\WIN1\ssl>ftp winmvs60
Connected to winmvs60.hursley.ibm.com.
220-FTPD1 IBM FTP CS V1R7 at WINMVS60.HURSLEY.IBM.COM, 14:14:50 on
2006-06-29.
220 Connection will close if idle for more than 5 minutes.
User (winmvs60.hursley.ibm.com:(none)): emir
331 Send password please.
Password:
230 EMIR is logged on. Working directory is "EMIR.".
ftp> cd sslkey
250 "EMIR.SSLKEY." is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp> get win1 WIN1.p12
200 Port request OK.
125 Sending data set EMIR.SSLKEY.WIN1
250 Transfer completed successfully.
ftp: 2316 bytes received in 0.02Seconds 115.80Kbytes/sec.
ftp> bye
221 Quit command received. Goodbye.
```

Note the binary file transfer and the file extension (p12) of the exported certificate.

3.4.4 Creating a key repository for WIN1

The following instructions show how to create a key repository for queue manager WIN1.

This is already described in detail in Chapter 1, “Connecting two Windows queue managers using SSL” on page 1, so this section omits the screen captures.

Perform the following steps:

1. Open a Windows command prompt and enter **strmqikm**. This starts the IBM Key Management (iKeyman) GUI.
2. 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\WIN1\ssl**
In this example: **C:\MQV6\Qmgrs\WIN1\ssl**Click **OK**.
3. Enter a password (and remember it, because you need it later) and select **Stash the password to a file?** Click **OK**.
4. You see a message saying that the password was encrypted and saved. Click **OK**.

You created a key repository for queue manager WIN1.

After creating the key repository, the GUI shows the installed certification authority certificates provided with iKeyman. Use the menu (top right) to switch to viewing **Personal Certificates**.

Do *not* close the iKeyman GUI.

3.4.5 Importing the certificate

We now import the certificate (file WIN1.p12 in **<MQDir>\Qmgrs\WIN1\ssl**). Perform the following steps:

1. Click the **Import** button (to the right of the Personal Certificates pane). This opens the Import Key window (Figure 3-4 on page 80).

2. Use the following values:
 - Change the Key file type to **PKCS12**.
 - Change the File Name to WIN1.p12
 - Change the Location to the directory where you placed the exported certificate (<MQDir>\Qmgrs\WIN1\ssl).
- Click **OK**.

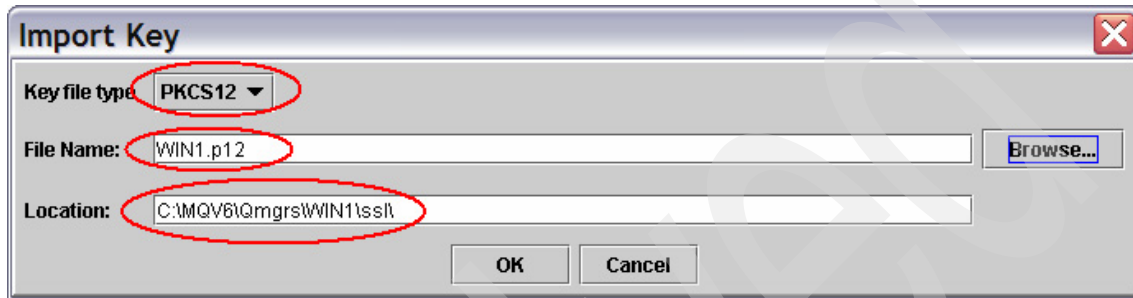


Figure 3-4 Import Key

3. This open the password window. Enter the password you gave when exporting the certificate from z/OS, and then click **OK**.

4. You have the option to change the labels (Figure 3-5). The queue manager (personal) certificate already has the correct label.

The certification authority certificate (which signs the personal certificate) is the one we created as MQ Root CA on z/OS. It appears with the label `0cn=swgs_wmq,o=ibm`. It was exported (and will be imported) together with the personal certificate.

There is no need to change either of these labels, so click **OK**.

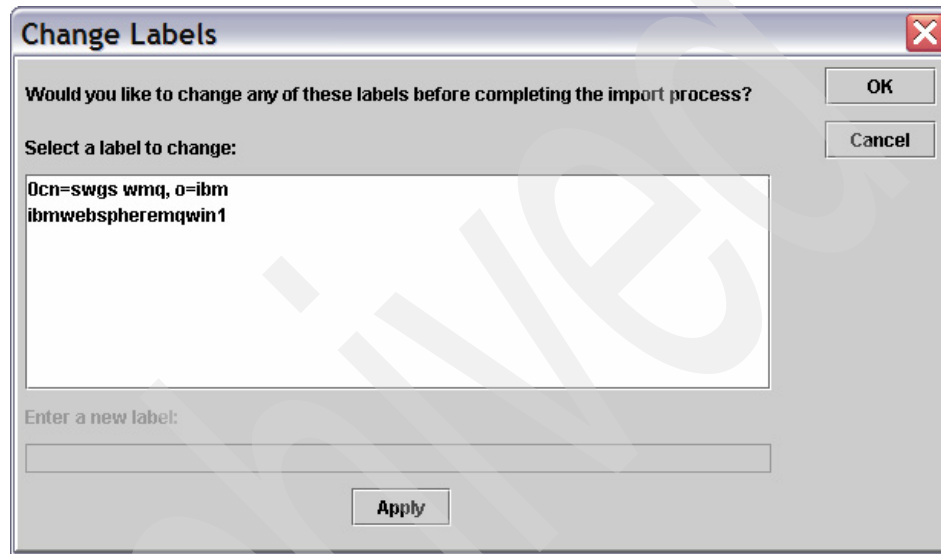


Figure 3-5 Change Labels

You see the queue manager's certificate shown in Figure 3-6.

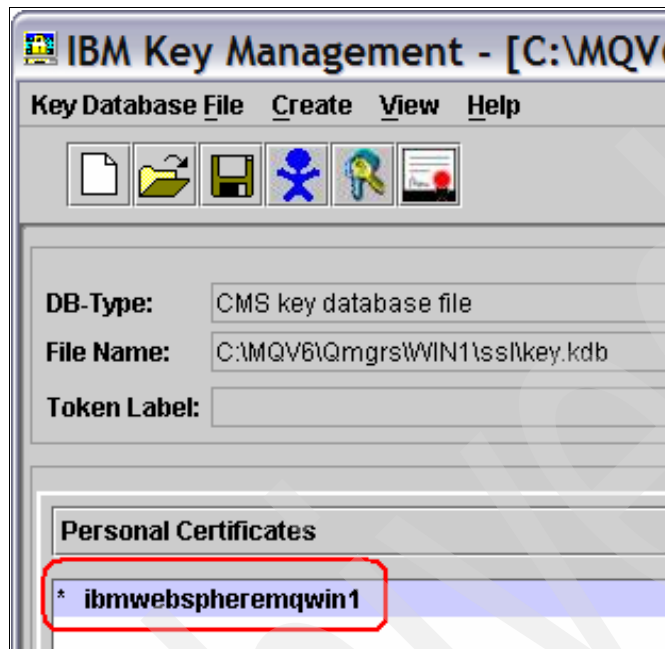


Figure 3-6 Personal Certificates

5. If you switch the view to **Signer Certificates** (top-right corner of main pane), you see the certification authority certificate, as shown in Figure 3-7.

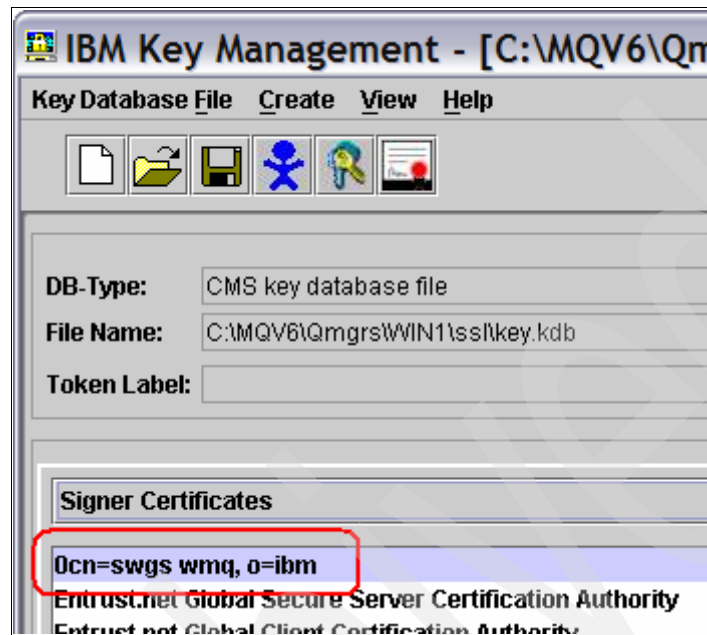


Figure 3-7 Signer (CA) Certificate

6. Close the key repository by selecting **Key Database File** → **Close**.
7. Exit the iKeyman GUI.

3.4.6 Altering the channel attributes

Now both the z/OS and Windows queue managers have their certificates. To use SSL, we need to change the channel attributes:

1. Change the cipherspec attribute to NULL_MD5 on the sender (WI04.WIN1) and receiver (WIN1.WI04) channels on z/OS to NULL_MD5 (see 3.3, “z/OS to z/OS” on page 55 for instructions).
2. Using WebSphere MQ Explorer, change the sender (WIN1.WI04) and receiver (WI04.WIN1) channels to use the NULL_MD5 cipher specification:
 - a. Right-click the sender channel, select **Properties**, and select the **SSL** tab.
 - b. Select **NULL_MD5** from the Standard Settings menu. Click **OK**.
 - c. Right-click the receiver channel, select **Properties**, and select the **SSL** tab.

- d. Select **NULL_MD5** from Standard Settings and click **OK**.
3. Stop and start the channels. The channels should start, meaning they have successfully exchanged their SSL certificates.

This concludes the setup of SSL WebSphere MQ channels between z/OS and Windows.

3.5 z/OS to AIX 5L

This section describes how to set up SSL channels between a z/OS queue manager and an AIX 5L queue manager.

Assumptions

The instructions that follow assume the configuration listed in Table 3-6.

Table 3-6 Queue manager configuration

Platform	z/OS	AIX 5L
Queue manager name	WI05	AIX1
Queue manager user ID	EMIR	emir
CHINIT user ID	EMIR	n/a
Host name	winmvs60	allium
Listener port	20005	20001

Ensure that you have a working sender/receiver channel pair between the two queue managers.

Sample JCL for queue manager WI05

Example 3-43 shows sample JCL for queue manager WI05.

Example 3-43 JCL for WI05

```
//CSQUTIL EXEC PGM=CSQUTIL,PARM=WI05
//STEPLIB DD DISP=SHR,DSN=MQM.V600.SCSQANLE
//          DD DISP=SHR,DSN=MQM.V600.SCSQAUTH
//          DD DISP=SHR,DSN=MQM.V600.SCSQLOAD
//CSQUCMD DD *
DEF QL(AIX1) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WI05.AIX1)
DEF CHL(WI05.AIX1) CHLTYPE(SDR) REPLACE +
    CONNAME('ALLIUM(20001)') XMITQ(AIX1)
```

```

DEF CHL(AIX1.WI05) CHLTYPE(RCVR) REPLACE
DEF QL(WI05.Q1) REPLACE
DEF QR(AIX1.Q1) REPLACE +
    RNAME(AIX1.Q1) RQMNAME(AIX1)
//SYSIN DD *
    COMMAND TGTQMGR(WI05) DDNAME(CSQUCMD)
//SYSPRINT DD SYSOUT=*

```

Sample MQSC commands for queue manager AIX1

Example 3-44 shows sample MQSC commands for queue manager AIX1.

Example 3-44 MQSC for AIX1

```

DEF QL(WI05) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(AIX1.WI05)
DEF CHL(AIX1.WI05) CHLTYPE(SDR) REPLACE +
    CONNAME('WINMVS60(20005)') XMITQ(WI05)
DEF CHL(WI05.AIX1) CHLTYPE(RCVR) REPLACE
DEF QL(AIX1.Q1) REPLACE
DEF QR(WI05.Q1) REPLACE +
    RNAME(WI05.Q1) RQMNAME(WI05)

```

Task summary

Table 3-7 provides a task summary.

Table 3-7 Task summary

Task	Comments
1. Enable SSL on the z/OS queue manager.	Already done when setting up SSL between two z/OS queue managers. See 3.3.1, “Enabling SSL on the queue managers” on page 57.
2. Create a certificate and assign it to the z/OS queue manager.	Already done when setting up SSL between two z/OS queue managers. See: <ul style="list-style-type: none"> ▶ 3.3.2, “Creating the queue manager certificate” on page 63 ▶ 3.3.3, “Connecting the certificate to the key ring” on page 66
3. Use RACF to create a certificate for the AIX 5L queue manager.	See 3.5.1, “Creating a certificate for the AIX 5L queue manager” on page 86.
4. Export the certificate from RACF.	See 3.5.2, “Exporting the certificate” on page 89.

Task	Comments
5. Download (FTP) the certificate to AIX 5L.	See 3.5.3, "Downloading the certificate to AIX 5L" on page 91.
6. Create a key repository for the AIX 5L queue manager.	See 3.5.4, "Creating a key repository for AIX1" on page 91.
7. Import the certificate in AIX 5L.	See 3.5.5, "Importing the certificate for AIX1" on page 95.
8. Alter the channel definitions to start using SSL.	See 3.5.6, "Altering the channel attributes" on page 95.

3.5.1 Creating a certificate for the AIX 5L queue manager

To create a certificate for the AIX 5L queue manager:

1. From the RACF certificates panel, select 1 (Generate a certificate).
2. Fill in the Generate a Digital Certificate panel.

Enter the values shown in Example 3-45. Press PF8.

Example 3-45 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

Personal Certificate
 (user ID) or Site or Authority
 Certificate Type => **emir** => _ => _

Enter the name of the data set which contains the PKCS10 request data:

More: +

Enter the following information about the Signing Authority:
 (Required if a PKCS10 data set name was entered above)

(default) Certificate
 Signing Personal or Site or Authority
 Certificate Type: _ => _ => **x**
 and
 Label(in quotes): '**MQ Root CA**' _____

Enter the Subject's X.509 Distinguished Name:

Common Name(in quotes): (ex: 'John Q. Public')

'AIX1' _____

Title(in quotes): (ex: 'Systems Programmer')

Enter the values shown in Example 3-46. Press PF8.

Example 3-46 Generate a Digital Certificate

RACF - Generate a Digital Certificate
COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR _____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

_____ More: - +

Organizational Unit(in quotes): (ex: 'S390','MVS')

Organization(in quotes): (ex: 'IBM')

'IBM' _____

Locality(in quotes): (ex: 'Poughkeepsie')

State/Province(in quotes): (ex: 'New York')

There is nothing to enter on the following panel, as shown in Example 3-47.
Press PF8.

Example 3-47 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ===>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR _____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Country(in quotes): (ex: 'US')

Enter the decimal size of the private key:

_____ (Default is 1024)

Enter any character to use IBM's crypto service provider:

_ Integrated Cryptographic Support Facility (ICSF).

Enter when the certificate is valid:

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)

Enter the values shown in Example 3-48. Note that the label prefix in UNIX must be all lowercase. Press Enter.

Example 3-48 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ===>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR _____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

NOTBEFORE Date: **2025-12-31** (Default is current local date)

NOTBEFORE Time: HH:MM:SS (Default is 00:00:00)

NOTAFTER Date: YYYY-MM-DD (Default is 1 year after NOTBEFORE)

NOTAFTER Time: HH:MM:SS (Default is 23:59:59)

Enter the label name that is to be assigned to this certificate:

Label(in quotes): 'ibmwebspheremqaix1' _____

Enter the key usage (more than one may be chosen):

- ☒ Handshake - identification during security handshakes
 - ☐ Data encrypt - encrypting data
 - ☐ Docsign - legally binding signatures
 - ☐ Certsign - signs other certificates and CRLs
-

You only receive a response if the command fails.

3.5.2 Exporting the certificate

To export the certificate:

1. From the RACF certificates panel, select option 3 (Write a certificate to a data set).
2. Fill in the Write a Certificate to a Data Set panel.

Enter the values shown in Example 3-49. RACF exports the key to a data set called userid.SSLKEY.AIX1. Press PF8.

Example 3-49 Export a certificate

RACF - Write a Certificate to a Data Set

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	emir _____	=> _	=> _

More: +

Enter the label name of the certificate:
'ibmwebspheremqaix1' _____ (in quotes)

Enter the data set name to which the certificate is to be written:
sslkey.aix1 _____

Enter any character to indicate the Certificate Format Type:

- _ DER encoded X.509
- _ Base64 encoded X.509 (default)
- x** DER encoded PKCS12
- _ Base64 encoded PKCS12
- _ DER encoded PKCS7
- _ Base64 encoded PKCS7

Note that you must enter/reenter a password (in quotation marks). You need the password later when you import the certificate. Press Enter.

Example 3-50 Export a certificate

```

RACF - Write a Certificate to a Data Set
COMMAND ==>

                Personal      Certificate
                (user ID)    or Site    or Authority
Certificate Type => EMIR_____ => _    => _
More:           -

    _ DER encoded X.509
    _ Base64 encoded X.509 (default)
    x DER encoded PKCS12
    _ Base64 encoded PKCS12
    _ DER encoded PKCS7
    _ Base64 encoded PKCS7
Password for PKCS12 format data set (in quotes):
=> 'password' [not visible]          <=

Re-enter password to verify (in quotes):
=> 'password' [not visible]          <=

```

If the command worked, you see (ISPF option 3.4) the data set
userid.SSLKEY.AIX1.

3.5.3 Downloading the certificate to AIX 5L

Use FTP to download the exported certificate to AIX 5L. See Example 3-51. We put it in the queue manager's ssl directory (/var/mqm/qmgrs/AIX1/ssl).

Example 3-51 Download the certificate

```
/var/mqm/qmgrs/AIX1/ssl:>ftp winmvs60
Connected to winmvs60.hursley.ibm.com.
220-FTPD1 IBM FTP CS V1R7 at WINMVS60.HURSLEY.IBM.COM, 17:19:46 on
2006-06-29.
220 Connection will close if idle for more than 5 minutes.
Name (winmvs60:emir): emir
331 Send password please.
Password:
230 EMIR is logged on. Working directory is "EMIR.".
ftp> cd sslkey
250 "EMIR.SSLKEY." is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp> get aix1 AIX1.p12
200 Port request OK.
125 Sending data set EMIR.SSLKEY.AIX1
250 Transfer completed successfully.
2316 bytes received in 0.000192 seconds (1.178e+04 Kbytes/s)
local: AIX1.p12 remote: aix1
ftp> bye
221 Quit command received. Goodbye.
```

Note the binary file transfer and the file extension (p12) of the exported certificate.

3.5.4 Creating a key repository for AIX1

By default, the key repository for the AIX1 queue manager is in /var/mqm/qmgrs/AIX1/ssl.

AIX 5L prerequisites

Note the following AIX 5L prerequisites.

X Window System

To run the iKeyman GUI under AIX 5L, you need Exceed (or a similar X Window System server). Perform the following steps:

1. Start Exceed (or your X Window System server) on your PC.

2. On AIX 5L, add this line to your .profile (remember to rerun your .profile; the easiest way is to exit and log in again):

```
export DISPLAY='<your PC's hostname or IP address>:0.0'
```

For example:

```
export DISPLAY='9.20.84.21:0.0'
```

3. To test, enter **xclock**:

```
/tmp/emir:>xclock
```

The clock shown in Figure 3-8 appears in the top-left corner of your PC screen.

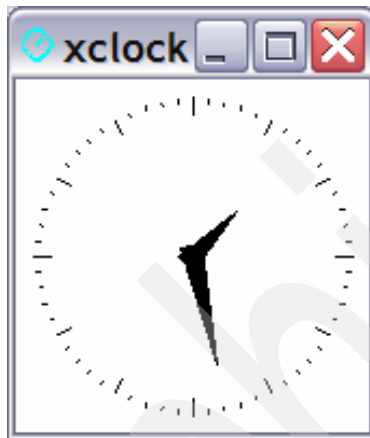


Figure 3-8 *xclock*

If you cannot run X Window System applications, use the command line.

Everything the iKeyman GUI does, you can also do from the command line, using the command **gsk7cmd**. If you cannot run the iKeyman GUI (**gsk7ikm**), download SupportPac™ MO04 (WebSphere MQ SSL Wizard) from the following Web site:

http://www.ibm.com/support/docview.wss?rs=171&uid=swg24010367&loc=en_US&cs=utf-8&lang=en

The SupportPac consists of a program that takes as input your queue manager and channel configuration, and generates all the commands that you need to set up SSL channels between two queue managers.

Java

Set your PATH environment variable so that it runs the JVM™ that comes with WebSphere MQ:

```
PATH=/usr/mqm/ssl/jre/bin:$PATH
export PATH
```

Notes:

- ▶ Your path might be different.
- ▶ Remember to re-run your .profile (. ./profile or exit and log in again).

Check that you can run Java™ (Example 3-52).

Example 3-52 Check Java version

```
/tmp/emir:>java -version
java version "1.4.2"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2)
Classic VM (build 1.4.2, J2RE 1.4.2 IBM AIX build ca142-20050609 (JIT
enabled: jitc))
```

The reason for using the JVM that comes with WebSphere MQ is that you need the correct versions of the files shown in Example 3-53.

Example 3-53 Policy JAR files

```
/usr/mqm/ssl/jre/lib/security/local_policy.jar
/usr/mqm/ssl/jre/lib/security/US_export_policy.jar
```

At the time of writing, not all Java runtimes have these files at the level required by iKeyman.

If you use a different JVM, you might be unable to import the queue manager certificate into the key repository (see 3.5.5, “Importing the certificate for AIX1” on page 95).

Follow this link for an article about how to obtain up-to-date versions of the policy JAR files, in case you opt to use your system's standard JVM:

<http://www.ibm.com/support/docview.wss?uid=swg21201170>

JAVA_HOME

Your .profile must set the JAVA_HOME variable, for example:

```
export JAVA_HOME=/usr/mqm/ssl/jre
```

Notes:

- ▶ We are using the Java runtime that comes with WebSphere MQ.
- ▶ Remember to rerun your .profile before starting iKeyman.

AIX 5L instructions

To create a key repository:

1. Start the iKeyman GUI. Enter **gsk7ikm**. Note that it takes a few seconds for the GUI to appear.

When you have not yet started the X Window System server on your PC, the error shown in Example 3-54 occurs.

Example 3-54 X Window System error

```
/tmp/emir:>gsk7ikm
```

```
The java class could not be loaded. java.lang.InternalError: Can't
connect to X11 window server using '9.20.84.21:0.0' as the value of the
DISPLAY variable.
```

2. Select **Key Database File** → **New**. Enter the following values:

- Key database type: **CMS**
- File Name: **key.kdb**
- Location: **/var/mqm/qmgrs/AIX1/ssl**
(This is the default used by the queue manager.)

Click **OK**.

3. This opens a password window. Enter a password (twice). This is to protect the repository. (Remember this password, because you need it each time you work with this key repository.)

Select **Stash the password to a file**. Click **OK**.

4. After a few seconds, you see the following message:

```
"The password has been encrypted and saved in file:
/var/mqm/qmgrs/AIX1/ssl/key.sth."
```

Click **OK**.

This concludes the creation of the key repository for AIX1. Do not close the iKeyman GUI.

3.5.5 Importing the certificate for AIX1

To import the certificate for AIX1:

1. The GUI should display Signer Certificates (menu in top right corner of the main pane). Change the drop-down to **Personal Certificates**. The Import button to the right becomes active. Click **Import**.

2. This opens the Import Keys window. Use the following values:

- Key file type: **PKCS12**
- File Name: AIX1.p12
- Location: The directory where you placed the exported certificate
In our case: /var/mqm/qmgrs/AIX1/ssl

Click **OK**.

3. This opens the password window. Enter the password you gave when exporting the certificate from z/OS, and then click **OK**.

4. You have the option to change the labels. The queue manager (personal) certificate already has the correct label (ibmwebsphermqai x1).

The certification authority certificate (which signs the personal certificate) is the one we created as MQ Root CA on z/OS. It appears with the label 0cn=swgs_wmq,o=ibm. It was exported (and will be imported) together with the personal certificate.

There is no need to change any of these labels, so click **OK**.

5. You see the queue manager's certificate (label ibmwebsphermqai x1) in the Personal Certificates view.

If you switch the view to **Signer Certificates**, you see the certification authority certificate (label 0cn=swgs_wmq,o=ibm).

Close the key repository by selecting **Key Database File** → **Close**.

6. Exit the iKeyman GUI.

3.5.6 Altering the channel attributes

To alter the channel attributes:

1. Change the cipherspec attribute to NULL_MD5 on the sender (WI05.AIX1) and receiver (AIX1.WI05) channels on z/OS (see 3.3, “z/OS to z/OS” on page 55 for instructions).

2. On AIX 5L, change the sender (AIX1.WI05) and receiver (WI05.AIX1) channels to use the NULL_MD5 cipher specification. Use WebSphere MQ Explorer, or the MQSC commands shown in Example 3-55.

Example 3-55 Alter the channel

```
alter chl('AIX1.WI05') chltype(sdr) sslciph('NULL_MD5')  
alter chl('WI05.AIX1') chltype(rcvr) sslciph('NULL_MD5')
```

3. Stop and start the channels. The channels should start, meaning they have successfully exchanged their SSL certificates.

This concludes the setup of SSL WebSphere MQ channels between z/OS and AIX 5L.

3.6 AIX 5L to Windows

The section describes how to set up SSL channels between an AIX 5L queue manager and a Windows queue manager.

Assumptions

The instructions that follow assume the configuration listed in Table 3-8.

Table 3-8 Queue manager configuration

Platform	AIX 5L	Windows
Queue manager name	AIX1	WIN2
Queue manager user ID	emir	emir
CHINIT user ID	n/a	n/a
Host name	allium	garzae
Listener port	20001	20002

Ensure that you have a working sender/receiver channel pair between the two queue managers.

Example MQSC to set up the channels

The section provides example MQSC to set up the channels.

Example 3-56 provides an example for queue manager AIX1.

Example 3-56 MQSC for AIX1

```
DEF QL(WIN2) REPLACE USAGE(XMITQ) TRIGGER +  
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(AIX1.WIN2)  
DEF CHL(AIX1.WIN2) CHLTYPE(SDR) REPLACE +  
    CONNAME('garzae(20002)') XMITQ(WIN2)  
DEF CHL(WIN2.AIX1) CHLTYPE(RCVR) REPLACE  
DEF QL(AIX1.Q1) REPLACE  
DEF QR(WIN2.Q1) REPLACE +  
    RNAME(WIN2.Q1) RQMNAME(WIN2)
```

Example 3-57 provides an example for queue manager WIN2.

Example 3-57 MQSC for WIN2

```
DEF QL(AIX1) REPLACE USAGE(XMITQ) TRIGGER +  
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WIN2.AIX1)  
DEF CHL(WIN2.AIX1) CHLTYPE(SDR) REPLACE +  
    CONNAME('allium(20001)') XMITQ(AIX1)  
DEF CHL(AIX1.WIN2) CHLTYPE(RCVR) REPLACE  
DEF QL(WIN2.Q1) REPLACE  
DEF QR(AIX1.Q1) REPLACE +  
    RNAME(AIX1.Q1) RQMNAME(AIX1)
```

Task summary

Table 3-9 shows a summary of the necessary tasks.

Table 3-9 Task summary

Task	Comments
1. Use RACF to create a certificate for the AIX 5L queue manager.	Already done when setting up SSL between z/OS and AIX 5L. See 3.5.1, “Creating a certificate for the AIX 5L queue manager” on page 86.
2. Export the certificate from RACF.	Already done when setting up SSL between z/OS and AIX 5L. See 3.5.2, “Exporting the certificate” on page 89.
3. Download (FTP) the certificate to AIX 5L.	Already done. See 3.5.3, “Downloading the certificate to AIX 5L” on page 91.
4. Create a key repository for queue manager AIX1.	Already done. See 3.5.4, “Creating a key repository for AIX1” on page 91.

Task	Comments
5. Import the certificate for queue manager AIX1.	Already done. See 3.5.5, "Importing the certificate for AIX1" on page 95.
6. Use RACF to create and export a certificate for the Windows queue manager.	See 3.6.1, "Creating a certificate for the Windows queue manager" on page 98 and 3.6.2, "Exporting the certificate" on page 101.
7. Download (FTP) the certificate to Windows.	See 3.6.3, "Downloading the certificate to Windows" on page 102.
8. Create a key repository for the Windows queue manager.	See 3.6.4, "Creating a key repository for WIN2" on page 103.
9. Import the certificate into Windows.	See 3.6.5, "Importing the certificate" on page 104.
10. Alter the channel definitions to start using SSL.	See 3.6.6, "Altering the channel attributes" on page 105.

3.6.1 Creating a certificate for the Windows queue manager

To create a certificate for the Windows queue manager:

1. From the RACF certificates panel, select 1 (Generate a certificate).
2. Fill in the Generate a Digital Certificate panel.

Enter the values shown in Example 3-58. Press PF8.

Example 3-58 Generate a Digital Certificate

```

RACF - Generate a Digital Certificate
COMMAND ===>

                Personal          Certificate
                (user ID) or Site  or Authority
Certificate Type => emir_____ => _    => _

Enter the name of the data set which contains the PKCS10 request data:
_____

More:      +

Enter the following information about the Signing Authority:
(Required if a PKCS10 data set name was entered above)

                (default)          Certificate
Signing         Personal  or Site  or Authority
Certificate Type:  _        => _    => x
and
```


Label(in quotes): 'MQ Root CA'_____

Enter the Subject's X.509 Distinguished Name:

Common Name(in quotes): (ex: 'John Q. Public')

'WIN2'_____

Title(in quotes): (ex: 'Systems Programmer')

Enter the values shown in Example 3-59. Press PF8.

Example 3-59 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR_____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Organizational Unit(in quotes): (ex: 'S390','MVS')

Organization(in quotes): (ex: 'IBM')

'IBM'_____

Locality(in quotes): (ex: 'Poughkeepsie')

State/Province(in quotes): (ex: 'New York')

There is nothing to enter on the next panel, as shown in Example 3-60. Press PF8.

Example 3-60 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ===>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR_____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

Country(in quotes): (ex: 'US')

Enter the decimal size of the private key:

_____ (Default is 1024)

Enter any character to use IBM's crypto service provider:

_ Integrated Cryptographic Support Facility (ICSF).

Enter when the certificate is valid:

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)

Enter the values shown in Example 3-61. Press Enter.

Example 3-61 Generate a Digital Certificate

RACF - Generate a Digital Certificate

COMMAND ===>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	EMIR_____	=> _	=> _

Enter the name of the data set which contains the PKCS10 request data:

More: - +

NOTBEFORE Date: YYYY-MM-DD (Default is current local date)
NOTBEFORE Time: HH:MM:SS (Default is 00:00:00)

NOTAFTER Date: 2025-12-31 (Default is 1 year after NOTBEFORE)
NOTAFTER Time: HH:MM:SS (Default is 23:59:59)

Enter the label name that is to be assigned to this certificate:

Label(in quotes): 'ibmwebspheremqwin2' _____

Enter the key usage (more than one may be chosen):

☒ Handshake - identification during security handshakes
☐ Data encrypt - encrypting data
☐ Docsign - legally binding signatures
☐ Certsign - signs other certificates and CRLs

You only receive a response if the command fails.

3.6.2 Exporting the certificate

To export the certificate:

1. From the RACF certificates panel, select option 3 (Write a certificate to a data set).
2. Fill in the Write a Certificate to a Data Set panel.

Enter the values shown in Example 3-62. RACF exports the key to a data set called userid.SSLKEY.WIN2. Press PF8.

Example 3-62 Export a certificate

RACF - Write a Certificate to a Data Set

COMMAND ==>

	Personal (user ID)	or Site	Certificate or Authority
Certificate Type =>	emir_____	=> _	=> _

More: +

Enter the label name of the certificate:
'ibmwebspheremqwin2'_____ (in quotes)

Enter the data set name to which the certificate is to be written:
sslkey.win2_____

Enter any character to indicate the Certificate Format Type:

- _ DER encoded X.509
 - _ Base64 encoded X.509 (default)
 - x** DER encoded PKCS12
 - _ Base64 encoded PKCS12
 - _ DER encoded PKCS7
 - _ Base64 encoded PKCS7
-

Note that you must enter/reenter a password (in quotation marks). You need the password later when you import the certificate. Press Enter.

Example 3-63 Export a certificate

```
RACF - Write a Certificate to a Data Set
COMMAND ==>

Certificate Type => EMIR_____ => _ => _
                        Personal      Certificate
                        (user ID)    or Site    or Authority
More: -

_ DER encoded X.509
_ Base64 encoded X.509 (default)
x DER encoded PKCS12
_ Base64 encoded PKCS12
_ DER encoded PKCS7
_ Base64 encoded PKCS7
Password for PKCS12 format data set (in quotes):
=> 'password' [not visible] <=

Re-enter password to verify (in quotes):
=> 'password' [not visible] <=
```

If the command worked, you see (ISPF option 3.4) the data set userid.SSLKEY.WIN2.

3.6.3 Downloading the certificate to Windows

Use FTP to download the exported certificate to Windows (Example 3-64 on page 103). We place the exported certificate in the queue manager's ssl directory (<MQDir>\Qmgrs\WIN2\ssl).

```
C:\MQV6\Qmgrs\WIN1\ssl>ftp winmvs60
Connected to winmvs60.hursley.ibm.com.
220-FTPD1 IBM FTP CS V1R7 at WINMVS60.HURSLEY.IBM.COM, 14:14:50 on
2006-06-29.
220 Connection will close if idle for more than 5 minutes.
User (winmvs60.hursley.ibm.com:(none)): emir
331 Send password please.
Password:
230 EMIR is logged on. Working directory is "EMIR.".
ftp> cd sslkey
250 "EMIR.SSLKEY." is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp> get win1 WIN2.p12
200 Port request OK.
125 Sending data set EMIR.SSLKEY.WIN1
250 Transfer completed successfully.
ftp: 2316 bytes received in 0.02Seconds 115.80Kbytes/sec.
ftp> bye
221 Quit command received. Goodbye.
```

Note the binary file transfer and the file extension (p12) of the exported certificate.

3.6.4 Creating a key repository for WIN2

The following instructions show how to create a key repository for queue manager WIN2.

This is already described in detail in Chapter 1, “Connecting two Windows queue managers using SSL” on page 1, so this section omits the screen captures.

Perform the following steps:

1. Open a Windows command prompt and enter **strmqikm**. This starts the IBM Key Management (iKeyman) GUI.
2. 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\WIN2\ssl**
In this example: **C:\MQV6\Qmgrs\WIN2\ssl**

Click **OK**.

3. Enter a password (and remember it, because you need it later) and select **Stash the password to a file?** Click **OK**.
4. You see a message saying that the password was encrypted and saved. Click **OK**.

You have created a key repository for queue manager WIN2.

After creating the key repository, the GUI shows the installed certification authority certificates provided with iKeyman. Switch to viewing **Personal Certificates**.

Do *not* close the iKeyman GUI.

3.6.5 Importing the certificate

To import the certificate:

1. The GUI should display the Signer Certificates. Change to the **Personal Certificates** view.
The Import button to the right becomes active. Click **Import**.

2. This opens the Import Keys window. Use the following values:

- Key file type: **PKCS12**.
- File Name: WIN2.p12
- Location: The directory where you placed the exported certificate
In our case: `<MQdir>\Qmgrs\WIN2\ssl`

Click **OK**.

3. This opens the password window. Enter the password you gave when exporting the certificate from z/OS, and then click **OK**.
4. You have the option to change the labels. The queue manager (personal) certificate already has the correct label (ibmwebspheremqwin2).

The certification authority certificate (which signs the personal certificate) is the one we created as MQ Root CA on z/OS. It appears with the label `Ocn=swgs_wmq,o=ibm`. It was exported (and will be imported) together with the personal certificate.

There is no need to change any of these labels, so click **OK**.

5. You see the queue manager's certificate (label ibmwebspheremqwin2) in the Personal Certificates view. If you switch the view to **Signer Certificates**, you see the certification authority certificate (label `Ocn=swgs_wmq,o=ibm`).

Close the key repository by selecting **Key Database File** → **Close**.

6. Exit the iKeyman GUI.

3.6.6 Altering the channel attributes

To alter the channel attributes:

1. On AIX 5L, set the cipherspec attribute to NULL_MD5 on the sender (AIX1.WIN2) and receiver (WIN2.AIX1) channels.
2. On Windows, change the sender (WIN2.AIX1) and receiver (AIX1.WIN2) channels to use the NULL_MD5 cipher specification.
3. Stop and start the channels. The channels should start, meaning they have successfully exchanged their SSL certificates.

This concludes the setup of SSL WebSphere MQ channels between AIX 5L and Windows.

3.7 Windows to Windows

This section describes how to set up SSL channels between two Windows queue managers.

Assumptions

The instructions that follow assume the configuration listed in Table 3-10.

Table 3-10 Queue manager configuration

Platform	Windows	Windows
Queue manager name	WIN1	WIN2
Queue manager user ID	emir	emir
CHINIT user ID	N/A	N/A
Host name	garzae	garzae
Listener port	20001	20002

Ensure that you have a working sender/receiver channel pair between the two queue managers.

Example MQSC to set up the channels

This section provides example MQSC to set up the channels.

Example 3-65 provides an example for queue manager WIN1.

Example 3-65 MQSC for WIN1

```
DEF QL(WIN2) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WIN1.WIN2)
DEF CHL(WIN1.WIN2) CHLTYPE(SDR) REPLACE +
    CONNAME('garzae(20002)') XMITQ(WIN2)
DEF CHL(WIN2.WIN1) CHLTYPE(RCVR) REPLACE
DEF QL(WIN1.Q1) REPLACE
DEF QR(WIN2.Q1) REPLACE +
    RNAME(WIN2.Q1) RQMNAME(WIN2)
```

Example 3-66 provides an example for queue manager WIN2.

Example 3-66 MQSC for WIN2

```
DEF QL(WIN1) REPLACE USAGE(XMITQ) TRIGGER +
    INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(WIN2.WIN1)
DEF CHL(WIN2.WIN1) CHLTYPE(SDR) REPLACE +
    CONNAME('garzae(20001)') XMITQ(WIN1)
DEF CHL(WIN1.WIN2) CHLTYPE(RCVR) REPLACE
DEF QL(WIN2.Q1) REPLACE
DEF QR(WIN1.Q1) REPLACE +
    RNAME(WIN1.Q1) RQMNAME(WIN1)
```

Task summary

To set up SSL channels between two Windows queue managers, we perform the tasks listed in Table 3-11.

Table 3-11 Task summary

Task	Comments
1. Use RACF to create and export a certificate for queue manager WIN1.	Already done when setting up SSL between z/OS and Windows. See: <ul style="list-style-type: none">▶ 3.4.1, “Creating a certificate for WIN1” on page 74▶ 3.4.2, “Exporting the certificate from RACF” on page 77
2. Download (FTP) the certificate to Windows.	Already done when setting up SSL between z/OS and Windows. See 3.4.3, “Downloading the certificate to Windows” on page 78.

Task	Comments
3. Create a key repository for queue manager WIN1 and import the certificate.	Already done when setting up SSL between z/OS and Windows. See: <ul style="list-style-type: none"> ▶ 3.4.4, “Creating a key repository for WIN1” on page 79 ▶ 3.4.5, “Importing the certificate” on page 79
4. Use RACF to create and export a certificate for queue manager WIN2.	Already done when setting up SSL between AIX 5L and Windows. See: <ul style="list-style-type: none"> ▶ 3.6.1, “Creating a certificate for the Windows queue manager” on page 98 ▶ 3.6.2, “Exporting the certificate” on page 101
5. Download (FTP) the certificate to Windows.	Already done when setting up SSL between AIX 5L and Windows. See 3.6.3, “Downloading the certificate to Windows” on page 102.
6. Create a key repository for queue manager WIN2 and import the certificate.	Already done when setting up SSL between AIX 5L and Windows. See: <ul style="list-style-type: none"> ▶ 3.6.4, “Creating a key repository for WIN2” on page 103 ▶ 3.6.5, “Importing the certificate” on page 104
7. Alter the channels to use SSL.	See the following section.

3.7.1 Altering the channel attributes

To alter the channel attributes:

1. Set the cipherspec attribute to NULL_MD5 on the sender (WIN1.WIN2) and receiver (WIN2.WIN1) channels on queue manager WIN1.
2. Change the sender (WIN2.WIN1) and receiver (WIN1.WIN2) channels to use the NULL_MD5 cipher specification on queue manager WIN2.
3. Stop and start the channels. The channels should start, meaning they have successfully exchanged their SSL certificates.

This concludes the setup of SSL WebSphere MQ channels between two Windows queue managers.

Connecting the WebSphere Message Broker V6 Toolkit using SSL

This section presents a step-by-step guide to configuring WebSphere MQ Version 6 and WebSphere Message Broker Toolkit Version 6 (the Toolkit) for communication using WebSphere MQ SSL.

4.1 Process overview

We assume that you already have a WebSphere Message Broker V6 Configuration Manager and that your Toolkit can communicate with the Configuration Manager.

We follow this process to establish an SSL connection between the Toolkit and the Configuration Manager:

1. Create a self-signed certificate for the Configuration Manager queue manager.
2. Install the certification authority part of the queue manager's certificate in the Toolkit's key repository.
3. Set up the server connection channel for one-way (server only) SSL authentication.
4. Test the one-way SSL connection.
5. Create a self-signed certificate for the Toolkit client.
6. Install the certification authority part of the Toolkit's certificate in the queue manager's key repository.
7. Set up the channel for two-way SSL authentication and test.

4.2 One-way (server) SSL authentication

The following instructions assume that you have:

- ▶ A queue manager called MBV6, listening on port 2416. If your queue manager has a different name, change the instructions accordingly.
- ▶ Both the Configuration Manager and Toolkit run on Microsoft Windows. It does not affect the process if they run on the same, or on separate, Windows machines.

4.2.1 Creating a self-signed certificate for the queue manager

This task consists of the following steps, executed on the Configuration Manager queue manager machine:

1. Create a key repository for the queue manager.
2. Create a self-signed certificate for the queue manager.
3. Extract the certification authority part of the certificate (to pass it to the Toolkit). See 4.2.2, "Extracting the CA certificate" on page 114.

To create a key repository for the queue manager:

1. Open a command prompt and enter **strmqikm**. This starts the IBM Key Management (iKeyman) GUI.
2. Create a key repository for the queue manager. Select **Key Database File** → **New**.
3. Create a repository as shown in Figure 4-1:
 - Key database type: **CMS**
 - File Name: **key.kdb**
 - Location: **C:\MQV6\Qmgrs\MBV6\ssl**

Click **OK**.

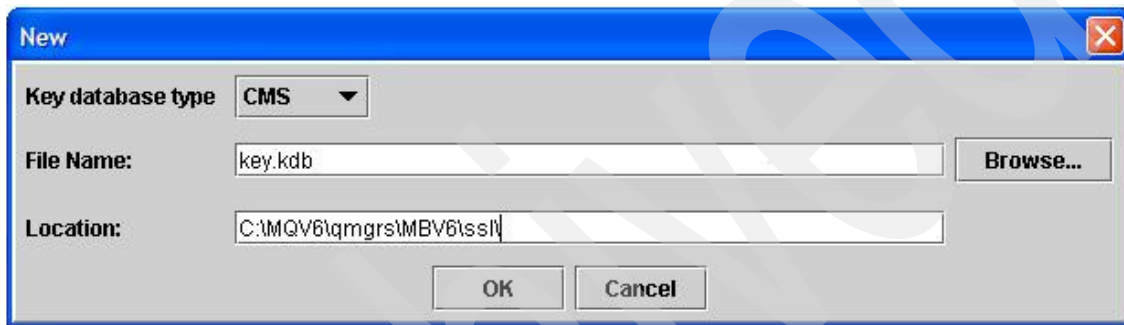


Figure 4-1 Create key repository

4. At the prompt, enter a password (and remember it, because you need it later). Select **Stash the password to a file?**, as shown in Figure 4-2. Click **OK**.

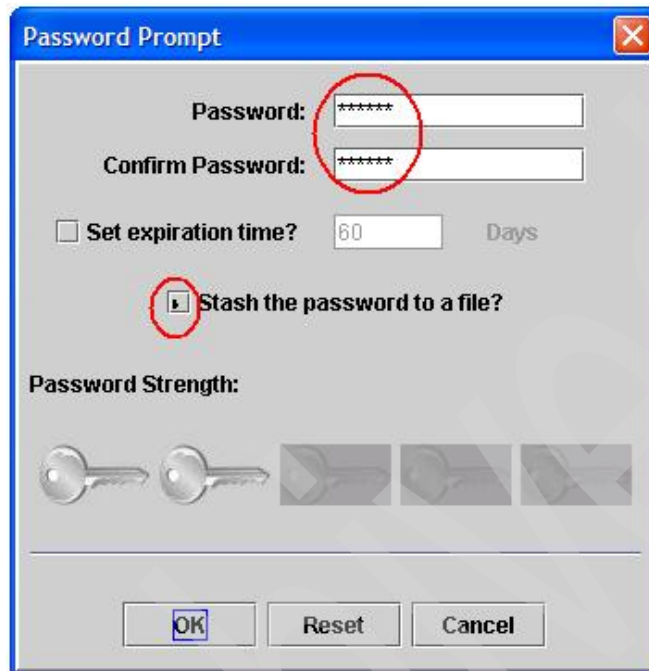


Figure 4-2 Key repository password

5. The message shown in Figure 4-3 opens. Click **OK**.

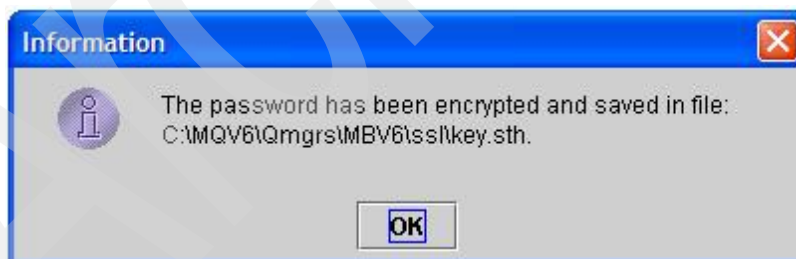


Figure 4-3 Password confirmation

Now that we have created a key repository, we can create a self-signed certificate for the queue manager.

To create a self-signed certificate for the queue manager:

1. After creating the key repository, the GUI shows the installed certification authority certificates provided with iKeyman. Switch to viewing **Personal Certificates**, as shown in Figure 4-4.

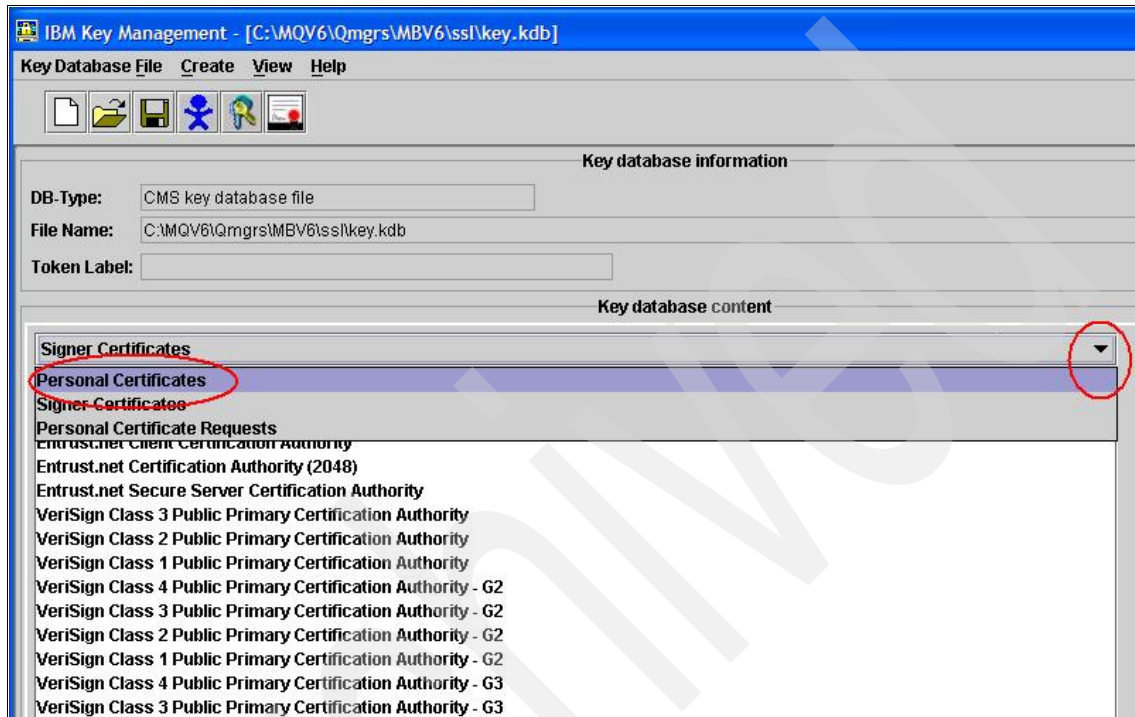


Figure 4-4 Switch to Personal Certificates view

2. Click **New Self-signed** (bottom-right corner).
3. Fill in the certificate attributes, as shown in Figure 4-5 on page 114:
 - Key Label (ibmwebspheremq followed by the queue manager name):
ibmwebspheremqmbv6
 - Common Name: mbv6
(You can have a different naming convention for the common name, feel free to enter any other value.)
 - Organization: ibm (or your company's name).

Leave all other fields unchanged. Click **OK**.

Create New Self-Signed Certificate

Please provide the following:

Key Label	ibmwebspheremqmbv6	
Version	X509 V3 ▼	
Key Size	1024 ▼	
Common Name	mbv6	
Organization	ibm	
Organization Unit (optional)		
Locality (optional)		
State/Province (optional)		
Zipcode (optional)		
Country or region	GB ▼	
Validity Period	365	Days

OK Reset Cancel

Figure 4-5 Create self-signed certificate

You see the certificate listed, with an asterisk to the left of the name (the asterisk means that this is the default certificate for the key repository).

Now the queue manager has a certificate. The queue manager presents this certificate to the Toolkit when the Toolkit connects (as an WebSphere MQ client). To validate the queue manager's certificate, the Toolkit needs the certification authority (CA) certificate.

4.2.2 Extracting the CA certificate

To extract the CA certificate:

1. Click **Extract Certificate** (bottom-right corner).
2. Enter the following values:
 - File Name: mbv6.arm

– Location: C:\MQV6\Qmgrs\MBV6\ssl\

Click **OK**.

This creates a file called mbv6.arm in C:\MQV6\Qmgrs\MBV6\ssl. If you open the file with Notepad, you see something similar to Example 4-1.

Example 4-1 Sample mbv6.arm file

```
-----BEGIN CERTIFICATE-----
MIIByzCCATSgAwIBAgIEQ1kcEDANBgkqhkiG9w0BAQQFADAQMswCQYDVQQGE...
ChMDaWJtMQ0wCwYDVQQDEwRtYnY2MB4XDTA1MTAyMDE2NDkyMFoXDTA2MTAyMT...
MAkGA1UEBhMCR0IxDDAKBgNVBAoTA21ibTENMA5GA1UEAxMEbWJ2NjCBnzANBg...
AA0BjQAwwYkCgYEAOKzAGeurDcg6J7kTCZcm5M3xtkaiNZO6Kq2KLPqaQH8cKC...
E29GBAQ4rW9FmVu6iUJUzi4Z8p7oaXw5Y6Y6JGdJNBsNaEwdKE3hIHNnoygpJ...
k1kbgTUDmVSG8fKsutWftbqJmKDXPDaBjRkCAwEAATANBgkqhkiG9w0BAQQFAA...
oGkiAT0I9TtkP31qZV6ZQmoIZ/UCm9FT7V8gMuux7C55V/Cgh1zsBmf6vVBjrn...
in5Y/Zq7ySx4Y1H2Fmv133cvyRXroujE9bInUSwzmieCg71tys4vEnN11Dm/md...
KKVCZ4oHYg==
-----END CERTIFICATE-----
```

3. You must now transfer the CA certificate file to the Toolkit's key repository. Before that, close the repository you are working with by selecting **Key Database File** → **Close** (Figure 4-6).

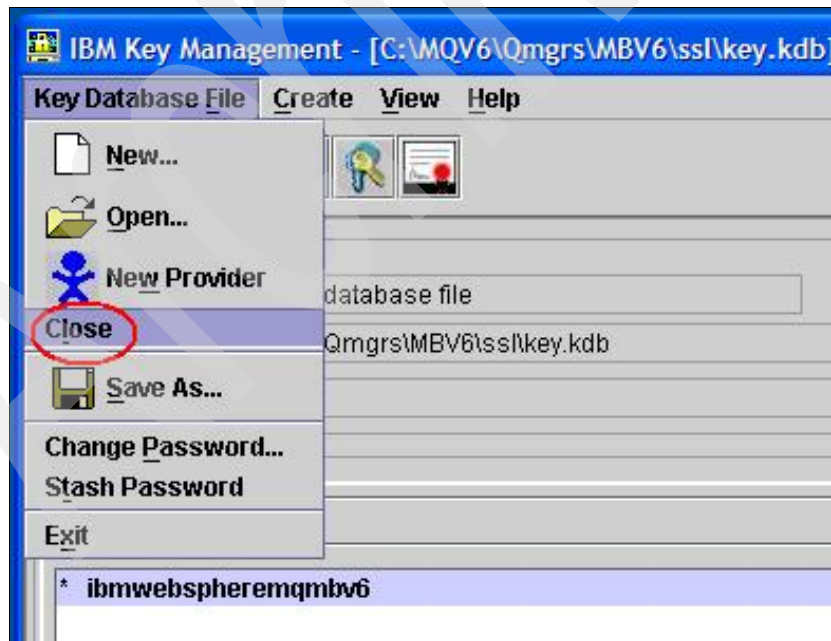


Figure 4-6 Close key repository

4. Close the iKeyman GUI.

4.2.3 Installing the CA part in the Toolkit's key repository

This task consists of the following steps, executed *on the Toolkit machine* (which can be the same as the Configuration Manager's machine):

1. Copy or transfer the CA file (mbv6.arm) to the Toolkit.
2. Create a key repository for the Toolkit.
3. Add the CA file to the Toolkit's key repository.

Transferring the CA file to the Toolkit

Using Windows Explorer, or a command prompt, create a directory called C:\MQCLIENT\Toolkit (Example 4-2).

Example 4-2 Create the Toolkit directory

```
C:\>md MQCLIENT
C:\>cd MQCLIENT
C:\MQCLIENT>md Toolkit
```

We place the CA exported file and the Toolkit's key repository in the C:\MQCLIENT\Toolkit directory.

Copy (or FTP ASCII) the exported CA certificate (mbv6.arm) to the C:\MQCLIENT\Toolkit directory.

Creating the Toolkit key repository

To create the Toolkit key repository:

1. Open a command prompt and enter **strmqikm**.
2. Create a key repository for the Toolkit. Select **Key Database File** → **New**.

3. Create a repository as shown in Figure 4-7:

- Key database type: **JKS**

The Key database type is important. Make sure that you create a JKS key repository. This is because the Toolkit is a WebSphere MQ Java client, which uses JKS repositories by default.

- File Name: key.jks
- Location: C:\MQCLIENT\Toolkit

Click **OK**.

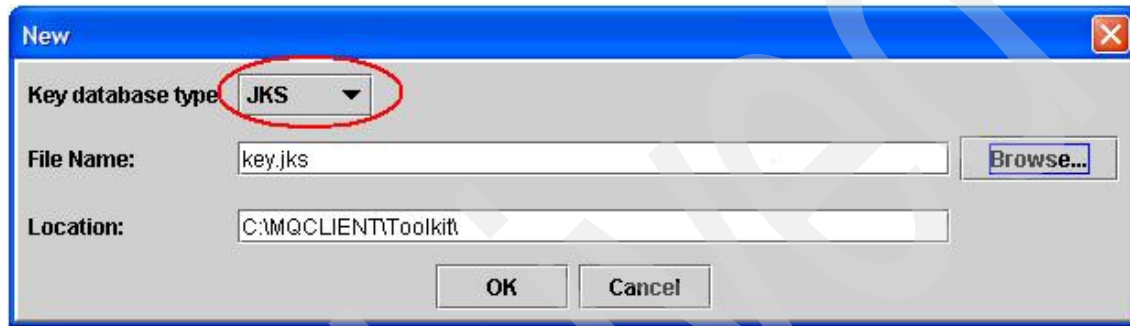


Figure 4-7 New JKS key repository

4. At the password prompt, enter `changeit` as the password for this repository, twice.

Notes:

- `changeit` is the default password for a JKS repository. Do not do this in a production system, but for this exercise, it is easier to just let the Toolkit take the defaults.
- Remember the password. You need it to access the key repository.
- If you use a password other than `changeit`, you must pass it as a Java virtual machine parameter when you start the Toolkit:

```
<Toolkit>\wmbt.exe -vmargs -Djavax.net.ssl.keyStorePassword=mypwd  
-vmargs must be the last parameter passed to wmbt.exe.
```

For example, this is the modified shortcut we used when setting a different password (shown as two lines for readability; everything goes in one line in the Target attribute of the shortcut):

```
C:\MBT\6.0\wmbt.exe -data "C:\Active\SSL\workspace"  
-vmargs -Djavax.net.ssl.keyStorePassword=passw0rd
```

Figure 4-8 shows a screen capture of the shortcut.

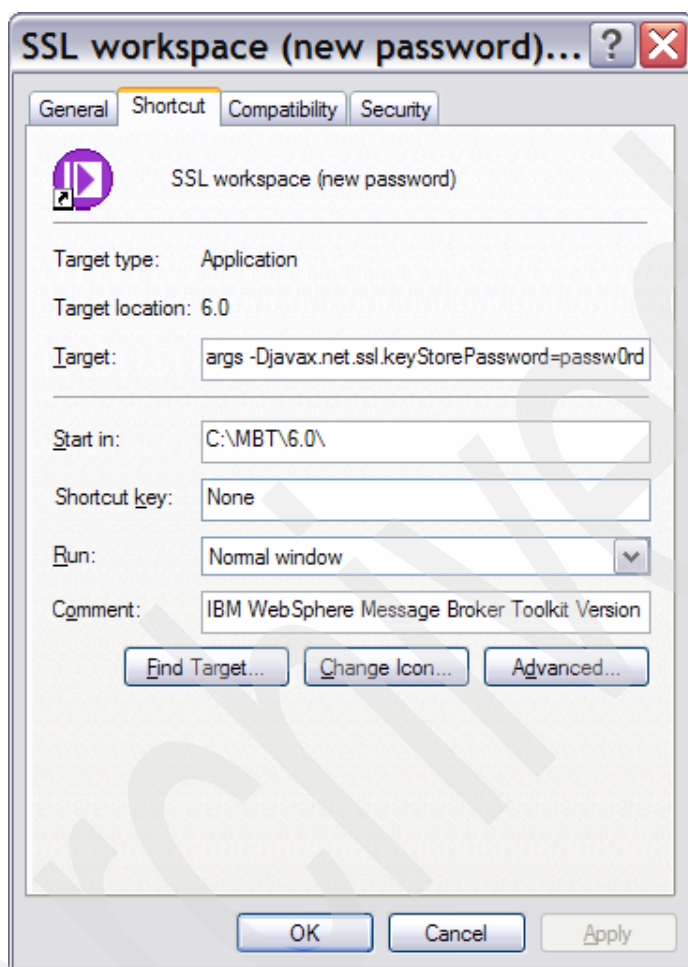


Figure 4-8 Shortcut to WebSphere Message Broker Toolkit

Adding the CA file to the Toolkit's key repository

To add the CA file to the Toolkit's key repository:

1. The iKeyman GUI shows a list of CA certificates provided by default. Click **Add** (to the right of the main pane).

2. In the next window (Figure 4-9), enter:
- Certificate file name: mbv6.arm
 - Location: C:\MQCLIENT\Toolkit\
- Click **OK**.

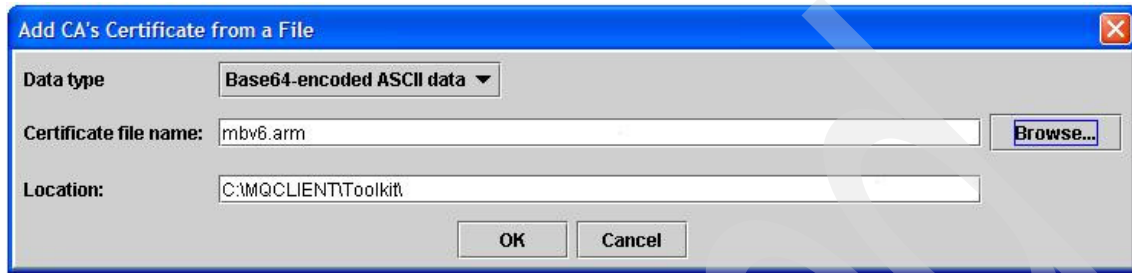


Figure 4-9 Add CA's Certificate to key repository

3. When prompted to enter a label for the certificate, enter mbv6. (You can enter any name you want; just make sure that it lets you identify the CA certificate as belonging to the queue manager.) Click **OK**.

The certificate now appears in the Signer Certificates repository, as shown in Figure 4-10.

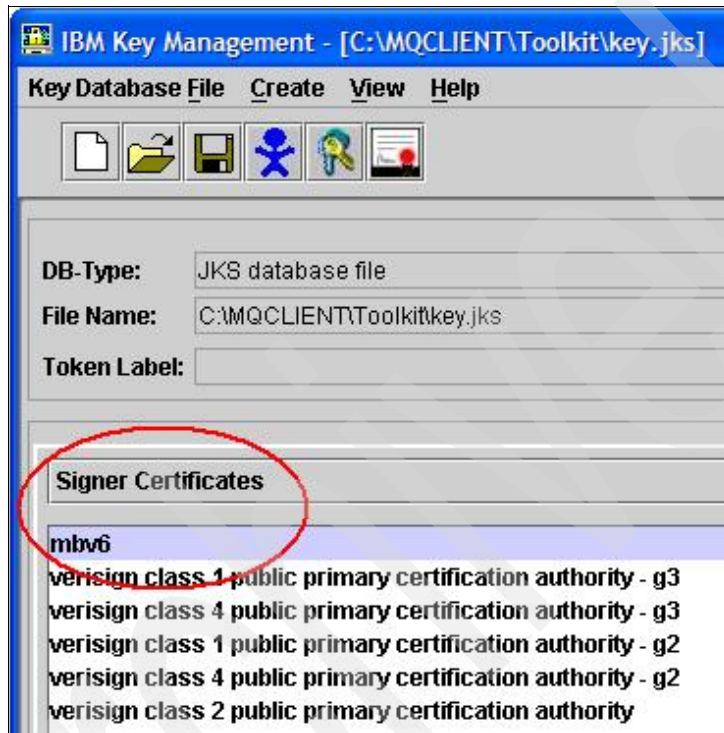


Figure 4-10 Signer Certificates view

4. Close the file (**Key Database File** → **Close**).
5. Close the iKeyman GUI.

At this point, the queue manager has a certificate and the Toolkit has the corresponding CA certificate. We are ready to set up server SSL (one-way) authentication.

4.2.4 Testing the one-way SSL connection

This task consists of the following steps:

1. Verify that the Toolkit connects to the Configuration Manager without SSL.

2. On the Configuration Manager queue manager machine: Set up the server connection channel used by the Configuration Manager.
3. On the Toolkit machine (if different): Set up the Configuration Manager domain and connect.

Before making any changes, check that the connection works without SSL:

1. Start WebSphere MQ Explorer (**Start → All Programs → IBM WebSphere MQ → WebSphere MQ Explorer**).
2. If not already started, start the queue manager MBV6.
3. If not already started, start the Configuration Manager.
4. Start the Toolkit.
5. Connect the Toolkit to the Configuration Manager. Verify that it connects without problems.
6. Disconnect the Toolkit from the Configuration Manager. (Do not close the WebSphere MQ Explorer and Toolkit windows.)

Now, we make the necessary changes to the queue manager. Switch to WebSphere MQ Explorer and perform the following steps:

1. Open the queue manager's Channels view (**Advanced** → **Channels**).

Make sure that you are not filtering out **SYSTEM.*** objects. Click **Show System Objects** (Figure 4-11) if you do not see this.

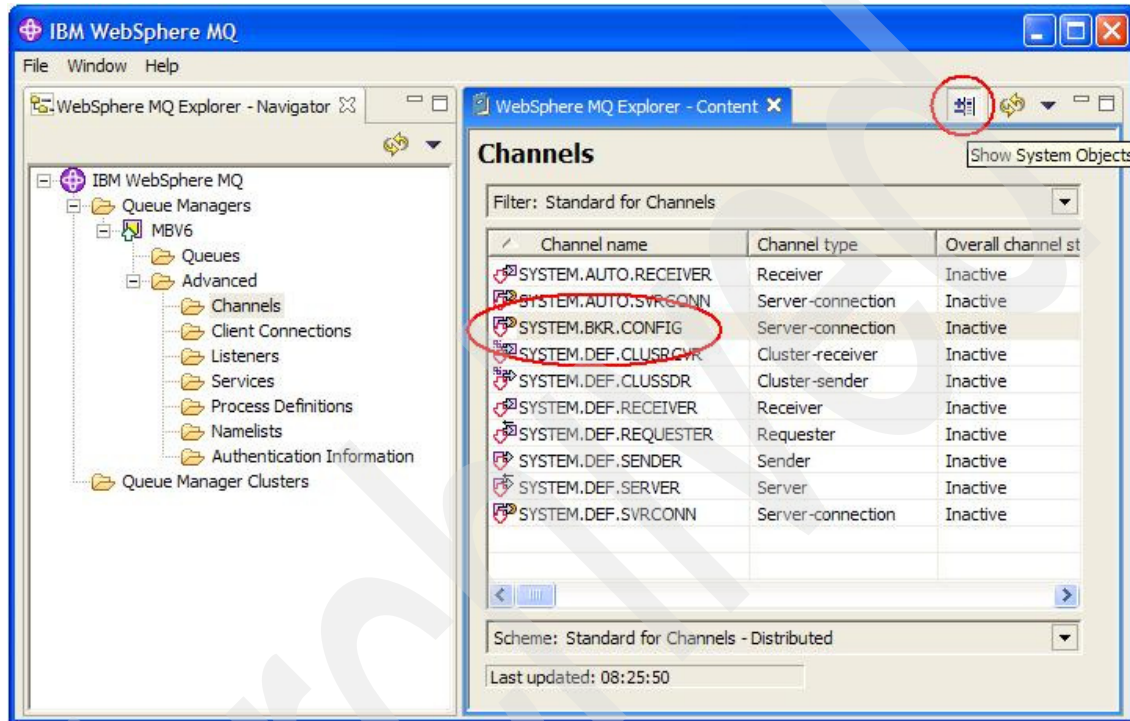


Figure 4-11 Queue Manager Channels

2. Right-click the **SYSTEM.BKR.CONFIG** channel. Select **Properties**.

3. Select the **SSL** tab.

Set the SSL CipherSpec attribute to **NULL_MD5**.

Set Authentication of parties initiating connections to **Optional**, as shown in Figure 4-12.

(Optional makes it one-way: The queue manager presents a certificate, but the Toolkit does not have to.)

Click **OK**.

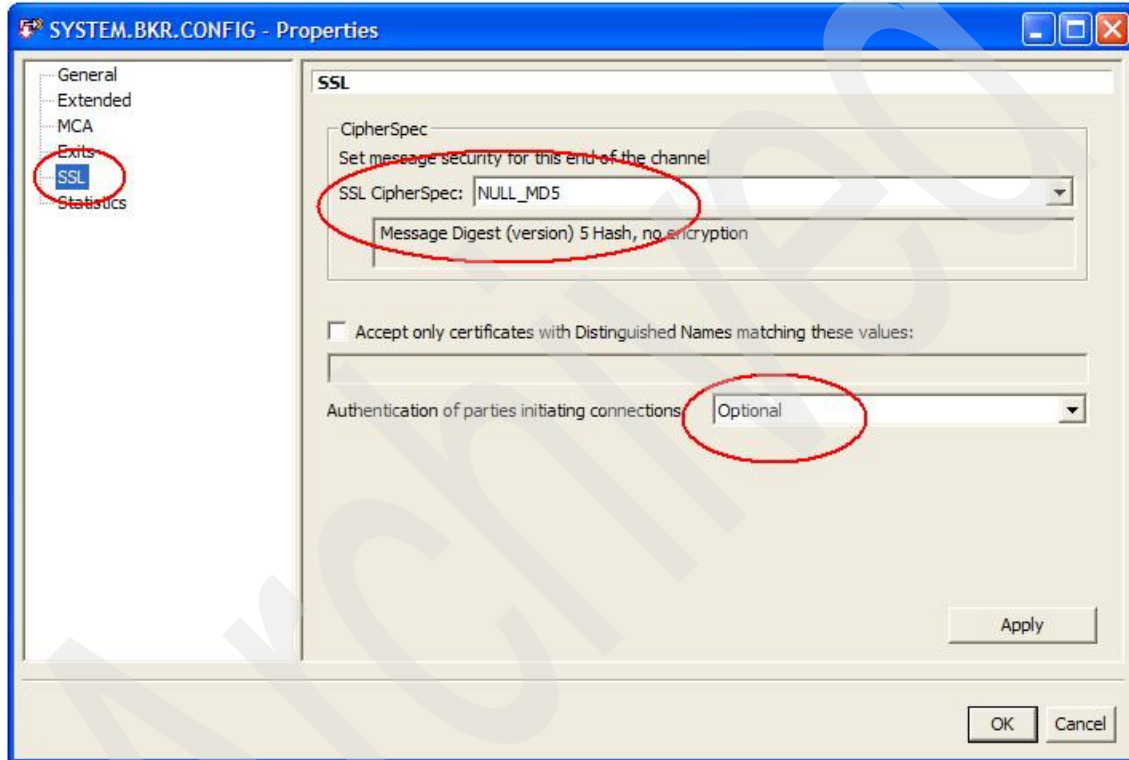


Figure 4-12 SSL channel properties

The queue manager side is ready. We now need to set up the Toolkit domain for SSL:

1. On the Toolkit machine (if different from the Configuration Manager's), switch to the Toolkit window.

The Toolkit should be disconnected from the Configuration Manager. If it is not, switch to the **Broker Administration** perspective and disconnect (or close the Toolkit and start it again).

2. Open the Configuration Manager domain (file extension configmgr), as shown in Figure 4-13.

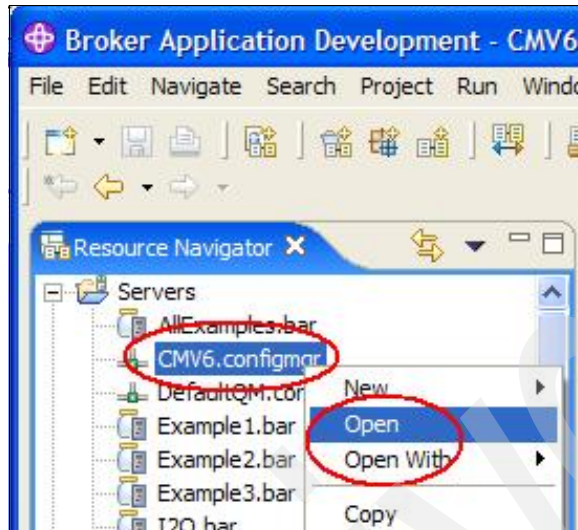


Figure 4-13 Configuration Manager domain

3. This opens the domain editor. Change the SSL settings as follows:

- Cipher Suite: SSL_RSA_WITH_NULL_MD5
- Key Store: C:\MQCLIENT\Toolkit\key.jks
- Trust Store: C:\MQCLIENT\Toolkit\key.jks

4. Save your changes (press Ctrl+s).

We are now ready to test the SSL connection.

Switch to the **Broker Administration** perspective and connect. It should work without having to stop anything. If it does not work, stop the Toolkit, stop the queue manager, restart both, and try again.

After verifying that it works, disconnect the Toolkit from the Configuration Manager. We are now ready to set up mutual authentication.

4.3 Two-way (mutual) SSL authentication

The process to implement mutual authentication consists of:

1. Create a certificate for the Toolkit.
2. Extract the certification authority (CA) part.

3. Transfer the CA part to the queue manager.
4. Install (import) the CA part in the queue manager's key repository.
5. Alter the channel to request client authentication.

4.3.1 Creating a certificate for the Toolkit

To create a certificate for the Toolkit:

1. *On the Toolkit machine*, open a command prompt and start the iKeyman GUI (`strmqikm`).
2. Select **Key Database File** → **Open**.
3. Set the following values in the next window (Figure 4-14):
 - Key database type: **JKS**
 - File Name: `key.jks`
 - Location: `C:\MQCLIENT\Toolkit`
 Click **OK**.

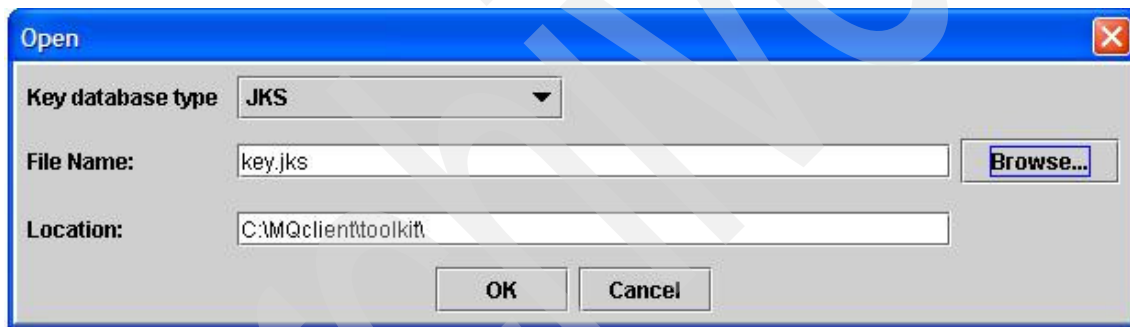
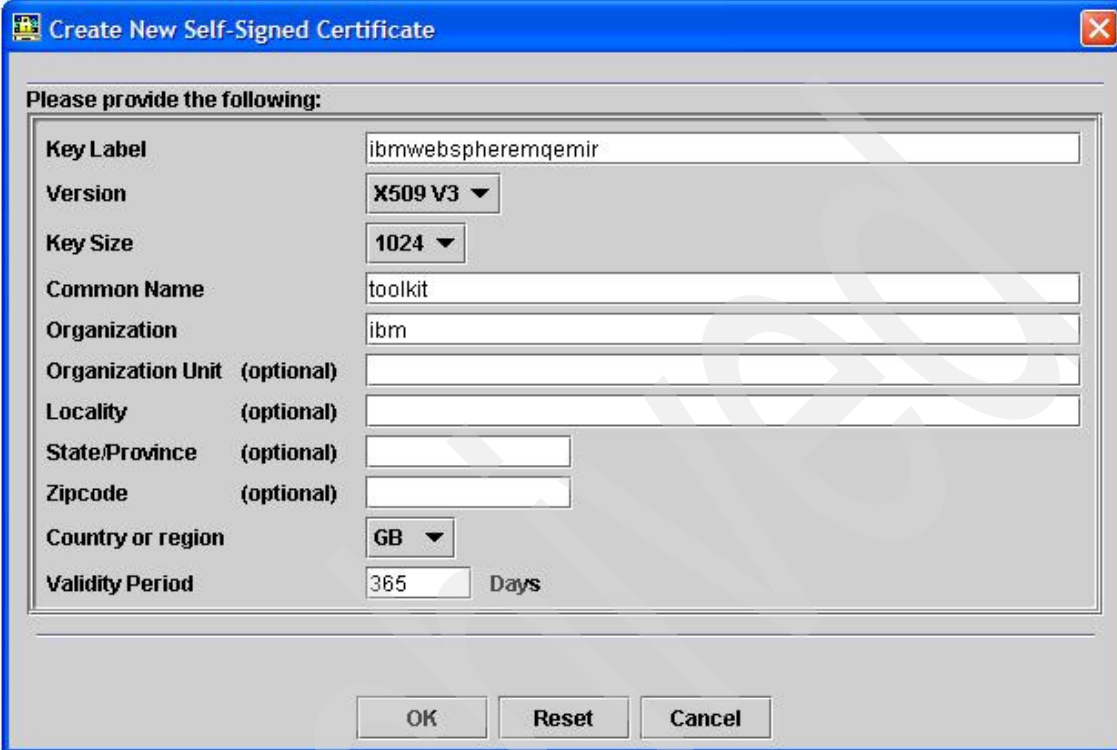


Figure 4-14 Open JKS key repository

4. At the password prompt, enter `changeit`.
5. Switch to the **Personal Certificates** view (menu near top-right corner).
Click **New Self-Signed** (bottom-right corner).
6. Enter the following values for the certificate as shown in Figure 4-15 on page 126:
 - Key Label: `ibmwebspheremq<toolkit user ID>`
In our case, the user ID is `emir`, so the label is `ibmwebspheremqemir`.
 - Common Name: `toolkit`
(You can give it any name you want.)
 - Organization: Your company or your department

Allow everything else to default.
Click **OK**.



The image shows a Windows-style dialog box titled "Create New Self-Signed Certificate". It contains several input fields and dropdown menus for configuring a certificate. The fields are: Key Label (text box with "ibmwebspheremqemir"), Version (dropdown with "X509 V3"), Key Size (dropdown with "1024"), Common Name (text box with "toolkit"), Organization (text box with "ibm"), Organization Unit (optional, empty text box), Locality (optional, empty text box), State/Province (optional, empty text box), Zipcode (optional, empty text box), Country or region (dropdown with "GB"), and Validity Period (text box with "365" and "Days" label). At the bottom are three buttons: "OK", "Reset", and "Cancel".

Key Label	ibmwebspheremqemir
Version	X509 V3
Key Size	1024
Common Name	toolkit
Organization	ibm
Organization Unit (optional)	
Locality (optional)	
State/Province (optional)	
Zipcode (optional)	
Country or region	GB
Validity Period	365 Days

Figure 4-15 Create New Self-Signed Certificate

The new certificate appears in the Personal Certificates list.

We now need to extract the certification authority part of the certificate and install it on the queue manager repository.

4.3.2 Extracting the CA certificate

To extract the CA certificate:

1. If not already selected, select (click) the certificate you just created.
Click **Extract Certificate** (bottom-right corner).
2. Extract the CA certificate using these values:
 - Certificate File Name: toolkit.arm
 - Location: C:\MQCLIENT\Toolkit

Click **OK**.

This create a file called toolkit.arm in the C:\MQCLIENT\Toolkit directory.

4.3.3 Transferring the CA file

To transfer the CA file:

1. Copy (or FTP in ASCII if on a different machine) toolkit.arm to the queue manager directory C:\MQV6\Qmgrs\MBV6\ssl.
2. Close the file (**Key Database File** → **Close**).
3. Close the iKeyman GUI.

4.3.4 Installing the CA certificate in the queue manager

To install the CA certificate in the queue manager:

1. On the Configuration Manager queue manager machine, start the iKeyman GUI (**strmqikm**).
2. Open the queue manager's key repository. Select **Key Database File** → **Open**.
3. Set the following values:
 - Key database type: **CMS**
 - File Name: key.kdb
 - Location: C:\MQV6\Qmgrs\MBV6\ssl
4. Switch to the **Signer Certificates** repository.
Click **Add** (top-right corner).
5. Enter these values:
 - Certificate File Name: toolkit.arm
 - Location: C:\MQV6\Qmgrs\MBV6\ssl

Click **OK**. This imports the CA certificate from the toolkit.arm file.

6. When prompted for a label, enter toolkit (or any name that helps you identify this as the Toolkit's CA certificate).

The newly imported CA certificate appears in the list of Signer Certificates.

At this point, the Toolkit has its personal certificate, and the queue manager has the CA certificate that enables it to authenticate the Toolkit.

7. Close the file (**Key Database File** → **Close**).
8. Close the iKeyman GUI.

We are ready to test mutual (two-way) authentication.

4.3.5 Setting up and testing two-way SSL

Make sure that the Toolkit is disconnected from the Configuration Manager (disconnect it if necessary). To set up and test two-way SSL:

1. Switch to WebSphere MQ Explorer **Channels** view.
2. Right-click **SYSTEM.BKR.CONFIG**. Select **Properties**.
3. Select the **SSL** tab.
 - SSL CipherSpec is already set to **NULL_MD5**.
 - Set Authentication of parties initiating connections to **Required**.Click **OK**.
4. No changes are needed in the Toolkit. We are ready to start the connection.
Close the Toolkit and stop the queue manager, and then restart both.
5. Connect the Toolkit to the Configuration Manager.

This concludes the instructions for connecting the WebSphere Message Broker V6 Toolkit using SSL.

4.4 What could go wrong?

While writing this, and sometimes not paying attention to our own instructions, we encountered the problems described in this section.

4.4.1 Unable to access stashed password

This error might occur when you:

1. Create a CMS key repository (see 4.2.1, “Creating a self-signed certificate for the queue manager” on page 110).
2. Give it a password.
3. Select **Stash password to a file**.
4. Attempt to start an SSL connection before closing the key repository (**Key Database File** → **Close**).

To recover from this error:

1. Start the iKeyman GUI (**strmqikm**).
2. Open the CMS key database file (**Key Database File** → **Open**).

3. Stash the password (**Key Database File** → **Stash Password**).
4. Close the key database file (**Key Database File** → **Close**).

4.4.2 WebSphere MQ reason code 2397

If you create a JKS key repository (see 4.2.3, “Installing the CA part in the Toolkit’s key repository” on page 116) with a password other than `changeit`, you must pass the password when starting the Toolkit. If you do not pass the password, you get the following error (Figure 4-16).

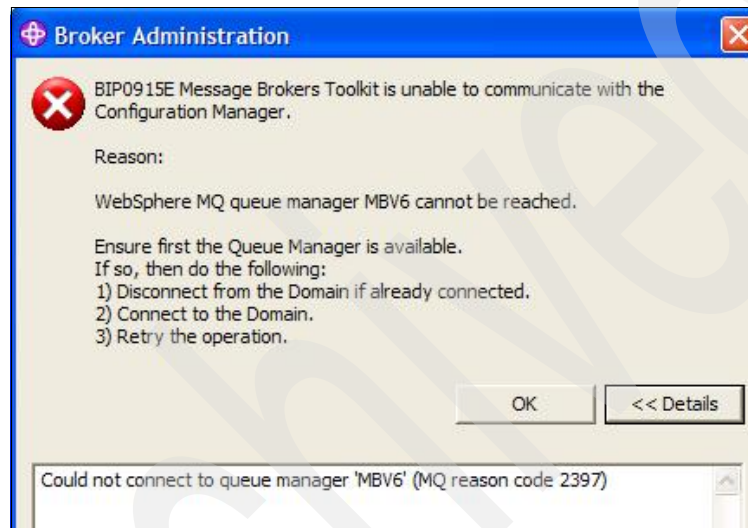


Figure 4-16 MQRC 2397

The easiest solution is to change the password:

1. Start the iKeyman GUI (`strmqikm`).
2. Open the JKS key repository (**Key Database File** → **Open**).
3. Change the password (**Key Database File** → **Change Password**) to `changeit`.
4. Close the key repository (**Key Database File** → **Close**).

If you must use a different password, pass it to the JVM when starting the Toolkit, for example:

```
<Toolkit>\wmbt.exe -vmargs -Djavax.net.ssl.keyStorePassword=passw0rd
```

Remember to place `-vmargs` as the last parameter. If you have, for example, `-data`, place it before `-vmargs`.

4.4.3 WebSphere MQ reason code 2009

Under certain circumstances, if you do not close the Toolkit and stop the queue manager (see 4.3.5, “Setting up and testing two-way SSL” on page 128), you receive the message shown in Figure 4-17.



Figure 4-17 MQRC 2009

The solution is to stop the queue manager and the Toolkit, and then restart both.

4.4.4 Configuration Manager proxy retry attempts

When attempting to connect the Toolkit, you might get an error message directing you to:

Select **Window** → **Preferences** → **Broker Administration** → **Configuration Manager Proxy** and increase the Maximum Retry Attempts or the time between attempts, or both.

We found that this is a transient error (probably due to a Java class loading delay). Whenever we tried again, it worked.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 132. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *WebSphere MQ Security in an Enterprise Environment*, SG24-6814

Other publications

These publications are also relevant as further information sources:

- ▶ *WebSphere MQ Security*, SC34-6588
- ▶ *WebSphere MQ Intercommunication*, SC34-6587
- ▶ *WebSphere MQ Script (MQSC) Command Reference*, SC34-6597
- ▶ *WebSphere MQ Messages*, GC34-6601

Online resources

These Web sites are also relevant as further information sources:

- ▶ The WebSphere MQ Information Center
<http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp>
- ▶ The WebSphere Message Broker Information Center
<http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>
- ▶ WebSphere MQ library
<http://www.ibm.com/software/integration/wmq/library/>
- ▶ SupportPac MO04 (WebSphere MQ SSL Wizard)
http://www.ibm.com/support/docview.wss?rs=171&uid=swg24010367&loc=en_US&cs=utf-8&lang=en

- Technote “Unable to import a PKCS12 file that is created by IIS or other non-IBM Web server keystores into a CMS or JKS database”

<http://www.ibm.com/support/docview.wss?uid=swg21201170>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- AIX 5L instructions 94
- AIX 5L prerequisites 91
- AIX 5L to Windows 96
 - altering channel attributes 105
 - assumptions 96
 - creating certificate for Windows queue manager 98
 - creating key repository for WIN2 103
 - downloading certificate to Windows 102
 - example MQSC 96
 - exporting certificate 101
 - importing the certificate 104
- authentication
 - one-way (server) SSL 110
 - SSL client 38
 - SSL server 27
 - two-way (mutual) SSL 124

C

- CA (certification authority) 32
- CA certificate
 - extracting 114, 126
 - installing 127
- certificate
 - labels 47
 - X.509 distinguished name 47
- certification authority (CA) 32
 - certificate 47
 - setup 47
- certification paths 47
- channel initiator 63
- channel setup
 - QM1 18
 - QM2 19
- command
 - ftp 78
 - gsk7cmd 92
 - MQSC 25
 - runmqsc 37, 43
 - set mqchl 25
 - strmqikm 6, 13, 28, 33, 39, 41, 79, 111, 116, 125, 127–129

- connecting certificate to key ring 66
- connecting Toolkit and Configuration Manager
 - process overview 110
- creating
 - certificate for the Toolkit 125
 - client connection channel 25
 - key repository 58
 - queue manager 2
 - queue manager certificate 63
 - root certificate 48
 - self-signed certificate 31
 - self-signed certificate for client 39
 - self-signed certificate for queue manager 110

E

- enabling SSL on queue managers 57
- environment variable
 - MQCHLLIB 24, 26
 - MQCHLTAB 24, 26
 - MQSERVER 23
 - MQSSLKEYR 36
 - setup 26
- extracting CA certificate 114, 126

I

- IBM Key Management (iKeyman) GUI 6, 28, 111
- installing CA certificate 127

J

- job log 67

K

- key database type 117
- key repository
 - adding CA file to Toolkit 118
 - creating 58
 - installing CA part 33
 - installing certificate 10
 - JKS 117
 - key ring 58
 - verifying location 19
- key ring

- assigning 60
- connecting to CA certificate 59
- enabling SSL tasks 60

M

- MQ client setup
 - process overview 22
- MQ Explorer 70
- MQ ISPF panel 68
- MQ SSL configuration 46
- MQ SSL on z/OS, AIX 5L, and Windows 45

N

- naming conventions 47
- non-SSL client connectivity, verifying 23
 - using channel tables 24
 - using MQSERVER 23
- non-SSL WMQ client setup 22

O

- one-way (server) SSL authentication 110
- one-way authentication 5
- one-way SSL 22
 - testing connection 120

P

- personal certificate 5
- process overview
 - connecting Toolkit and Configuration Manager 110

Q

- queue manager
 - basic configuration 2
 - channel setup 3
 - creating 2
 - key repository 6, 28
 - creating self-signed certificate 110
 - exporting certificate from Internet Explorer 10
 - importing certificate 13
 - installing CA part in key repository 41
 - installing certificate 10
 - name 2
 - obtaining certificate 8
 - SSLQM 22
 - creating key repository 28

R

- RACDCERT 48
- RACF main certificates panel 53
- receiver channel 2
- Redbooks Web site 132
 - Contact us xi

S

- SDSF DA panel 61
- Secure Sockets Layer (SSL) 1, 21
 - basics 5
 - sender channel 2
- Signer Certificates 83
- SSL (Secure Sockets Layer) 1
- SSL authentication and test
 - channel setup 18
- SSL certificates 48
- SSL client 5
- SSL client authentication 38
- SSL connection
 - process overview 6
- SSL handshake 5
- SSL server 5
- SSL server authentication 27
- starting channel 20

T

- testing MQ client with channel tables 27
- testing SSL client authentication 42
- testing SSL server authentication 36
- transferring CA file 127
- transferring CA file to Toolkit 116
- transferring channel table file 26
- troubleshooting
 - Configuration Manager proxy retry attempts 130
 - MQ reason code 2009 130
 - MQ reason code 2397 129
 - unable to access stashed password 128
- two-way (mutual) SSL authentication 124
- two-way SSL 22
 - setup 128

W

- WebSphere MQ installation directory 2
- Windows to Windows 105
 - altering channel attributes 107

assumptions 105
example MQSC 105

Z

z/OS to AIX 5L 84
 altering channel attributes 95
 assumptions 84
 creating key repository for AIX1 91
 downloading certificate to AIX 5L 91
 exporting certificate 89
 importing certificate for AIX1 95
 sample JCL for queue manager 84
 sample MQSC commands for queue manager
 AIX1 85
z/OS to Windows 71
 altering channel attributes 83
 assumptions 72
 creating certificate for WIN1 74
 creating key repository for WIN1 79
 downloading certificate to Windows 78
 example CSQUTIL JCL 72
 export certificate from RACF 77
 import certificate 79
z/OS to z/OS 55



Redpaper

WebSphere MQ V6, WebSphere Message Broker V6, and SSL

WebSphere MQ SSL channels on Windows

WebSphere MQ V6 and WebSphere Message Broker V6 (the Toolkit)

Connecting WebSphere MQ and Message Broker using SSL

This IBM Redpaper provides step-by-step guides to implement IBM WebSphere MQ Secure Sockets Layer (SSL) channels in a variety of configurations:

- ▶ Microsoft Windows to and from Windows
- ▶ WebSphere MQ clients to WebSphere MQ queue managers (both on Windows)
- ▶ Any-to-any WebSphere MQ channel connections on IBM z/OS, AIX 5L, and Windows, using RACF as the certification authority
- ▶ WebSphere Message Broker Toolkit

The aim is for you to learn the basics of WebSphere MQ SSL using simple connectivity examples.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks