IBM WebSphere
Technical Conference
2011
October 10-14, 2011
Berlin, Germany

ML2 Hands-on Lab: WebSphere MQ Security (Distributed Platforms)

Morag Hughson | hughson@uk.ibm.com

T.Rob Wyatt | t.rob.wyatt@us.ibm.com

Author: T.Rob Wyatt
t.rob.wyatt@us.ibm.com

Author: Morag Hughson
hughson@uk.ibm.com

Download the most current version any time at https://t-rob.net/links

Document History

Document Location

This is a snapshot of an on-line document. Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

Revision History

Date of this revision: 12 October 2011	Date of next revision *(date)*

Revision Number	Revision Date	Summary of Changes
(#)	(-)	(Describe change)
3	20100901	Correct JKS name in Module #3
3	20100916	WTC Branding
4	20110227	IMPACT 2011 branding, more updates from WTC, SupportPac MO05
5	20110319	New WebSphere Services Zone slide, SupportPac MH05
6	20110415	Updates from IMPACT 2011 session
7	20110913	Updates for WebSphere MQ V7.1 new function
8	20011012	Updates from Berlin WebSphere Tech Conference 2011 session

Contents

I. Preface: Key reference notes for Lab

A. Passwords and shortcuts

Password information	
WMQ admin privileged accounts	User: MQLab Password: mq1lab User: mqm Password: N/A
Non-privileged accounts	User: mqadmin Password: N/A
Channel service accounts	User: mqmmca User: mqmmqi Password: N/A
Linux Server Note: Do not sign onto lab as root or you will not be able to run WMQ Explorer.	User: root Password: mq2lab
Names and locations of server processes, workspaces, etc.	
MARS	/var/mqm/qmgrs/MARS
VENUS	/var/mqm/qmgrs/VENUS
QMgr keystores	/var/mqm/qmgrs/<qmgr name>/ssl
User keystores (for WMQ Explorer)	/home/MQLab/ssl
Naming conventions	
Supporting files	/home/MQLab/Desktop/LabFiles
Modules	Each module has a corresponding directory with a leading numerical prefix to match the module number.
Scripts	Each folder has a build.ksh script that deletes the queue managers and rebuilds the environment to the state required to begin the next module. For example, running the build script for Module 4 completes the configurations from modules 1 thru 4 to ready the environment to begin Module 5. Scripts for each module are numbered to correspond to the exercises. Supporting files are not numbered but may be called from within the scripts.
File system shortcuts	
$PATH	The $PATH variable has been modified to include /opt/mqm/bin, /opt/mqm/samp/bin and '.' (the current directory).

1. Module 1 – Introduction

The objective of this module is to familiarize you with the objectives of the lab as well as the VMware image and the various tools that will be used throughout the lab.

1.1 Course Objectives

The material presented in this course is intended to accomplish the following:

- Familiarize you with the WebSphere MQ authorization commands.
- Familiarize you with some of the security-related WebSphere MQ SupportPacs.
- Gain hands-on experience configuring a security exit to perform authentication.
- Gain hands-on experience configuring SSL to perform authentication.
- Walk through the base hardening tasks for a queue manager.

1.2 VMware Image

The lab is set up to run on a RHEL5 32bit on VMWare Workstation 6.5 with the following components already installed:

- WebSphere MQ v7.1 Server and Client
- WebSphere MQ Explorer with SupportPac MS0P installed
- IBM Global Security Kit (GSKit) V8

In addition, the following components have been downloaded and installed

- SupportPacs
    - MA01 – Q Program
    - MO04 – SSL Wizard
    - MS0P – WebSphere MQ Explorer Configuration and Display Extension Plug-ins
    - MSL1 – Linux Init Scripts
- Several shell scripts residing in /home/MQLab/scripts

1.3 Accounts & Groups

Several accounts and groups have been pre-configured for use in this course and are listed in the format account:group[, group…]. These include the following:

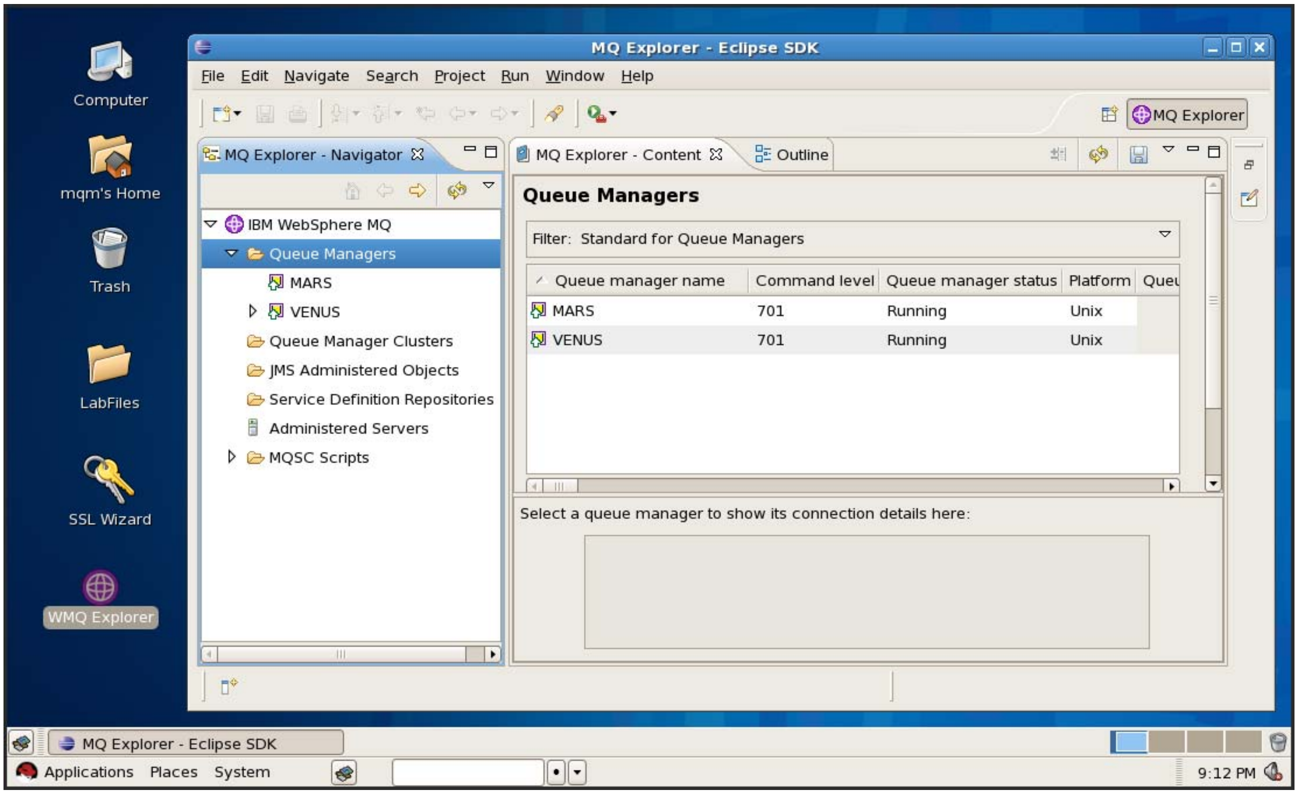
- root:root, bin, daemon, sys, adm, disk, wheel (Note that root is NOT in the mqm group.)
- mqm:mqm
- MQLab:mqm
- mqadmin:mqadmin (disabled, non-login account)
- mqmmca:mqmmca (disabled, non-login account)
- mqmmcq:mqmmqi (disabled, non-login account)

The password for all active accounts is listed in the Preface.

Please sign in as MQLab if not already signed in, and take a moment to locate the various software assets described above. The Linux image has a desktop manager that provides many of the same functions as a Windows desktop. Double-click on the Computer icon on the desktop to open the Nautilus file browser. You can use this to browse through the file system similar to Windows Explorer.

If you prefer a command line, right-click anywhere on the desktop and select the Open Terminal option.

Once signed on, feel free to open WebSphere MQ Explorer and view the queue managers.



Queue Managers

Filter: Standard for Queue Managers

Queue manager name	Command level	Queue manager status	Platform	Queu
MARS	701	Running	Unix	
VENUS	701	Running	Unix	

Select a queue manager to show its connection details here:

1.4 Running this lab in your own environment

One of the goals when designing the lab was to allow it to be portable to other environments. For this reason, much of the setup of the lab has been scripted so that it is easy to tear it down and rebuild.

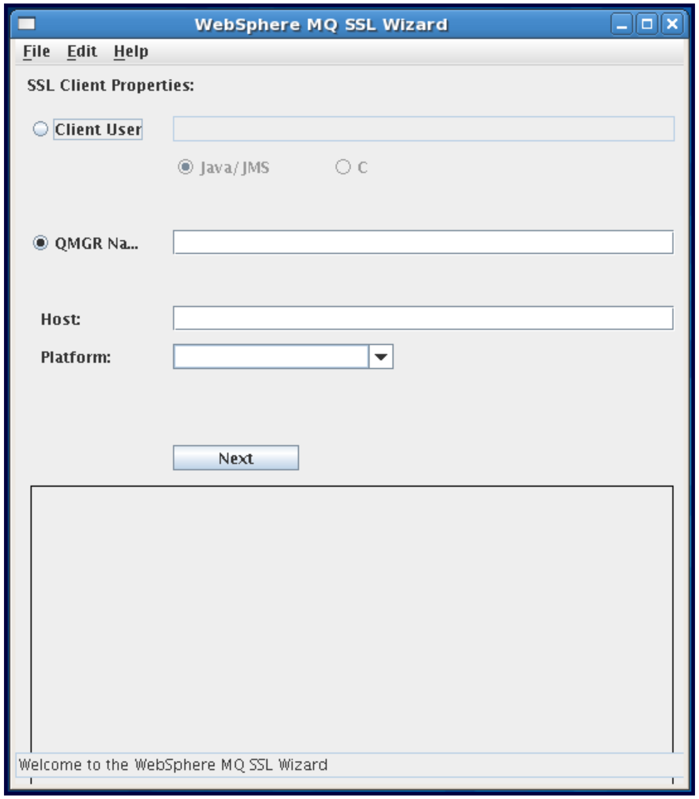
If you are attending this lab at IMPACT or other IBM sponsored event, the initial setup scripts will have already been run for you. This is done so that you can spend your time in the lab focusing on the security content. However, the setup scripts are provided as student materials which may be used along with this guide to set up and execute the lab modules on most UNIX or Linux platforms.

A copy of this lab guide and the scripts can be downloaded at any time from https://t-rob.net/links They have been tested on Linux but should run without (much) modification on other UNIX variants.

2. Use MO04 SSL Wizard to connect two queue managers

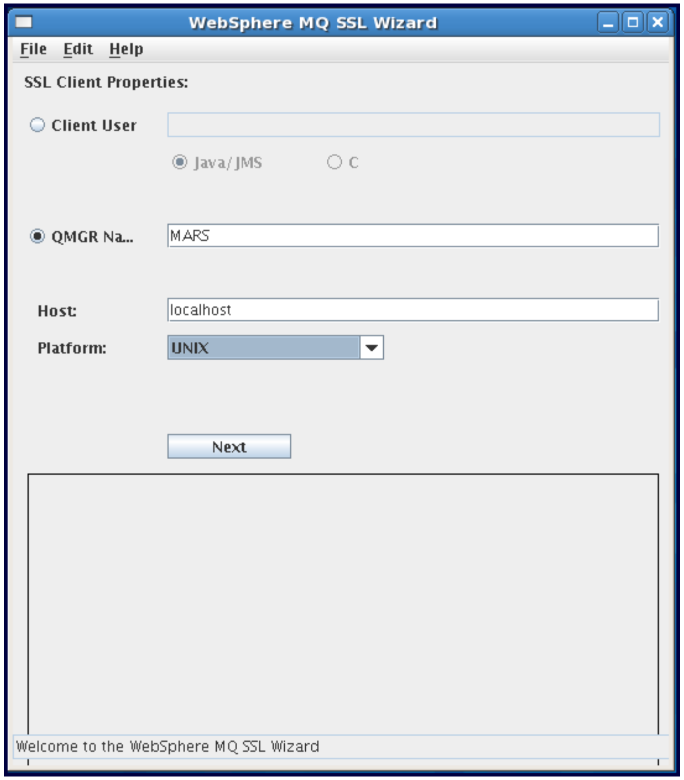
This module will connect the MARS and VENUS queue managers using SupportPac MO04 - SSL Wizard. The wizard is a Java application that prompts for details about the connection and then constructs the commands necessary to build the SSL keystores, certificates and channel definitions. The VMWare image will automatically log you on as MQLab (a user in the mqm group).

2.1 Running the Wizard



Welcome to the WebSphere MQ SSL Wizard

1) ____ To get started, double-click the SSL Wizard icon on the desktop.



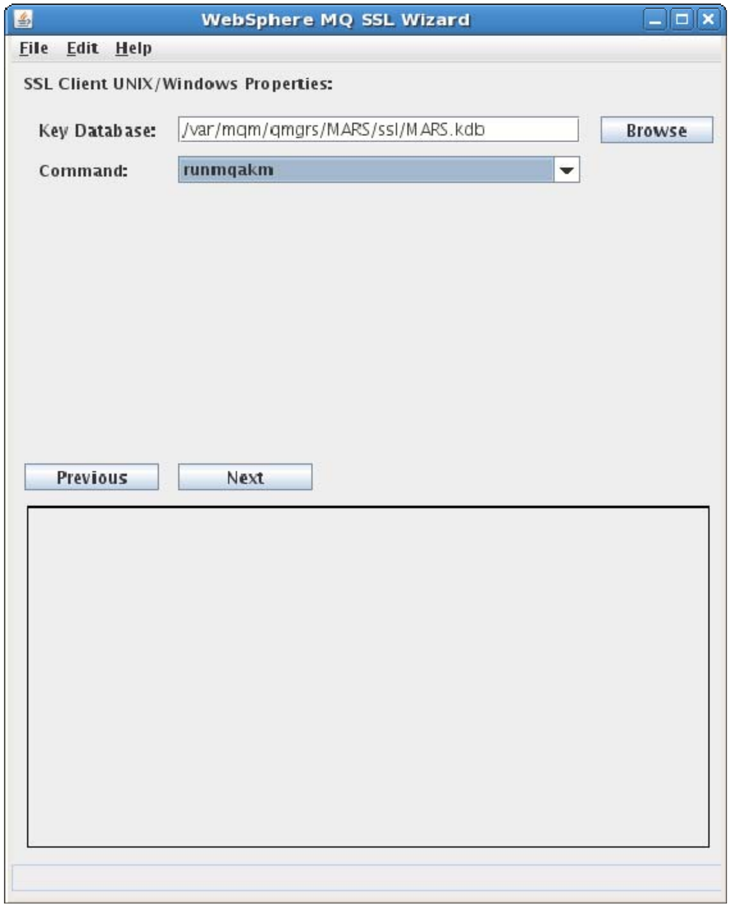
● QMGR Na... MARS

Host: localhost

Platform: UNIX

Welcome to the WebSphere MQ SSL Wizard

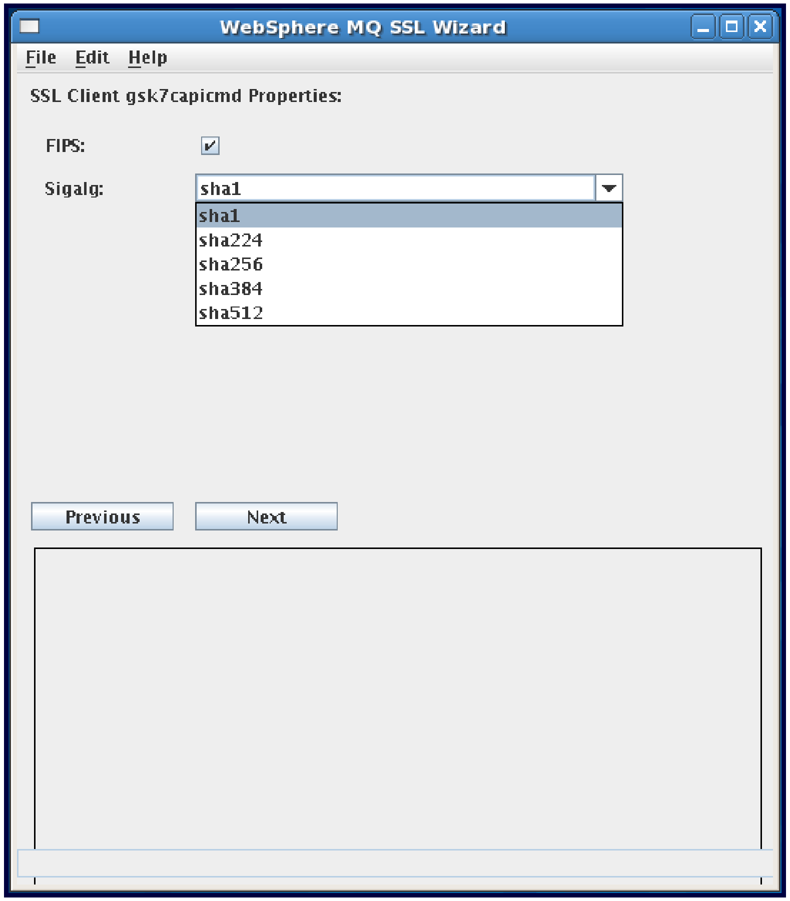
2) ____ In SSL terms, the "client" is the thing that initiates the connection. In this case, we are going to connect MARS to VENUS so MARS is the client. Fill in the details and click Next.



SSL Client UNIX/Windows Properties:

Key Database: /var/mqm/qmgrs/MARS/ssl/MARS.kdb

Command: runmqakm



SSL Client gsk7capicmd Properties:

FIPS: ☑

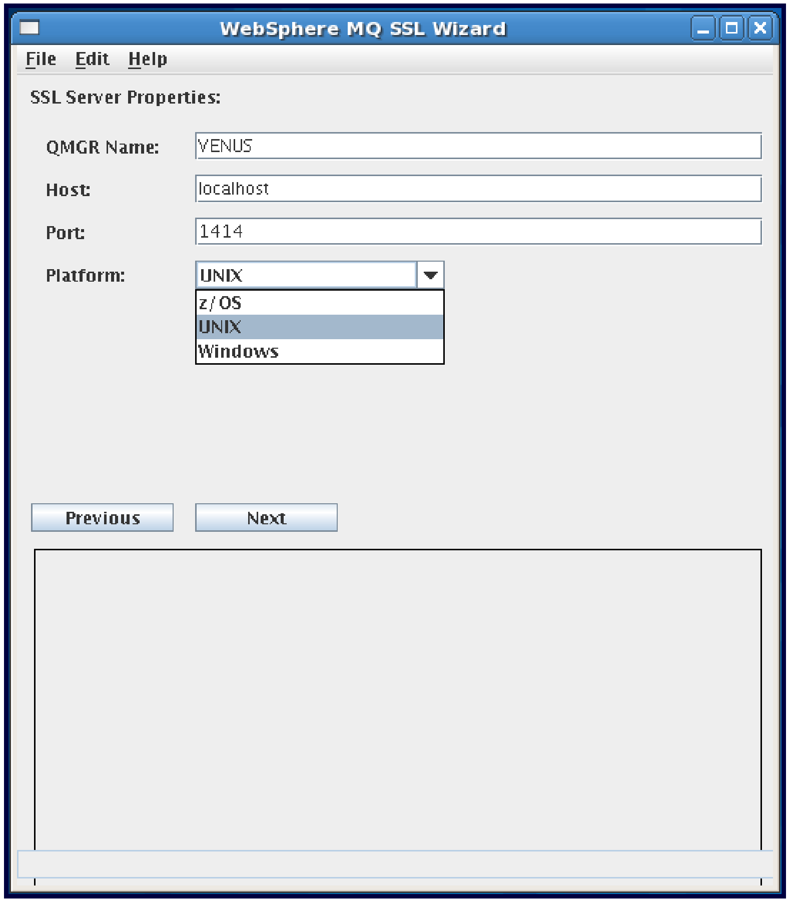
Sigalg: sha1

3) ___ Choose runmqakm from the command drop down. Click Next.

4) ____ The FIPS option uses libraries certified for Federal Information Processing Standards in order to build the keystore and certificates. There is no performance or negative impact to leaving it checked.

The -sigalg option specifies the signing algorithms available for certificate generation. For our purposes, SHA1 is sufficient.

Click Next.



SSL Server Properties:

QMGR Name: VENUS

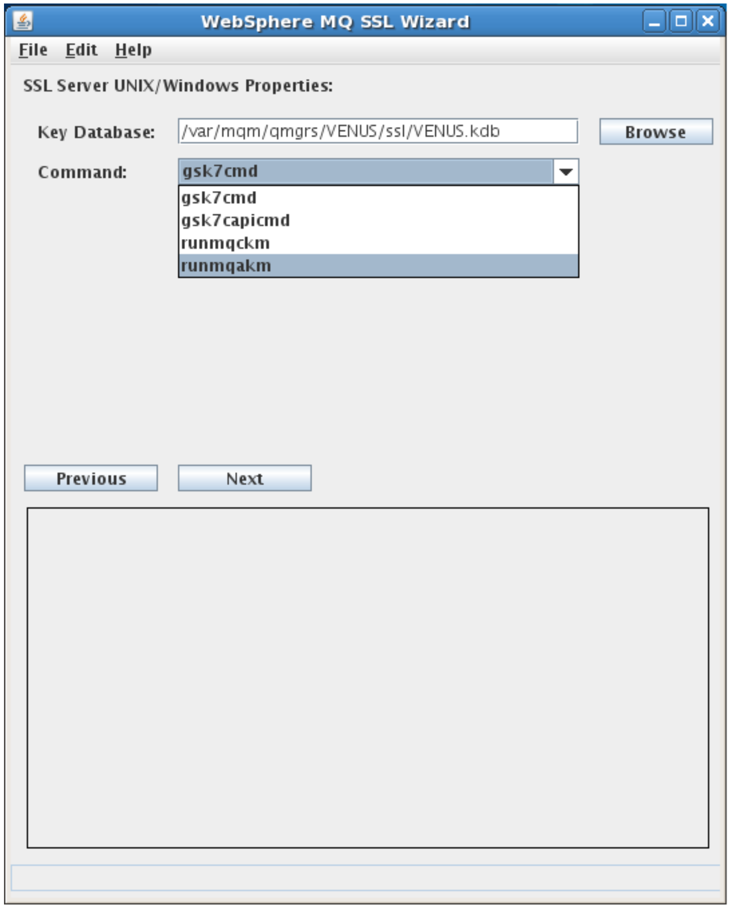
Host: localhost

Port: 1414

Platform: UNIX

5) ____ In SSL terms, the server is the thing to which a connection is being requested. In this case VENUS is the server. For simplicity we will use localhost in the demo. In practice you would put an actual DNS name here. The VENUS QMgr has a listener of port 1414. Select UNIX from the drop-down list for any UNIX or Linux queue managers.

After filling in the necessary fields, click Next.



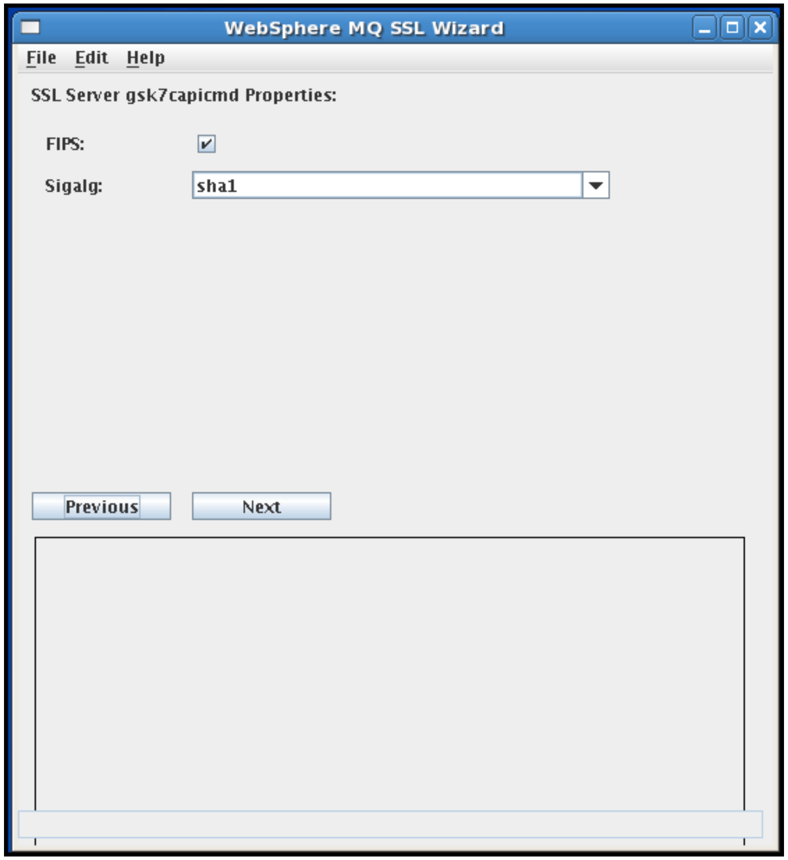
SSL Server UNIX/Windows Properties:

Key Database: /var/mqm/qmgrs/VENUS/ssl/VENUS.kdb

Command: gsk7cmd

6) ____ It is worth mentioning that there are several tools available to manage keystores and certificates. The runmqakm is a compiled C program that runs very quickly and is FIPS capable. You could also use the Java-based runmqckm or the iKeyman GUI. This wizard also allows the choice of the V7.0 command names (commands beginning with “gsk7”).

Ensure that runmqakm is selected and click Next.



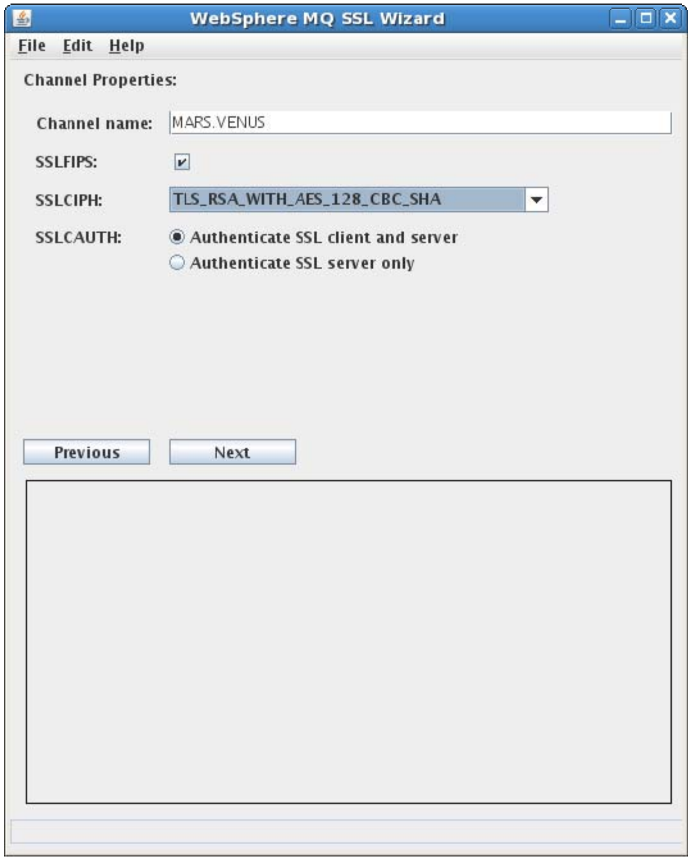
SSL Server gsk7capicmd Properties:

FIPS: ☑

Sigalg: sha1

7) ____ The FIPS option and SHA1 algorithm are still appropriate for this demo. Note that these options apply only at build time when generating the keystore and certificates. Selecting different values here will not prevent the queue managers from communicating.

Click Next.



Channel Properties:

Channel name: MARS.VENUS

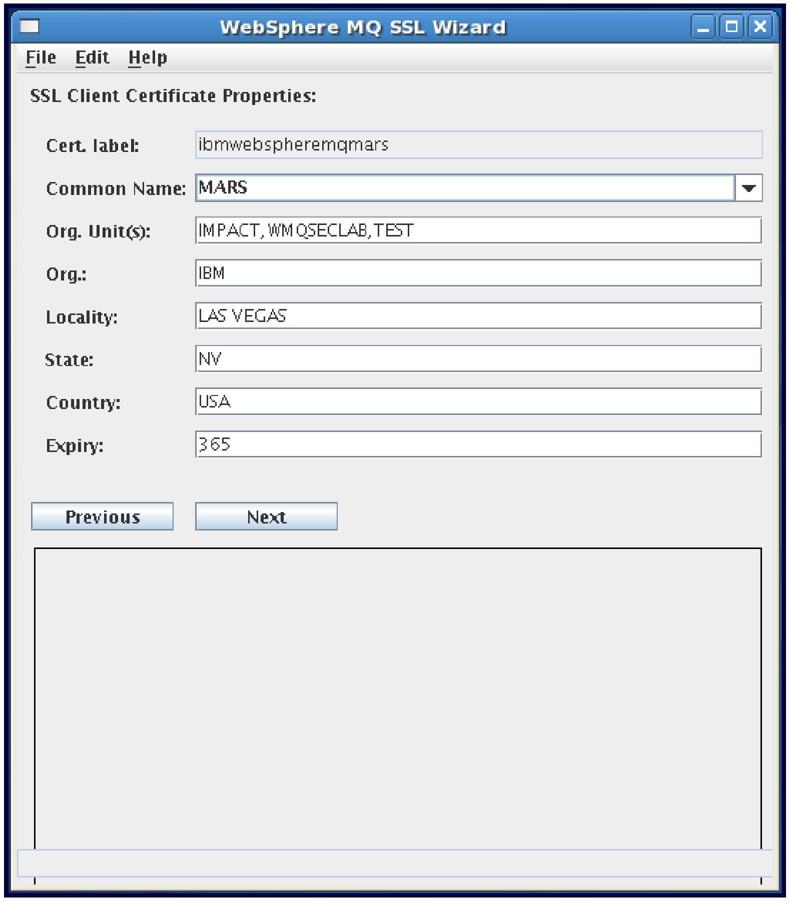
SSLFIPS: ☑

SSLCIPH: TLS_RSA_WITH_AES_128_CBC_SHA

SSLCAUTH: ● Authenticate SSL client and server
○ Authenticate SSL server only

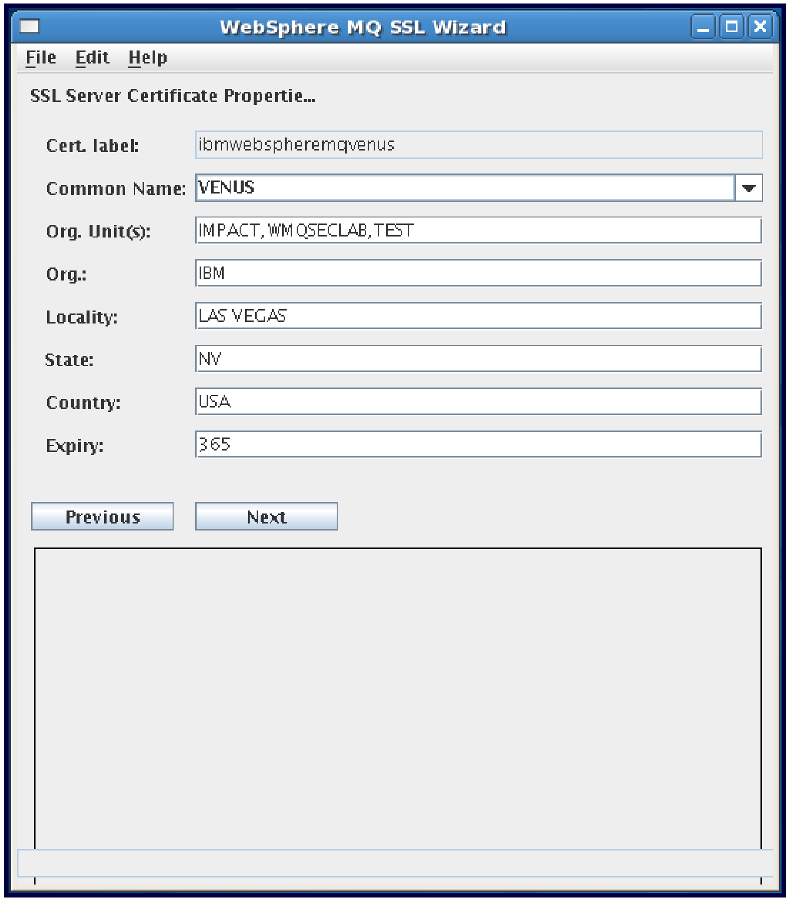
8) ____ This screen allows you to select from among several different cipherspecs. These values do affect interaction at run time and must match on both queue managers. I have enabled the FIPS option which limits your selection to a few of the strongest encryption choices. From these, select the one option named TLS_RSA_WITH_AES_128_CBC_SHA.

Click Next when done making your selections.



SSL Client Certificate Properties:

Cert. label:	ibmwebspheremqmars
Common Name:	MARS
Org. Unit(s):	IMPACT, WMQSECLAB, TEST
Org.:	IBM
Locality:	LAS VEGAS
State:	NV
Country:	USA
Expiry:	365



SSL Server Certificate Propertie...

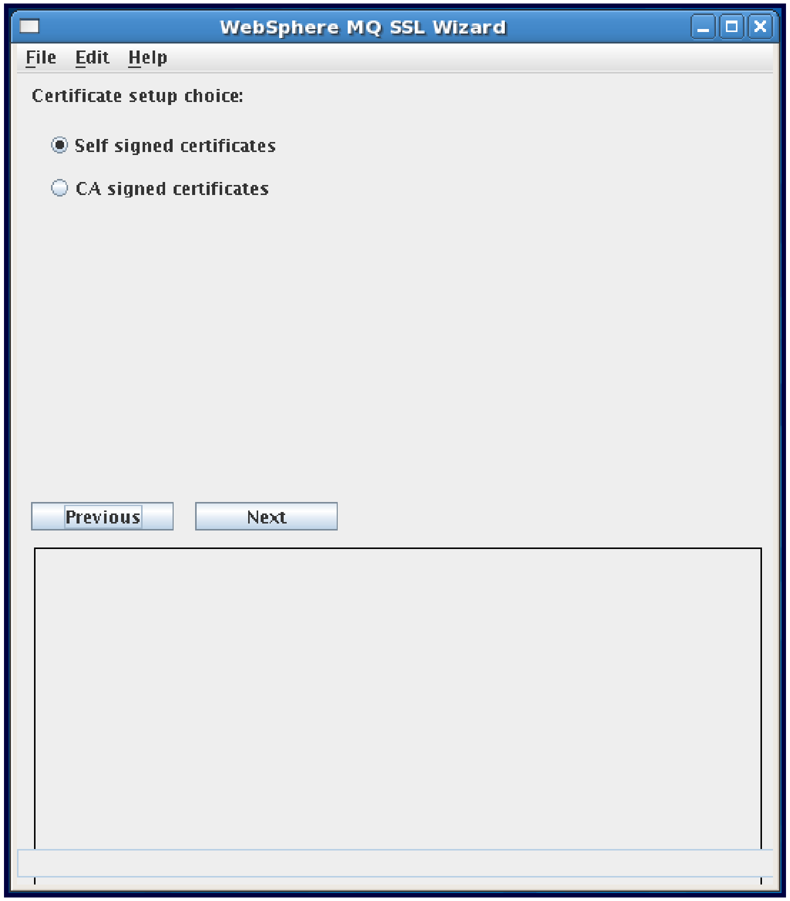
Cert. label:	ibmwebspheremqvenus
Common Name:	VENUS
Org. Unit(s):	IMPACT, WMQSECLAB, TEST
Org.:	IBM
Locality:	LAS VEGAS
State:	NV
Country:	USA
Expiry:	365

9) ____ This screen contains the fields required for the Distinguished Name of a certificate. The choice of a naming convention determines the level of granularity that can be achieved in filtering connection requests. Note that it is possible to enter multiple Organization Unit or OU fields in a comma-separated list. The order and case of these fields is important so pick a standard and do not vary from it. Here we have used all UPPERCASE for consistency.

Enter the requested information and click Next.

10) ____ Enter the same information for the VENUS queue manager and click Next.

Use this to save the instructions, NOT the File/Save menu! The File/Save is to allow you to capture the parameters you just entered so that you can re-start the SSL Wizard later.



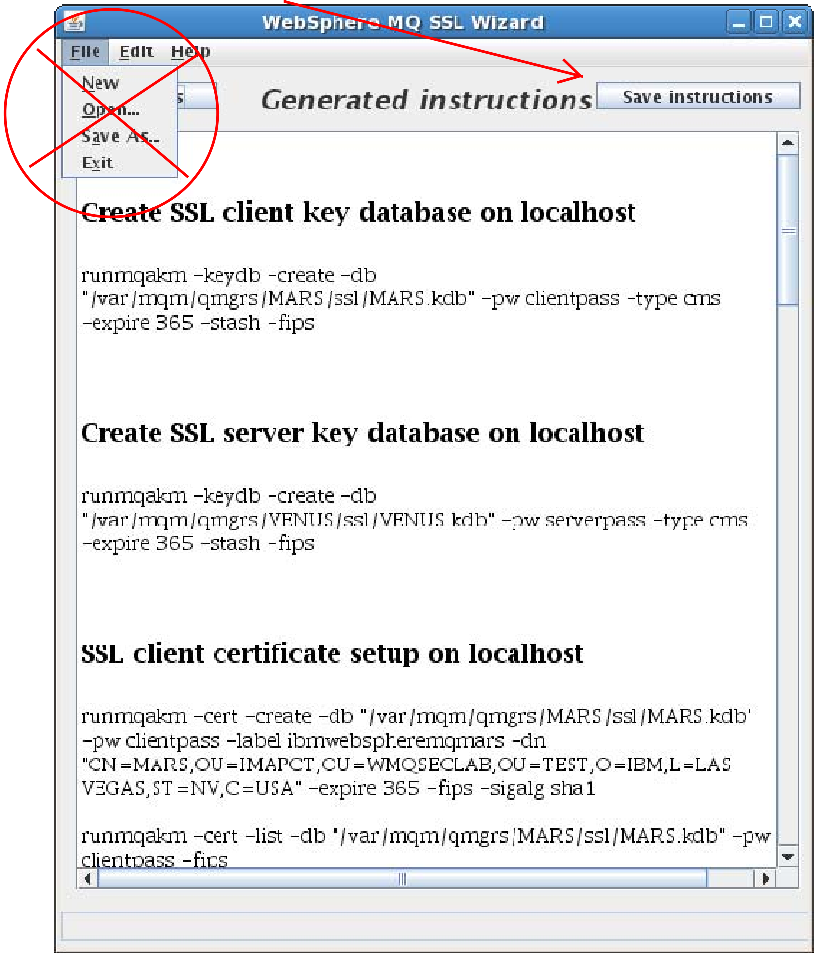
Certificate setup choice:

● Self signed certificates

○ CA signed certificates

11) ____ This screen allows you to choose between self-signed certificates versus using a Certificate Authority. In practice this would be determined by your company's policies or the size of the network. Performance and resource consumption are impacted as the size of the keystore grows. Self-signed certificates are useful in smaller networks but do not scale well.

For the lab we will be using self-signed certificates which is the default. Click Next.



Create SSL client key database on localhost

```
runmqakm -keydb -create -db
"/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -type cms
-expire 365 -stash -fips
```

Create SSL server key database on localhost

```
runmqakm -keydb -create -db
"/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -type cms
-expire 365 -stash -fips
```

SSL client certificate setup on localhost

```
runmqakm -cert -create -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb"
-pw clientpass -label ibmwebspheremqmars -dn
"CN=MARS,OU=IMAPCT,OU=WMQSECLAB,OU=TEST,O=IBM,L=LAS
VEGAS,ST=NV,C=USA" -expire 365 -fips -sigalg sha1

runmqakm -cert -list -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw
clientpass -fips
```

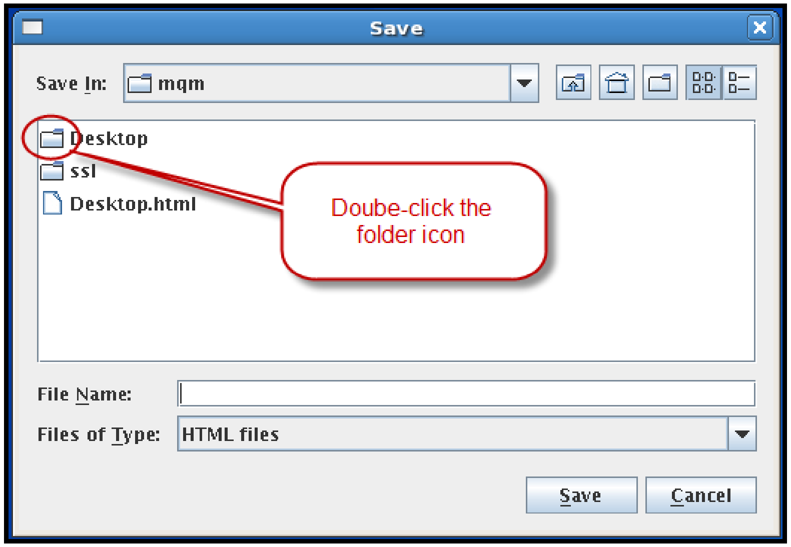
12) ____ The wizard will generate all of the commands to create the keystores, certificates and channels. The commands are listed on the next page. Take a few moments to review them here or in the wizard's output window. Note that the passwords for the keystores have defaulted to 'clientpass' and 'serverpass'. In practice you would want to use a stronger password.

2.2 Output from SSL Wizard

For reference, the output of the SSL wizard from the previous dialogs is listed below:

```
Create SSL client key database on localhost
runmqakm -keydb -create -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -type cms -expire 365 -
stash -fips

Create SSL server key database on localhost
runmqakm -keydb -create -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -type cms -expire 365 -
stash -fips

SSL client certificate setup on localhost
runmqakm -cert -create -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -label ibmwebspheremqmars
-dn "CN=MARS,OU=IMPACT,OU=WMQSECLAB,OU=TEST,O=IBM,L=LAS VEGAS,ST=NV,C=USA" -expire 365 -fips -sigalg
sha1

runmqakm -cert -list -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -fips

SSL server certificate setup on localhost
runmqakm -cert -create -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -label
ibmwebspheremqvenus -dn "CN=VENUS,OU=IMPACT,OU=WMQSECLAB,OU=TEST,O=IBM,L=LAS VEGAS,ST=NV,C=USA" -expire
365 -fips -sigalg sha1

runmqakm -cert -list -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -fips

Copy the public SSL client certificate to the SSL server side
runmqakm -cert -extract -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -label ibmwebspheremqmars
-target MARS.crt -format ascii -fips

runmqakm -cert -add -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -label ibmwebspheremqmars -
file MARS.crt -format ascii -fips

runmqakm -cert -list -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -fips

Copy the public SSL server certificate to the SSL client side
runmqakm -cert -extract -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -label
ibmwebspheremqvenus -target VENUS.crt -format ascii -fips
```

```
runmqakm -cert -add -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -label ibmwebspheremqvenus -
file VENUS.crt -format ascii -fips

runmqakm -cert -list -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -fips

MQSC commands for SSL client side queue manager MARS
NOTE: The step below is optional because SSLKEYR may already be set.
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/MARS/ssl/MARS')

NOTE: The step below is optional because SSLFIPS may already be set.
ALTER QMGR SSLFIPS(YES)

DEFINE CHANNEL('MARS.VENUS') CHLTYPE(SDR) TRPTYPE(TCP) XMITQ('VENUS') CONNAME('localhost(1414)')
SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA) SSLPEER('CN=VENUS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS
VEGAS,ST=NV,C=USA') REPLACE

DEFINE QL(VENUS) USAGE(XMITQ)

MQSC commands for SSL server side queue manager VENUS
NOTE: The step below is optional because SSLKEYR may already be set.
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/VENUS/ssl/VENUS')

NOTE: The step below is optional because SSLFIPS may already be set.
ALTER QMGR SSLFIPS(YES)

DEFINE CHANNEL('MARS.VENUS') CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA)
SSLCAUTH(REQUIRED) SSLPEER('CN=MARS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS VEGAS,ST=NV,C=USA')
REPLACE

MQSC commands for both queue managers
REFRESH SECURITY TYPE(SSL)

MQSC commands for SSL client side queue manager MARS
START CHANNEL('MARS.VENUS')
```

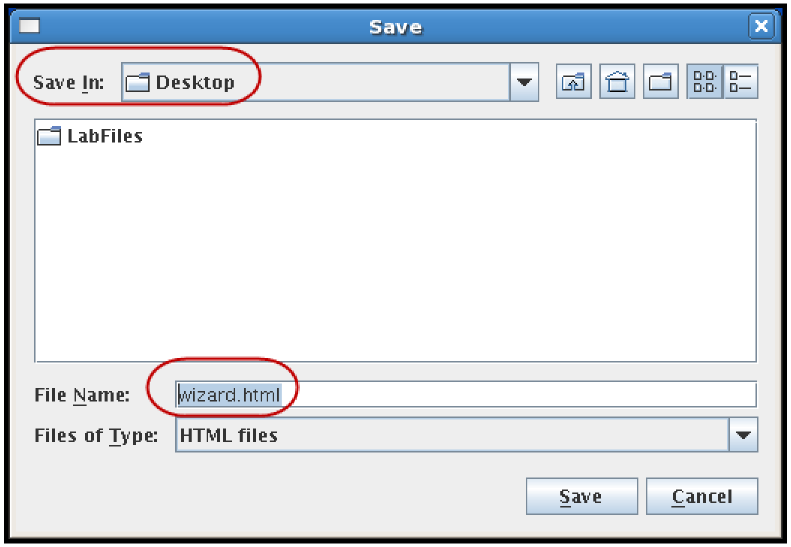


Save In: mqm

File Name:

Files of Type: HTML files

Doube-click the folder icon



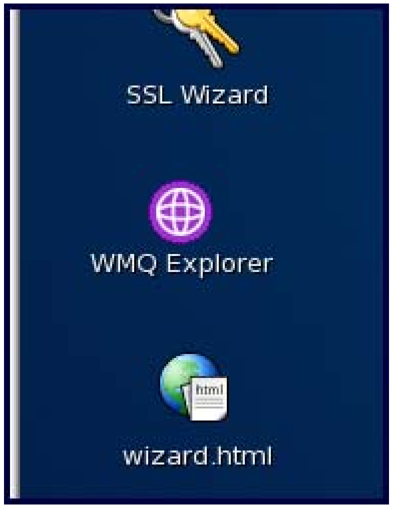
Save In: Desktop

File Name: wizard.html

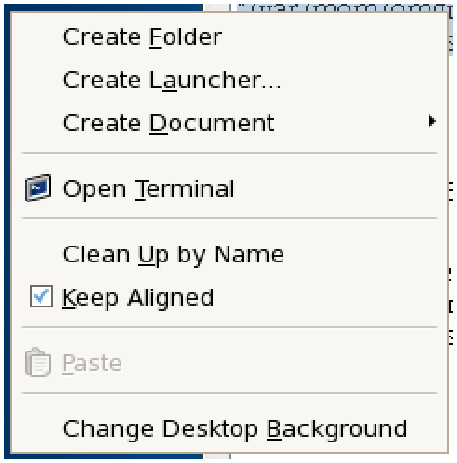
Files of Type: HTML files

13) ____ Click Save Instructions and drill down to the desktop. Save the instructions as wizard.html.

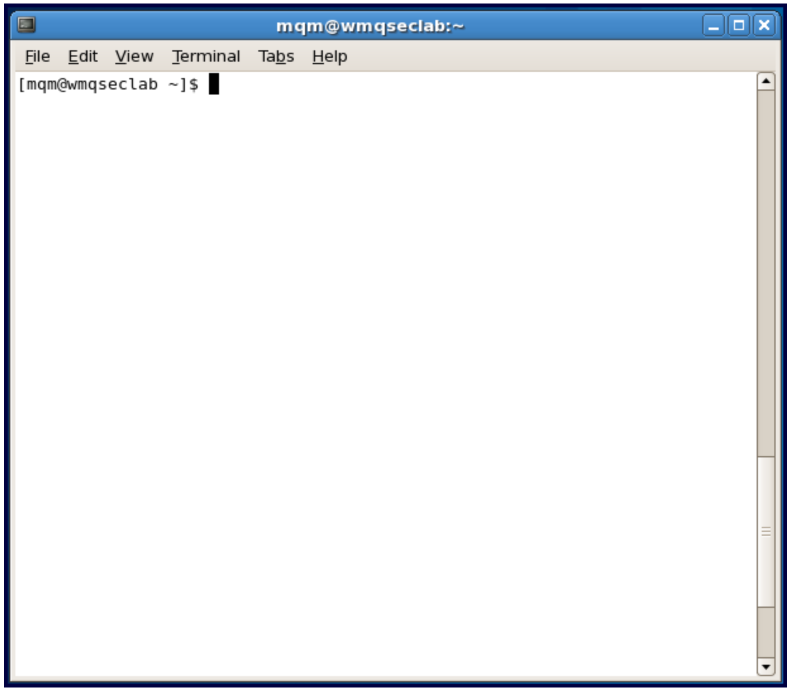
14) ____ Now double-click the wizard.html document to open it in Firefox.



2.3 Executing the Wizard output

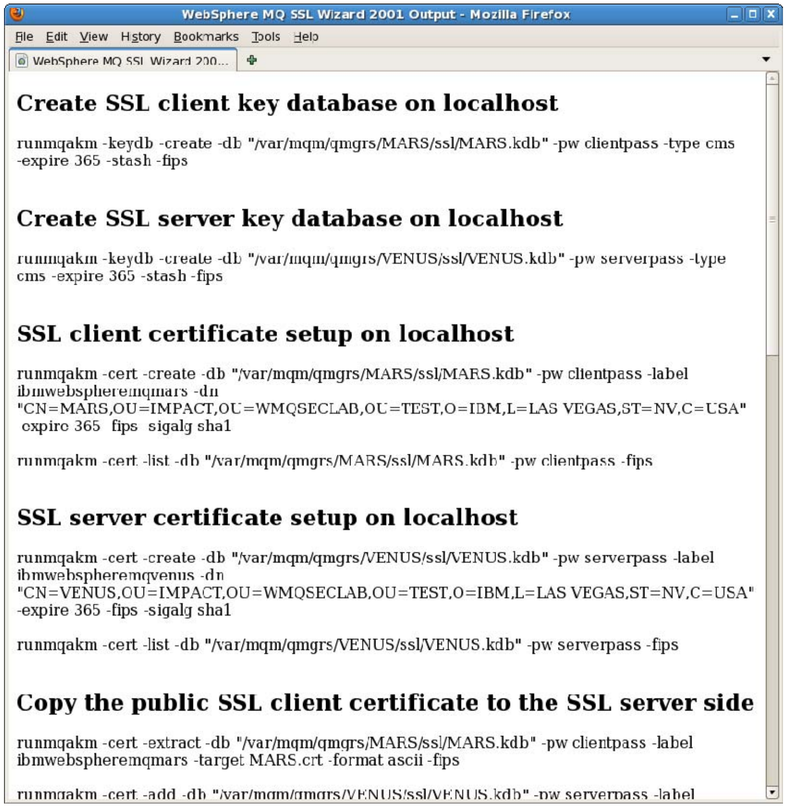


15) ____ Right-click on the desktop and select Open Terminal.



```
[mqm@wmqseclab ~]$ 
```

A terminal window will open where you will be able to enter commands. **When in a terminal window** you can use <SHIFT-CTRL-C> to copy and <SHIFT-CTRL-V> to paste. Most other application, including the text editor, use the normal <CTRL-C> and <CTRL-V> commands.



WebSphere MQ SSL Wizard 2001 Output - Mozilla Firefox
File Edit View History Bookmarks Tools Help
WebSphere MQ SSL Wizard 200...

Create SSL client key database on localhost

```
runmqakm -keydb -create -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -type cms
-expire 365 -stash -fips
```

Create SSL server key database on localhost

```
runmqakm -keydb -create -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -type
cms -expire 365 -stash -fips
```

SSL client certificate setup on localhost

```
runmqakm -cert -create -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -label
ibmwebspheremqmars -dn
"CN=MARS,OU=IMPACT,OU=WMQSECLAB,OU=TEST,O=IBM,L=LAS VEGAS,ST=NV,C=USA"
-expire 365 -fips -sigalg sha1

runmqakm -cert -list -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -fips
```

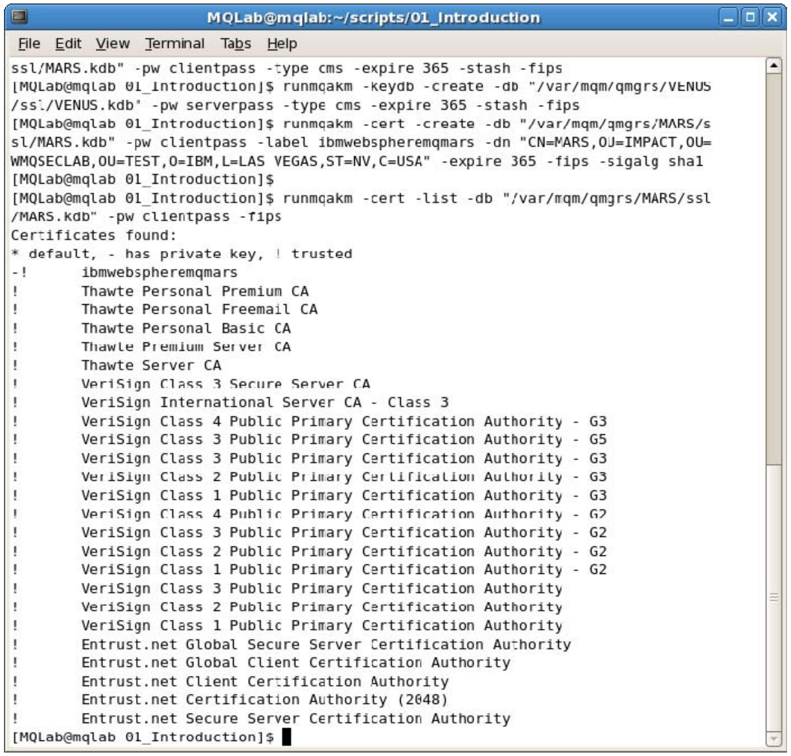
SSL server certificate setup on localhost

```
runmqakm -cert -create -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -label
ibmwebspheremqvenus -dn
"CN=VENUS,OU=IMPACT,OU=WMQSECLAB,OU=TEST,O=IBM,L=LAS VEGAS,ST=NV,C=USA"
-expire 365 -fips -sigalg sha1

runmqakm -cert -list -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -fips
```

Copy the public SSL client certificate to the SSL server side

```
runmqakm -cert -extract -db "/var/mqm/qmgrs/MARS/ssl/MARS.kdb" -pw clientpass -label
ibmwebspheremqmars -target MARS.crt -format ascii -fips

runmqakm -cert -add -db "/var/mqm/qmgrs/VENUS/ssl/VENUS.kdb" -pw serverpass -label
```

16) ____ Copy each of the commands from the browser and paste it into the command-line. You can use <SHIFT-CTRL-V> to paste while in the command line terminal. In the screen shot on the right of this page you can see the result of the first four commands.



MQLab@mqlab:~/scripts/01_Introduction
File Edit View Terminal Tabs Help

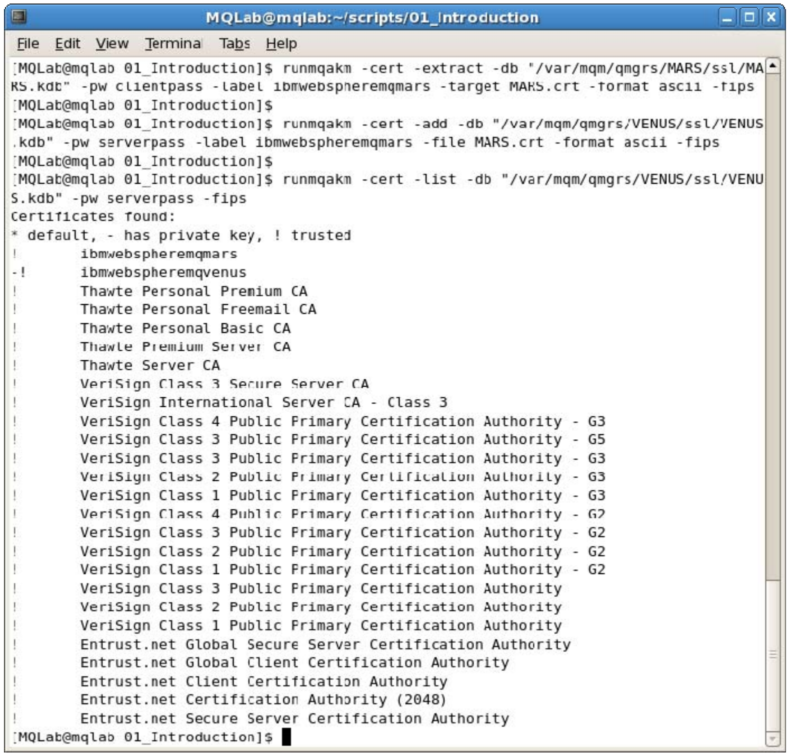
```
ssl/MARS.kdb" -pw clientpass -type cms -expire 365 -stash -fips
[MQLab@mqlab 01_Introduction]$ runmqakm -keydb -create -db "/var/mqm/qmgrs/VENUS
/ssl/VENUS.kdb" -pw serverpass -type cms -expire 365 -stash -fips
[MQLab@mqlab 01_Introduction]$ runmqakm -cert -create -db "/var/mqm/qmgrs/MARS/s
sl/MARS.kdb" -pw clientpass -label ibmwebspheremqmars -dn "CN=MARS,OU=IMPACT,OU=
WMQSECLAB,OU=TEST,O=IBM,L=LAS VEGAS,ST=NV,C=USA" -expire 365 -fips -sigalg sha1
[MQLab@mqlab 01_Introduction]$
[MQLab@mqlab 01_Introduction]$ runmqakm -cert -list -db "/var/mqm/qmgrs/MARS/ssl
/MARS.kdb" -pw clientpass -fips
Certificates found:
* default, - has private key, ! trusted
-!      ibmwebspheremqmars
!       Thawte Personal Premium CA
!       Thawte Personal Freemail CA
!       Thawte Personal Basic CA
!       Thawte Premium Server CA
!       Thawte Server CA
!       VeriSign Class 3 Secure Server CA
!       VeriSign International Server CA - Class 3
!       VeriSign Class 4 Public Primary Certification Authority - G3
!       VeriSign Class 3 Public Primary Certification Authority - G5
!       VeriSign Class 3 Public Primary Certification Authority - G3
!       VeriSign Class 2 Public Primary Certification Authority - G3
!       VeriSign Class 1 Public Primary Certification Authority - G3
!       VeriSign Class 4 Public Primary Certification Authority - G2
!       VeriSign Class 3 Public Primary Certification Authority - G2
!       VeriSign Class 2 Public Primary Certification Authority - G2
!       VeriSign Class 1 Public Primary Certification Authority - G2
!       VeriSign Class 3 Public Primary Certification Authority
!       VeriSign Class 2 Public Primary Certification Authority
!       VeriSign Class 1 Public Primary Certification Authority
!       Entrust.net Global Secure Server Certification Authority
!       Entrust.net Global Client Certification Authority
!       Entrust.net Client Certification Authority
!       Entrust.net Certification Authority (2048)
!       Entrust.net Secure Server Certification Authority
[MQLab@mqlab 01_Introduction]$ █
```

Note that the key database is created pre-populated with a number of Certificate Authority root certificates. In practice you would immediately delete all of the ones that are not required for your implementation. For purposes of the lab we will skip this step in the interest of time.

Tip: The lab scripts for this module delete the default signer certificates when creating the keystore and provide a good example of how this could be done for a real-world deployment.

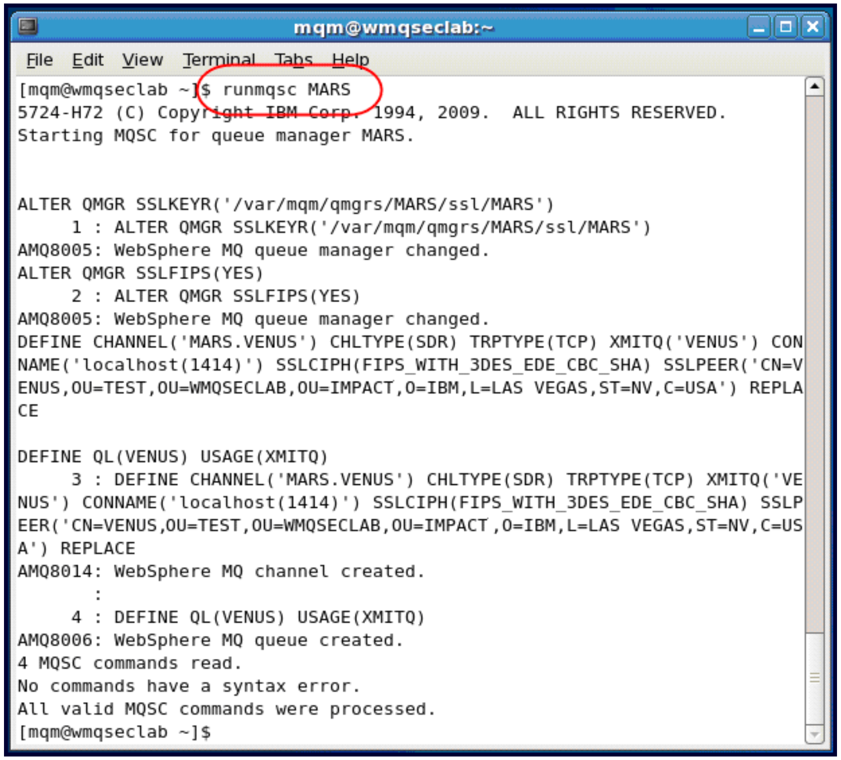
17) ____ Continue cutting runmqakm commands from the browser and pasting into the terminal window. Take a moment to read and understand the commands. In order for each queue manager to authenticate the other one, both need their own private key and both need the public key of the other one. The steps involved are:

1. Build the keystore files and generate the queue manager certificates.
2. Export the public keys to certificate files.
3. Exchange the public keys.
4. The public keys are then imported into the keystores of the corresponding queue managers.



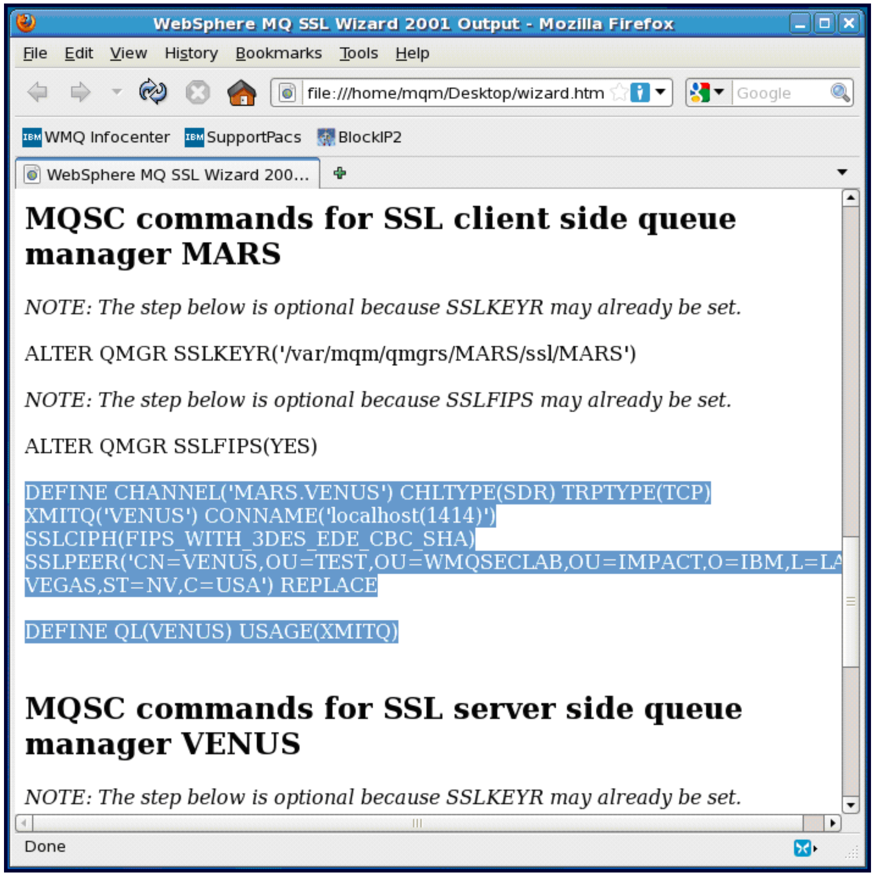
MQLab@mqlab:~/scripts/01_Introduction
File Edit View Terminal Tabs Help

```
[MQLab@mqlab 01_Introduction]$ runmqakm -cert -extract -db "/var/mqm/qmgrs/MARS/ssl/MA
RS.kdb" -pw clientpass -label ibmwebspheremqmars -target MARS.crt -format ascii -fips
[MQLab@mqlab 01_Introduction]$
[MQLab@mqlab 01_Introduction]$ runmqakm -cert -add -db "/var/mqm/qmgrs/VENUS/ssl/VENUS
.kdb" -pw serverpass -label ibmwebspheremqmars -file MARS.crt -format ascii -fips
[MQLab@mqlab 01_Introduction]$
[MQLab@mqlab 01_Introduction]$ runmqakm -cert -list -db "/var/mqm/qmgrs/VENUS/ssl/VENU
S.kdb" -pw serverpass -fips
Certificates found:
* default, - has private key, ! trusted
!       ibmwebspheremqmars
-!      ibmwebspheremqvenus
!       Thawte Personal Premium CA
!       Thawte Personal Freemail CA
!       Thawte Personal Basic CA
!       Thawte Premium Server CA
!       Thawte Server CA
!       VeriSign Class 3 Secure Server CA
!       VeriSign International Server CA - Class 3
!       VeriSign Class 4 Public Primary Certification Authority - G3
!       VeriSign Class 3 Public Primary Certification Authority - G5
!       VeriSign Class 3 Public Primary Certification Authority - G3
!       VeriSign Class 2 Public Primary Certification Authority - G3
!       VeriSign Class 1 Public Primary Certification Authority - G3
!       VeriSign Class 4 Public Primary Certification Authority - G2
!       VeriSign Class 3 Public Primary Certification Authority - G2
!       VeriSign Class 2 Public Primary Certification Authority - G2
!       VeriSign Class 1 Public Primary Certification Authority - G2
!       VeriSign Class 3 Public Primary Certification Authority
!       VeriSign Class 2 Public Primary Certification Authority
!       VeriSign Class 1 Public Primary Certification Authority
!       Entrust.net Global Secure Server Certification Authority
!       Entrust.net Global Client Certification Authority
!       Entrust.net Client Certification Authority
!       Entrust.net Certification Authority (2048)
!       Entrust.net Secure Server Certification Authority
[MQLab@mqlab 01_Introduction]$
```



```
[mqm@wmqseclab ~]$ runmqsc MARS
5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager MARS.


ALTER QMGR SSLKEYR('/var/mqm/qmgrs/MARS/ssl/MARS')
     1 : ALTER QMGR SSLKEYR('/var/mqm/qmgrs/MARS/ssl/MARS')
AMQ8005: WebSphere MQ queue manager changed.
ALTER QMGR SSLFIPS(YES)
     2 : ALTER QMGR SSLFIPS(YES)
AMQ8005: WebSphere MQ queue manager changed.
DEFINE CHANNEL('MARS.VENUS') CHLTYPE(SDR) TRPTYPE(TCP) XMITQ('VENUS') CON
NAME('localhost(1414)') SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA) SSLPEER('CN=V
ENUS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS VEGAS,ST=NV,C=USA') REPLA
CE

DEFINE QL(VENUS) USAGE(XMITQ)
     3 : DEFINE CHANNEL('MARS.VENUS') CHLTYPE(SDR) TRPTYPE(TCP) XMITQ('VE
NUS') CONNAME('localhost(1414)') SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA) SSLP
EER('CN=VENUS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS VEGAS,ST=NV,C=US
A') REPLACE
AMQ8014: WebSphere MQ channel created.
       :
     4 : DEFINE QL(VENUS) USAGE(XMITQ)
AMQ8006: WebSphere MQ queue created.
4 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
[mqm@wmqseclab ~]$
```

18) ____ Next the channels and queues must be defined. Start a runmqsc session for the MARS queue manager and paste the commands in one at a time. Type end to exit the runmqsc interpreter when done.


WebSphere MQ SSL Wizard 2001 Output - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///home/mqm/Desktop/wizard.htm
Google
WMQ Infocenter
SupportPacs
BlockIP2
WebSphere MQ SSL Wizard 200...
Done

MQSC commands for SSL client side queue manager MARS

NOTE: The step below is optional because SSLKEYR may already be set.

ALTER QMGR SSLKEYR('/var/mqm/qmgrs/MARS/ssl/MARS')

NOTE: The step below is optional because SSLFIPS may already be set.

ALTER QMGR SSLFIPS(YES)

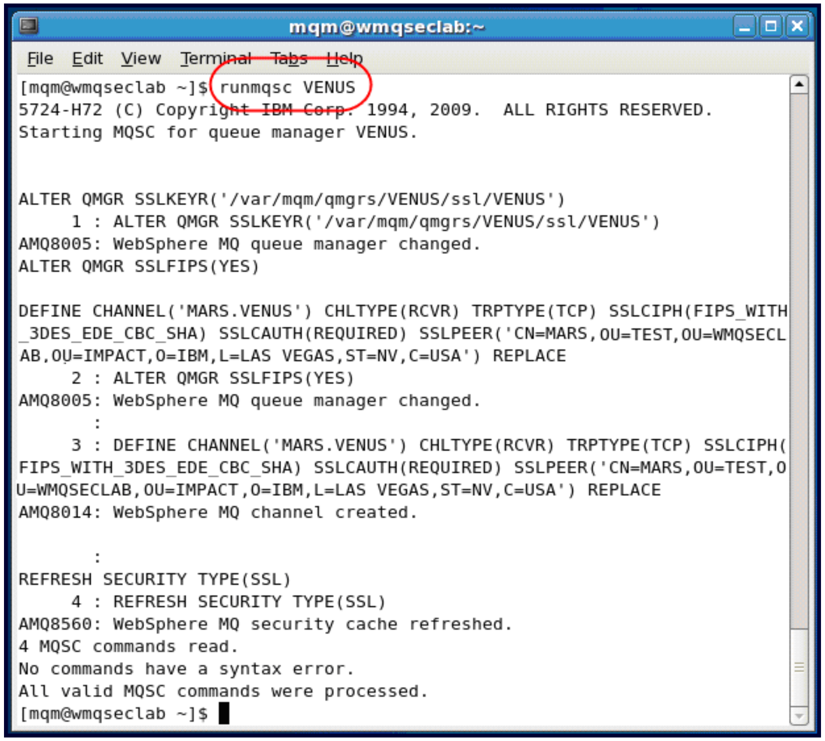
DEFINE CHANNEL('MARS.VENUS') CHLTYPE(SDR) TRPTYPE(TCP) XMITQ('VENUS') CONNAME('localhost(1414)') SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA) SSLPEER('CN=VENUS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LA
VEGAS,ST=NV,C=USA') REPLACE

DEFINE QL(VENUS) USAGE(XMITQ)

MQSC commands for SSL server side queue manager VENUS

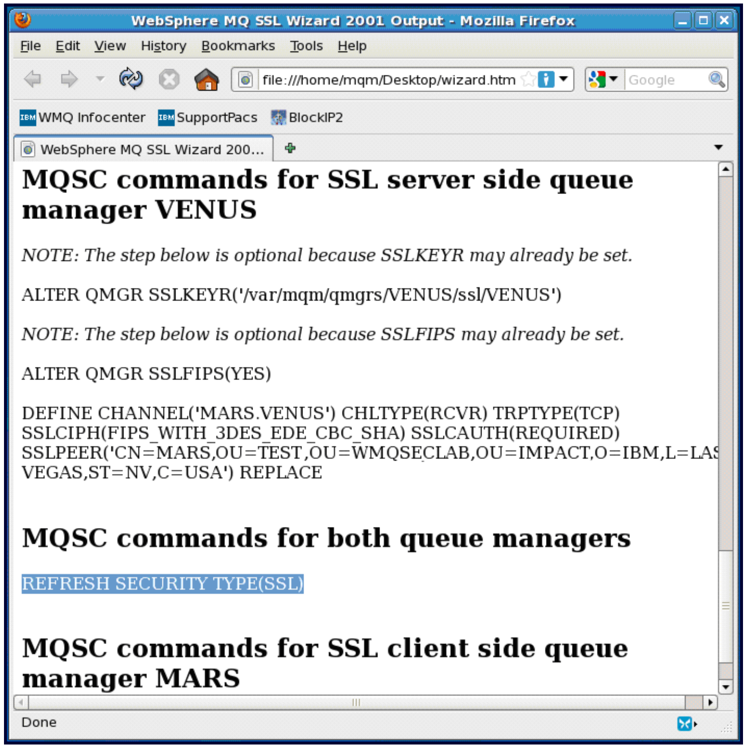
NOTE: The step below is optional because SSLKEYR may already be set.

Important! Although the SSL Wizard lists some steps as optional, they must be executed for the lab to work. When performing maintenance on a live queue manager where SSL has already been configured, these steps will have already been executed. But here in the lab where we are setting up SSL keystores for the first time, the queue manager alterations are required.



```
[mqm@wmqseclab ~]$ runmqsc VENUS
5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager VENUS.


ALTER QMGR SSLKEYR('/var/mqm/qmgrs/VENUS/ssl/VENUS')
     1 : ALTER QMGR SSLKEYR('/var/mqm/qmgrs/VENUS/ssl/VENUS')
AMQ8005: WebSphere MQ queue manager changed.
ALTER QMGR SSLFIPS(YES)

DEFINE CHANNEL('MARS.VENUS') CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(FIPS_WITH
_3DES_EDE_CBC_SHA) SSLCAUTH(REQUIRED) SSLPEER('CN=MARS,OU=TEST,OU=WMQSECL
AB,OU=IMPACT,O=IBM,L=LAS VEGAS,ST=NV,C=USA') REPLACE
     2 : ALTER QMGR SSLFIPS(YES)
AMQ8005: WebSphere MQ queue manager changed.
       :
     3 : DEFINE CHANNEL('MARS.VENUS') CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(
FIPS_WITH_3DES_EDE_CBC_SHA) SSLCAUTH(REQUIRED) SSLPEER('CN=MARS,OU=TEST,O
U=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS VEGAS,ST=NV,C=USA') REPLACE
AMQ8014: WebSphere MQ channel created.

       :
REFRESH SECURITY TYPE(SSL)
     4 : REFRESH SECURITY TYPE(SSL)
AMQ8560: WebSphere MQ security cache refreshed.
4 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
[mqm@wmqseclab ~]$
```

19) ____ Repeat this step on the VENUS queue manager using the commands tailored specifically for it.


WebSphere MQ SSL Wizard 2001 Output - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///home/mqm/Desktop/wizard.htm
Google
WMQ Infocenter
SupportPacs
BlockIP2
WebSphere MQ SSL Wizard 200...
Done

MQSC commands for SSL server side queue manager VENUS

NOTE: The step below is optional because SSLKEYR may already be set.

ALTER QMGR SSLKEYR('/var/mqm/qmgrs/VENUS/ssl/VENUS')

NOTE: The step below is optional because SSLFIPS may already be set.

ALTER QMGR SSLFIPS(YES)

DEFINE CHANNEL('MARS.VENUS') CHLTYPE(RCVR) TRPTYPE(TCP)
SSLCIPH(FIPS_WITH_3DES_EDE_CBC_SHA) SSLCAUTH(REQUIRED)
SSLPEER('CN=MARS,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,L=LAS
VEGAS,ST=NV,C=USA') REPLACE

MQSC commands for both queue managers

REFRESH SECURITY TYPE(SSL)

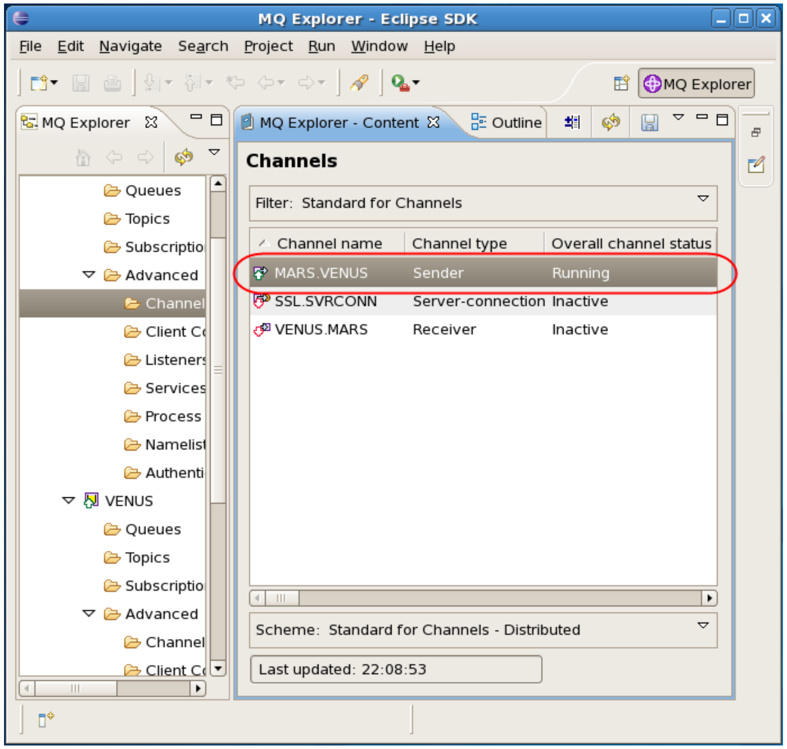
MQSC commands for SSL client side queue manager MARS

20) ____ Run the REFRESH SECURITY TYPE(SSL) command on both queue managers. This causes them to flush the cache and reload the certificates from the keystore.

Tip: We need to do the next step because we are not logged in as the mqm user id. Recommend that keystores are created using the mqm user id to restrict access to a minimum.

21) ____ The channels that will use the keystore, need access to it. Type the following commands:-

```
chmod g+r /var/mqm/qmgrs/MARS/ssl/*
chmod g+r /var/mqm/qmgrs/VENUS/ssl/*
```



Channels

Filter: Standard for Channels

Channel name	Channel type	Overall channel status
MARS.VENUS	Sender	Running
SSL.SVRCONN	Server-connection	Inactive
VENUS.MARS	Receiver	Inactive

Scheme: Standard for Channels - Distributed

Last updated: 22:08:53

22) ____ Right-click the channel and select Start. The channel should now go into Running state.

2.4 Summary

In this module you configured two queue managers with certificates, exchanged the public keys and then set up SSL channels between the queue managers.

In SSL terms, the thing making the connection request is considered the client and the thing responding to the request is called the server. The server always presents a certificate on the connection request. Because you specified SSLCAUTH(REQUIRED) on the RCVR channels, the client (in this case the queue manager initiating the connection) is required to present its certificate as well. When both sides are required to present certificates, this is known as mutual authentication.

Although the channels now authenticate one another, they still allow administrative access. In order to address this, it is necessary to link authentication to authorization in a meaningful way. In a later module we will see how this is accomplished with the use of channel authentication records to set the channel's runtime MCAUSER.

But first, we will see how to authenticate interactive sessions from human users. This will provide a restricted access path for administrators as well the means to apply granular security to non-administrators. In the next module you will configure SSL connections between the queue managers and WebSphere MQ Explorer.

3. Use iKeyman GUI to configure remote administrative access

The VMWare image will automatically log you on as MQLab (a user in the mqm group).

3.1 Module objectives

The objective of this module is to provision an authenticated access path for WebSphere MQ administrators. When access to the default and application channels is restricted, the administrators will require some other way to access the queue manager. The administrative channel should meet the following criteria:

- The channel is authenticated. Mere assertion of an identity must not be sufficient to obtain access.
- In addition to allowing administrators, the channel must demonstrate that connection requests from non-administrators are rejected.

Provisioning secure access for administrators would normally be among the first steps taken to perform base hardening of a WebSphere MQ Queue Manager.

3.2 Background

Authentication is the verification of an identity – *who are you?*.
Authorization is the enforcement of access policies – *what actions are you allowed to perform?*

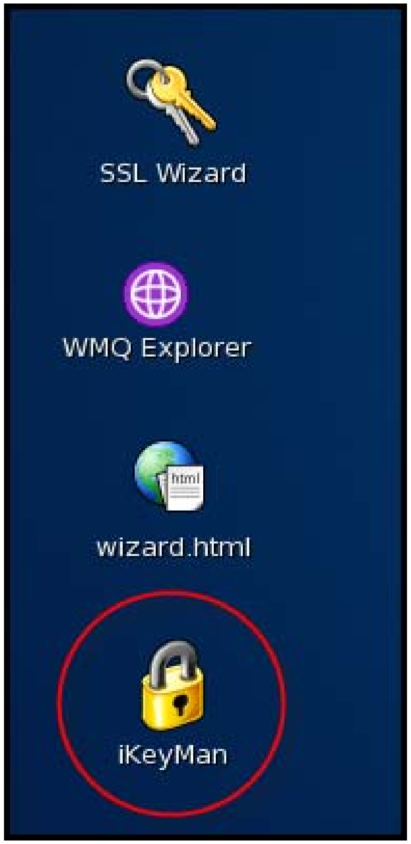
Authorization without authentication – enforcing access controls without verifying the ID that is presented – gives a false sense of security and therefore may be worse than having no authorization controls at all.

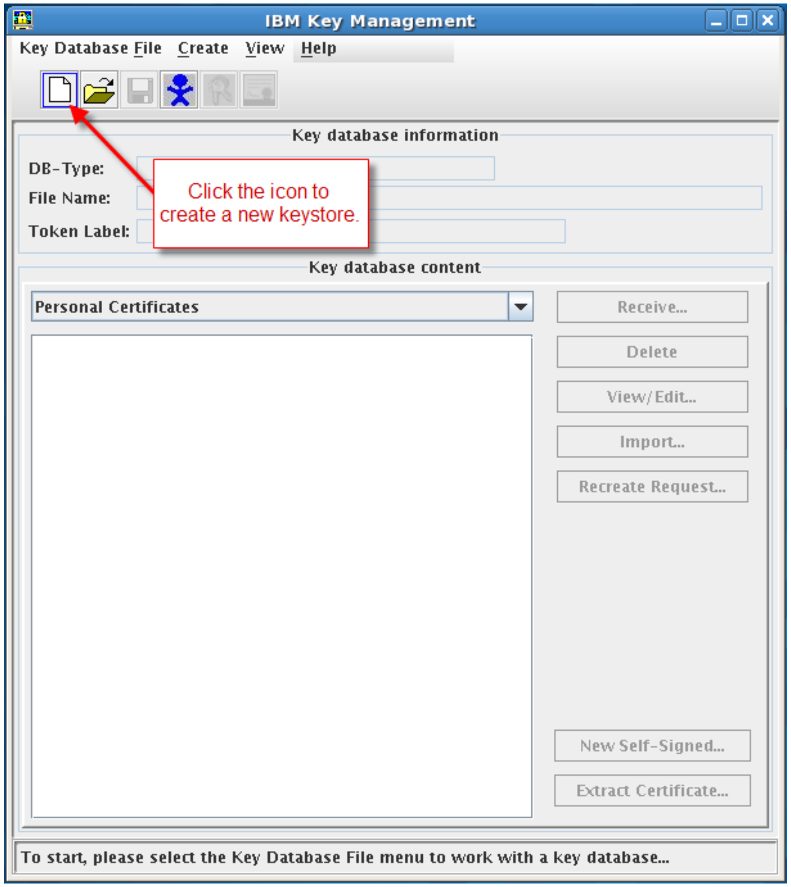
Authentication without authorization – going to the trouble of verifying identities but granting blanket administrative access to all users – is only useful in the rare case that all users are truly administrative.

These statements apply to any information processing system. They are not unique to WebSphere MQ. However, as you will see in this module, both of these situations can be present in a queue manager. The module will demonstrate first how to authenticate a remote connection using SSL certificates, then how to tie the authenticated identity to a specific user ID so that authorization can be enforced in a meaningful way.

The previous module used the SupportPac MO04 SSL Wizard to generate the commands to build a keystore for the queue manager in KDB format. WebSphere MQ Explorer is a Java application and therefore uses the Java standard format for keystores rather than the KDB. This module will use the iKeyMan GUI to build the Java Keystore (.jks file) and generate the personal certificate for the administrator. Next, the queue manager's and administrator's public keys are exchanged and added to each other's keystores.

3.3 Create the keystore



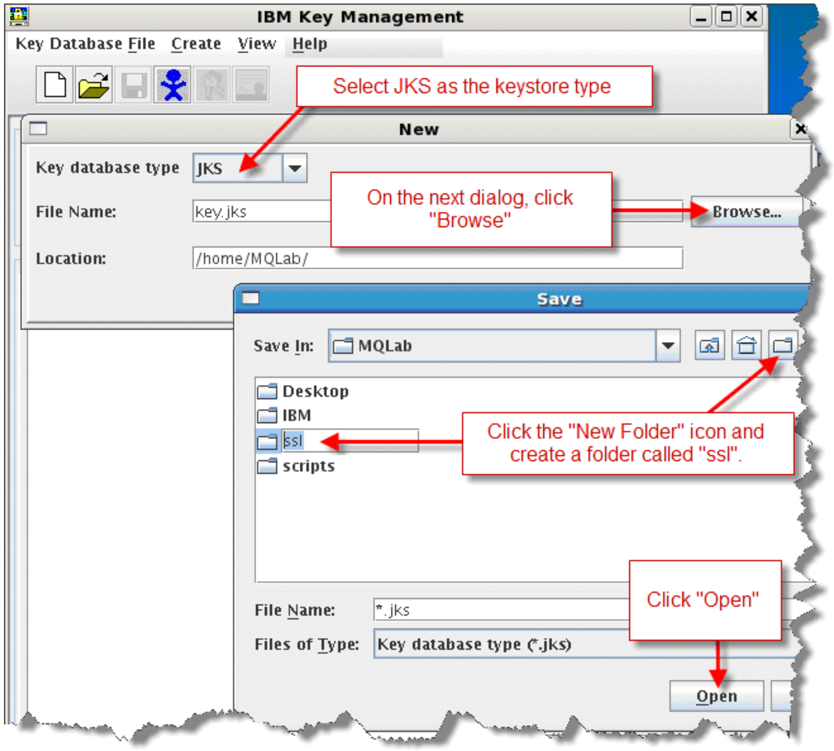


Click the icon to create a new keystore.

To start, please select the Key Database File menu to work with a key database...

01) ____ Double-click the lock icon on the desktop to start the iKeyMan GUI.

02) ____ When iKeyMan starts, click the "New" icon.



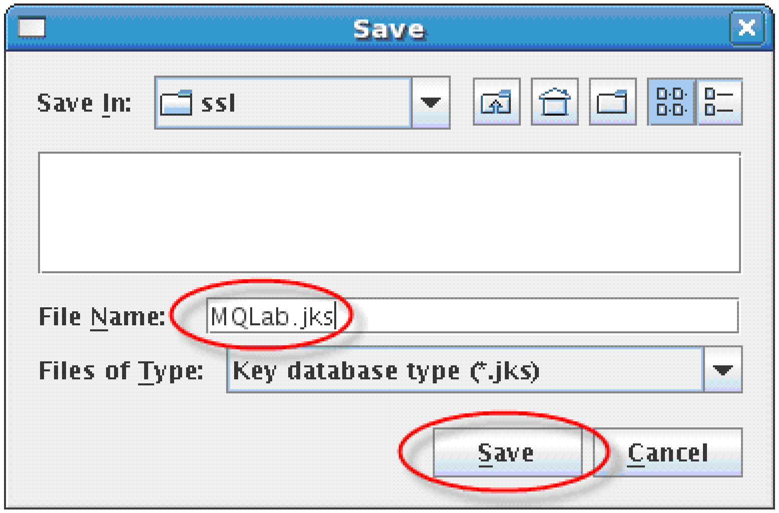
Select JKS as the keystore type

On the next dialog, click "Browse"

Click the "New Folder" icon and create a folder called "ssl".

Click "Open"

03) ____ Make sure that the Key Database Type is set to JKS before continuing. Select Browse to open the file chooser dialog. Next make sure the chooser is set to the MQLab home space. If the ssl directory does not already exist in /home/MQLab then create it. Select the ssl directory and click "Open".



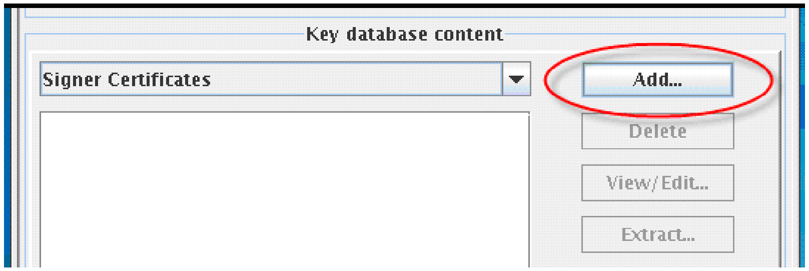
04) ____ The chooser dialog will enter the ssl directory and the file name will change to *.jks. Type in the file name MQLab.`jks` and click the "Save" button. Note that there is nothing special about the name of the Java keystore. Any valid file name would suffice.

Enter 'passw0rd' when prompted for a password.

Tip: During the lab, several keystores will be created with default file permissions. In real-world deployments, a Java keystore would be placed into a directory accessible only by its owner. Normally 'group' and 'other' read and write permissions are denied on keystores. For example, after creating the keystore above as the MQLab user, the administrator would then issue the following commands:

```
chmod 700 /home/MQLab/ssl
chmod 500 /home/MQLab/ssl/*
```

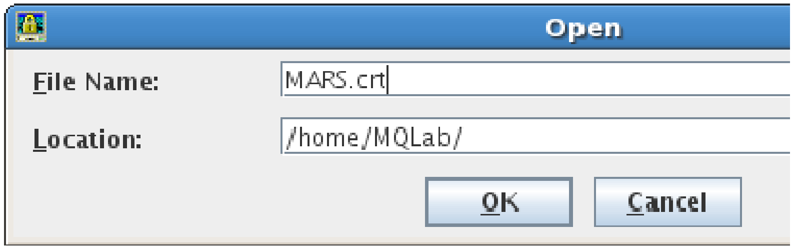
3.4 Loading QMgr keys into the user's keystore



Signer Certificates

05) ____ In order to connect to the queue managers, it is necessary to first load their keys into the keystore. Choose Signer Certificates, and click the Add button.

Tip: Many systems and applications support the use of separate databases for private keys and public keys. In this context, the database for public keys is called a Trust Store. This arrangement facilitates the sharing of a single Trust Store among several users or applications, each of which in turn stores its own private keys in a dedicated keystore that is not accessible to anyone else. The lab exercises will use a single keystore file as both the trust store and the key store.



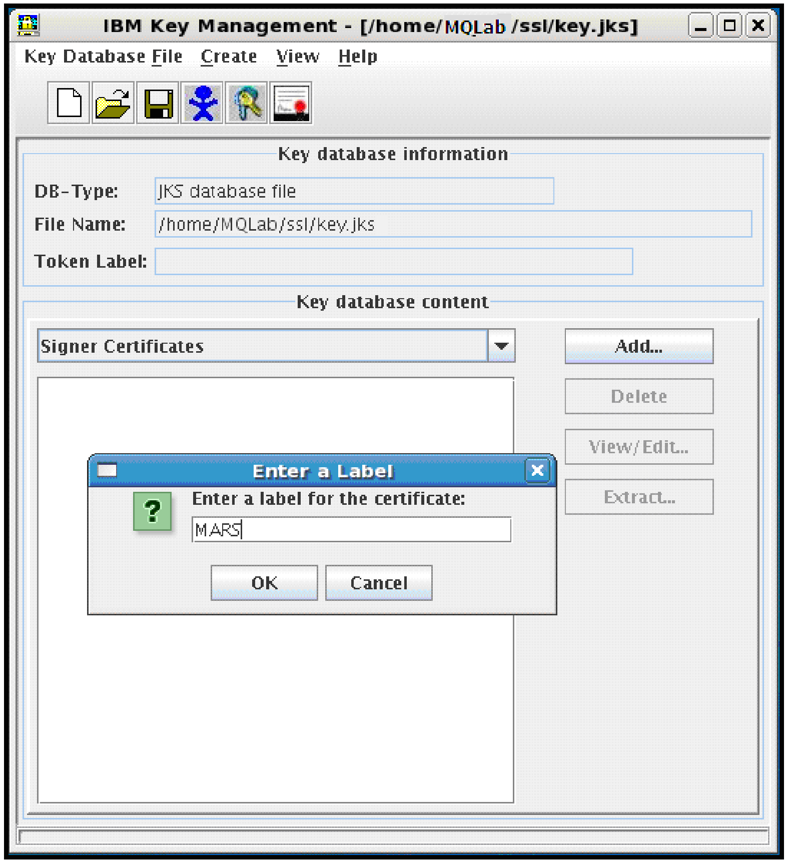
File Name: MARS.crt

Location: /home/MQLab/

06) ____ Enter `MARS.crt` in the Certificate File Name field. Enter `/home/MQLab` in the Location field. Click OK.

Note: The `*`.crt files will have been saved in the current working directory that was set when you ran the extract commands in the previous module. Usually this is /home/MQLab but if you had changed to a different directory you might need to find the files or re-run the commands from a known directory. If the files are not in /home/MQLab you can use the find command to locate them:

```
$ find /home -name '*.crt'
find: /home/virtuser: Permission denied
/home/MQLab/VENUS.crt
/home/MQLab/MARS.crt
$
```



IBM Key Management - [/home/MQLab /ssl/key.jks]

Key database information

DB-Type: JKS database file

File Name: /home/MQLab/ssl/key.jks

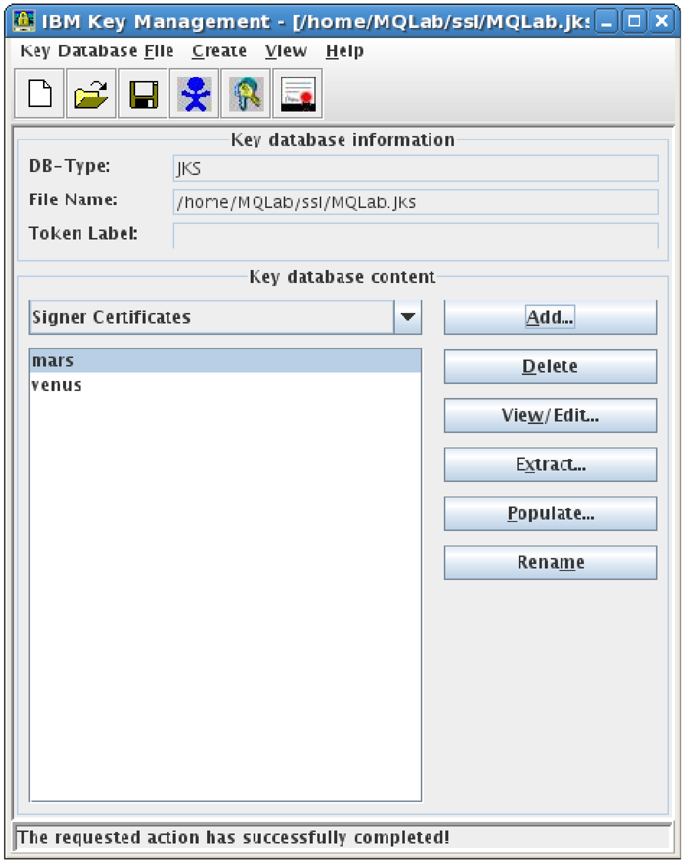
Token Label:

Key database content

Signer Certificates

Enter a Label

Enter a label for the certificate: MARS



IBM Key Management - [/home/MQLab/ssl/MQLab.jk:

Key database information

DB-Type: JKS

File Name: /home/MQLab/ssl/MQLab.jks

Token Label:

Key database content

Signer Certificates

mars
venus

The requested action has successfully completed!

07) ____ You are prompted for a certificate label. The labels for signer certificates are for your convenience in managing the keystore and you can select any meaningful value. We will be using the names of queue managers or accounts in the lab. In practice you might include other details in the label, such as the certificate expiration date. Devising a naming convention and using it consistently can reduce the overhead involved in certificate management.

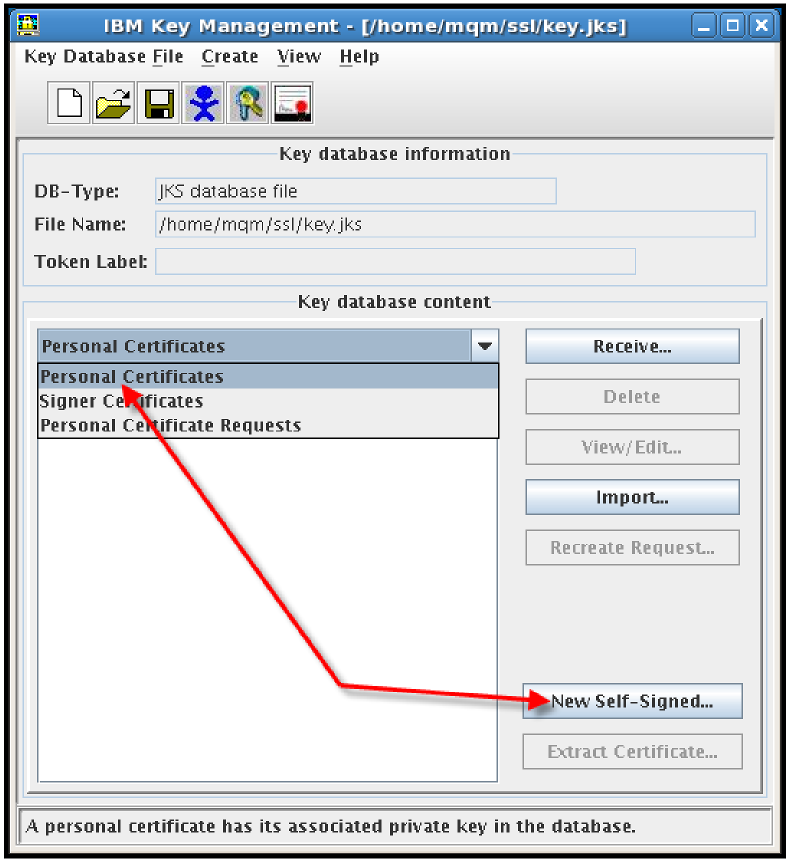
Note that file names are case-sensitive on UNIX systems.

08) ____ After clicking OK, the MARS queue manager's certificate will be visible.

09) ____ Add the certificate for the VENUS queue manager using the same procedure.

Make sure the certificates for both queue managers are visible before continuing.

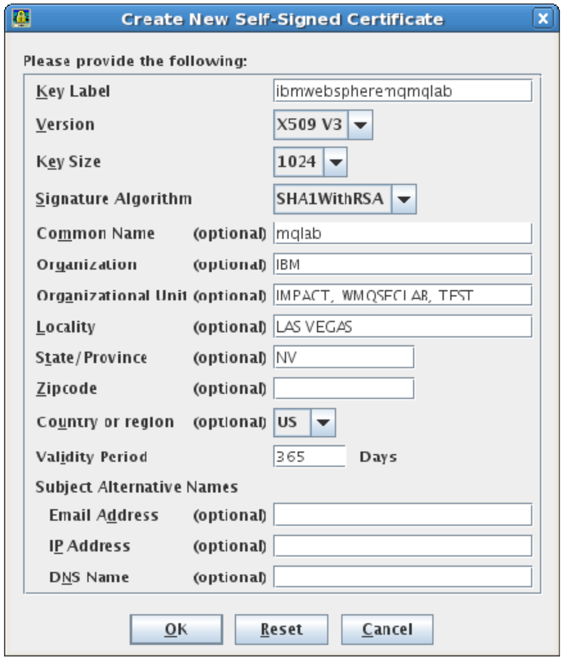
3.5 Generate your personal certificate



DB-Type:	JKS database file
File Name:	/home/mqm/ssl/key.jks
Token Label:	

A personal certificate has its associated private key in the database.

10) ____ It is now time to generate your personal certificate. Click the drop-down and select Personal Certificates. The Key Database content controls will change and a button for New Self Signed certificate will appear. Click it.



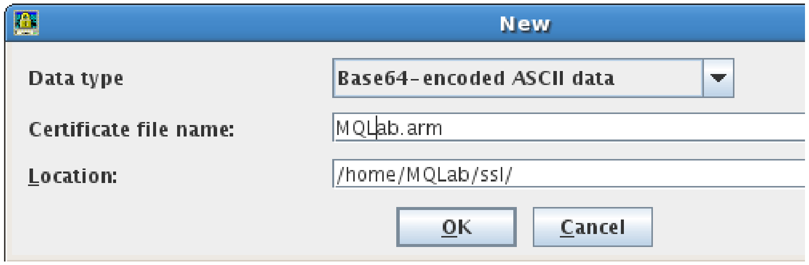
Please provide the following:

Key Label		ibmwebspheremqmqlab
Version		X509 V3
Key Size		1024
Signature Algorithm		SHA1WithRSA
Common Name	(optional)	mqlab
Organization	(optional)	IBM
Organizational Unit	(optional)	IMPACT, WMQSECLAB, TEST
Locality	(optional)	LAS VEGAS
State/Province	(optional)	NV
Zipcode	(optional)	
Country or region	(optional)	US
Validity Period		365 Days
Subject Alternative Names		
Email Address	(optional)	
IP Address	(optional)	
DNS Name	(optional)	

11) ____ Enter the details of the new certificate. Although the key label is not significant for signer certificates, there are cases where it is for personal certificates. When the name is significant, the required format is the literal string ibmwebspheremq followed by the queue manager or account name. We will follow that convention for the lab exercises. Enter the name ibmwebspheremqmqlab for the label.

The Distinguished Name (DN) of the certificate is comprised of several fields. On this dialog they begin with Common Name and continue through Country. It is advisable to spend some time developing a DN naming convention. For example, the Organization Unit (OU) illustrated here includes a node for the environment. This facilitates enforcement of isolation between TEST, QA, PROD and so forth.

3.6 Extract the public key



Data type: Base64-encoded ASCII data

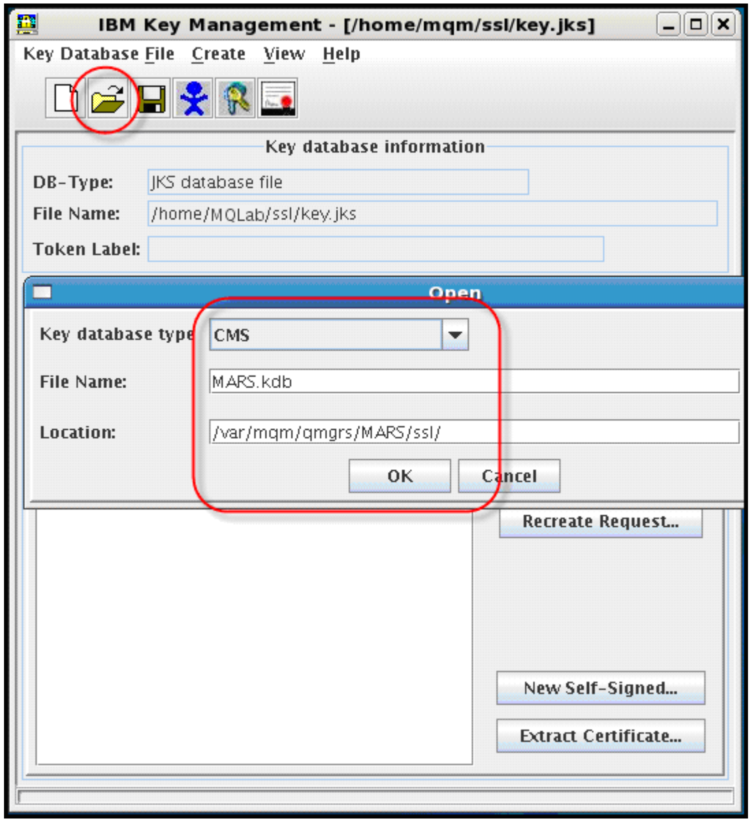
Certificate file name: MQLab.arm

Location: /home/MQLab/ssl/

12) ____ Click the Extract Certificate button.

On the Extract dialog, select Base 64-encoded ASCII data from the drop-down. Change the file name to MQLab.arm so that you will be able to identify it later. Enter `/home/MQLab/ssl` in the Location field and click the OK button.

3.7 Import the key into the QMgr's KDB



DB-Type: JKS database file

File Name: /home/MQLab/ssl/key.jks

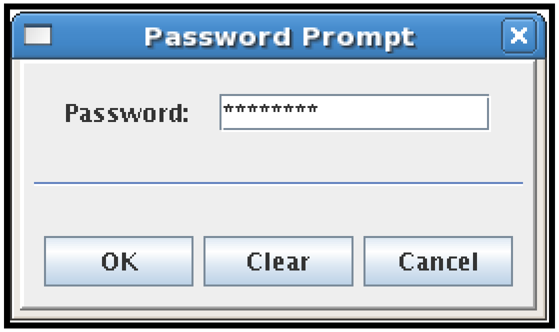
Token Label:

Key database type: CMS

File Name: MARS.kdb

Location: /var/mqm/qmgrs/MARS/ssl/

13) ____ The next step is to open the queue manager's kdb file and add the new public key to it. Click the Open icon at the top left of the iKeyman GUI. In the Open dialog, specify a database type of CMS, the file name `MARS.kdb` and a Location of `/var/mqm/qmgrs/MARS/ssl`. Click the OK button.

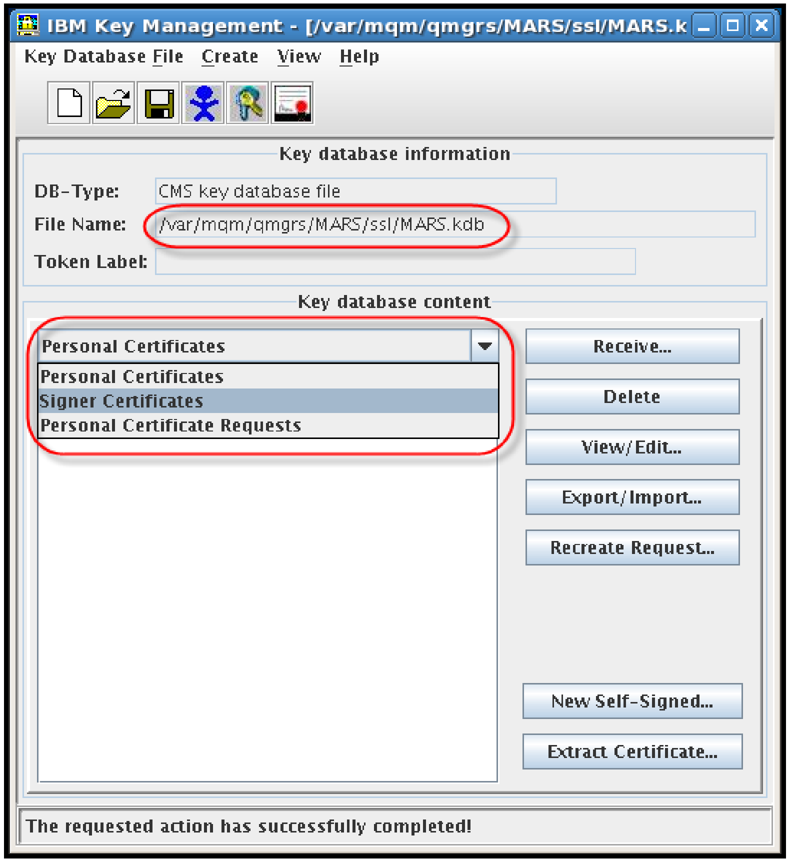


14) ____ Enter the password and click OK.

The password for the MARS KDB is clientpass.

The password for the VENUS KDB is serverpass.

Tip: Don't read too much into the passwords, they are generated by MO04 based on MARS queue manager being the client and VENUS queue manager being the server, but of course we may use them for other things too.



DB-Type: CMS key database file

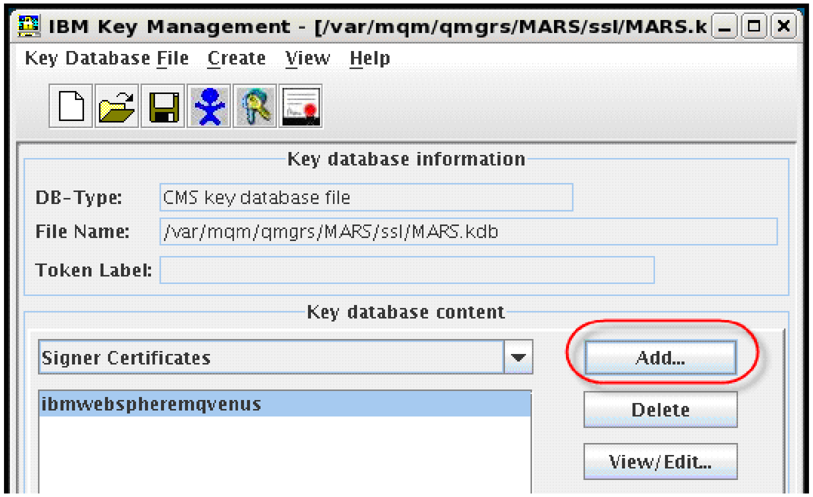
File Name: /var/mqm/qmgrs/MARS/ssl/MARS.kdb

Token Label:

Personal Certificates
- Personal Certificates
- Signer Certificates
- Personal Certificate Requests

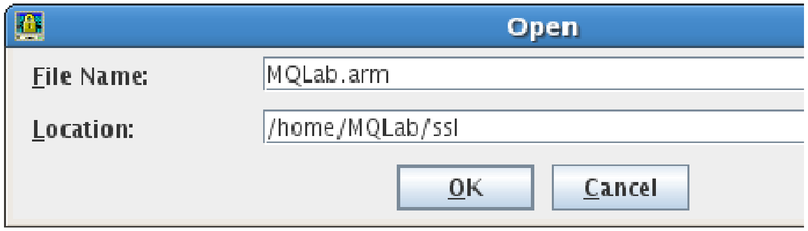
The requested action has successfully completed!

15) ____ The File Name field will change to indicate that the queue manager's KDB is now loaded into iKeyman. Select Signer Certificates from the Content drop-down.



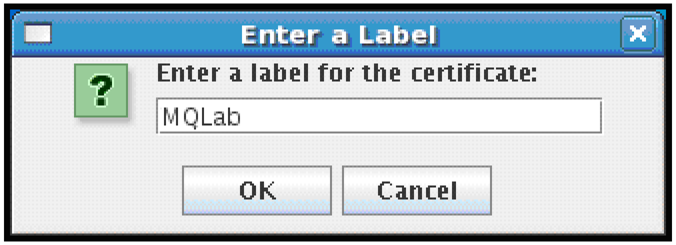
DB-Type: CMS key database file
File Name: /var/mqm/qmgrs/MARS/ssl/MARS.kdb
Token Label:

Signer Certificates
ibmwebspheremqvenus



File Name: MQLab.arm
Location: /home/MQLab/ssl

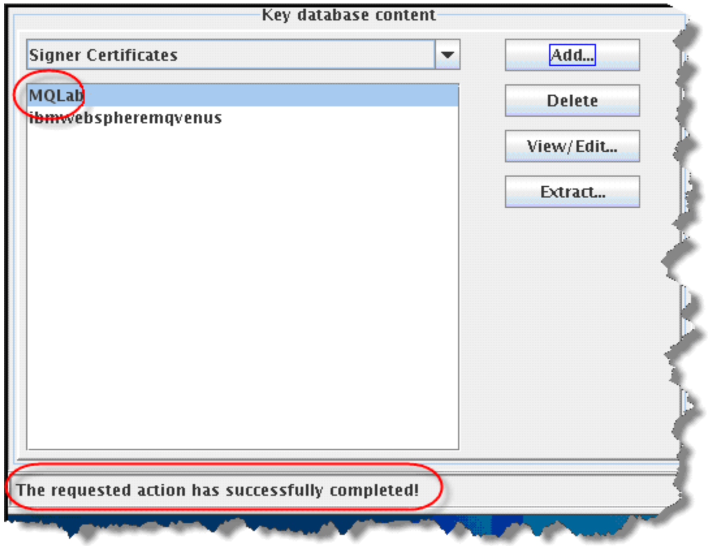
16) ____ The signer certificate in the keystore at this point is the one previously added and belonging to the other queue manager. Click the Add button. In the Open dialog, enter a file name of `MQLab.arm` and a Location of `/home/MQLab/ssl` then click OK.



Enter a label for the certificate: MQLab

17) ____ Enter the label for the certificate. As noted earlier, there are no restrictions on the label of a public key, however using the account or queue manager name helps with key management.

The certificate will now be visible in the Signers section of the KDB and a success message is displayed in the status line.



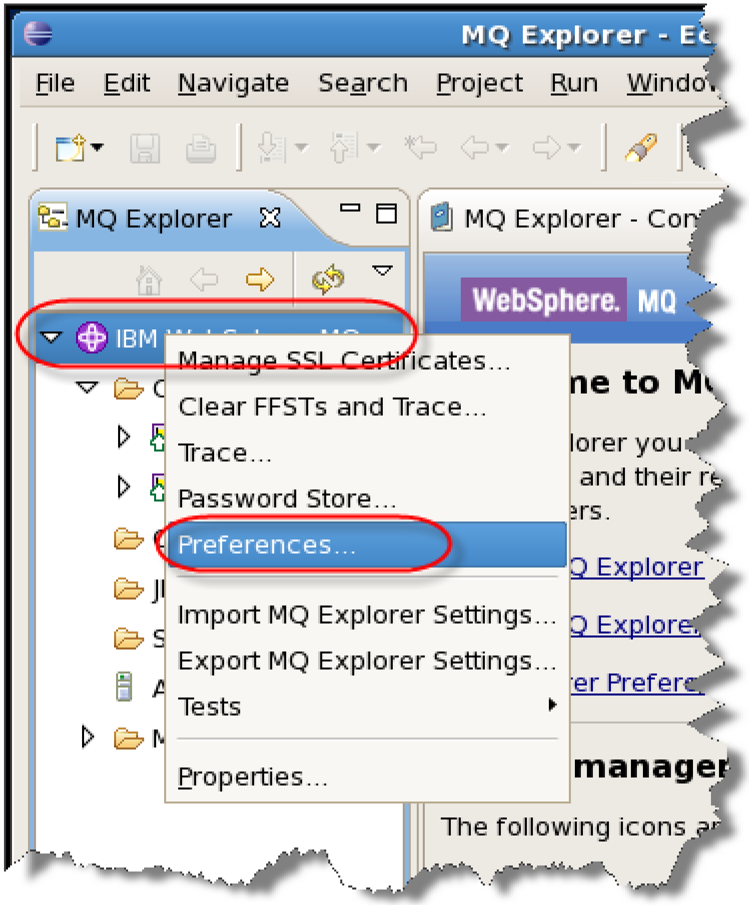
Signer Certificates
MQLab
ibmwebspheremqvenus

The requested action has successfully completed!

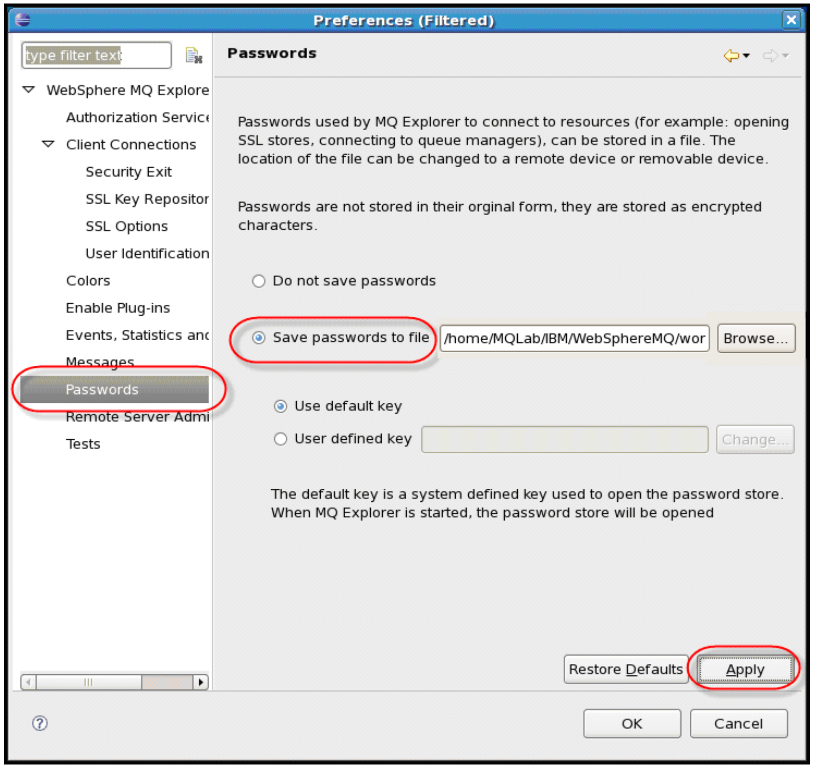
18) ____ Repeat steps 13 through 17 for the VENUS queue manager before continuing

19) ____ Issue REFRESH SECURITY TYPE(SSL) on both queue managers before continuing.

3.8 Configuring WebSphere MQ Explorer



21) ____ The lab will proceed much quicker if we enable WMQ Explorer's password store. Start the MQ Explorer if it is not already running. Right-click on IBM WebSphere MQ and select Preferences. (Consider any stored password policies your organization may have before enabling this feature on your own copy of WMQ Explorer.)



Passwords

Passwords used by MQ Explorer to connect to resources (for example: opening SSL stores, connecting to queue managers), can be stored in a file. The location of the file can be changed to a remote device or removable device.

Passwords are not stored in their orginal form, they are stored as encrypted characters.

○ Do not save passwords

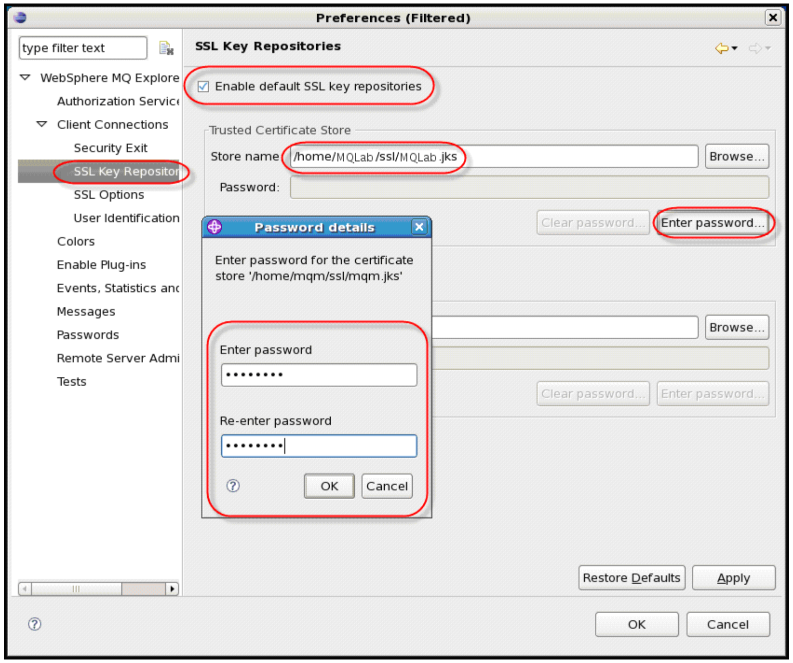
● Save passwords to file /home/MQLab/IBM/WebSphereMQ/wor Browse...

● Use default key

○ User defined key Change...

The default key is a system defined key used to open the password store. When MQ Explorer is started, the password store will be opened

22) ____ In the Preferences dialog, select the Passwords panel from the navigation tree on the left. Then select "Save passwords to file" and click Apply. The default file and key settings are sufficient.



☑ Enable default SSL key repositories

Store name: /home/MQLab/ssl/MQLab.jks

Enter password for the certificate store '/home/mqm/ssl/mqm.jks'

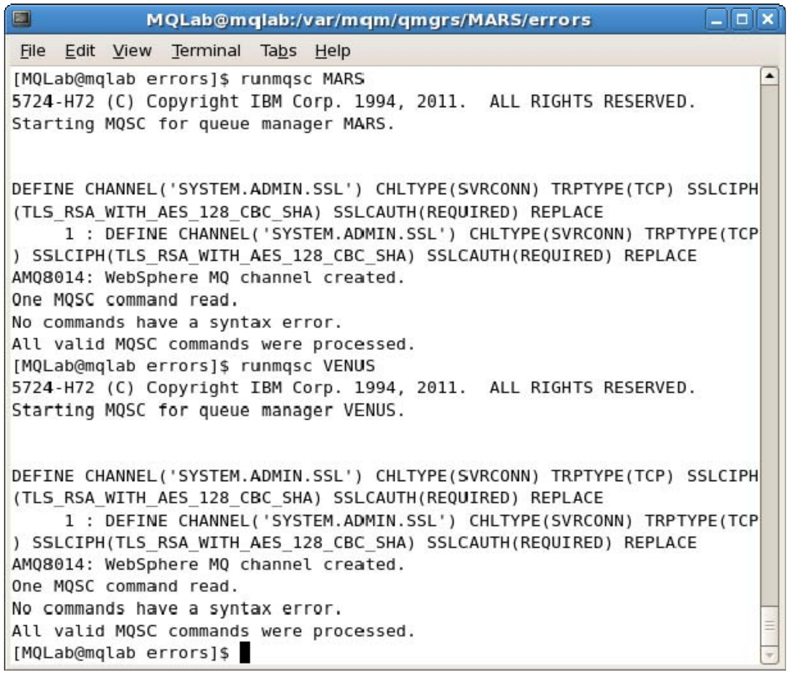
Enter password: ••••••••

Re-enter password: ••••••••

23) ____ Next, select "SSL Key Repositories" from the navigation tree. In the resulting panel, check the box to enable the default SSL key repositories. In the Store Name field enter `/home/MQLab/ssl/MQLab.jks` and click the Enter password button. Type the password (passw0rd) in each of the fields. When both passwords are entered and match, the OK button will enable. Click it.

24) ____ **Repeat Step 23 for the “Personal Certificate Store” section on the same dialog.**

25) ____ The administrator will require a secure channel. It is often convenient to leave SYSTEM.ADMIN.SVRCONN in place with restricted permissions for non-administrative users and create a new channel for administrators. That is the approach taken here. Right-click on the desktop and select Open Terminal to get a command-line prompt.


MQLab@mqlab:/var/mqm/qmgrs/MARS/errors
File Edit View Terminal Tabs Help

```
[MQLab@mqlab errors]$ runmqsc MARS
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager MARS.


DEFINE CHANNEL('SYSTEM.ADMIN.SSL') CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH
(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED) REPLACE
     1 : DEFINE CHANNEL('SYSTEM.ADMIN.SSL') CHLTYPE(SVRCONN) TRPTYPE(TCP
) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED) REPLACE
AMQ8014: WebSphere MQ channel created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
[MQLab@mqlab errors]$ runmqsc VENUS
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager VENUS.


DEFINE CHANNEL('SYSTEM.ADMIN.SSL') CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH
(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED) REPLACE
     1 : DEFINE CHANNEL('SYSTEM.ADMIN.SSL') CHLTYPE(SVRCONN) TRPTYPE(TCP
) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED) REPLACE
AMQ8014: WebSphere MQ channel created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
[MQLab@mqlab errors]$ 
```

26) ____ Type **`runmqsc MARS`** to start mqsc. Enter the following channel definition:

```
DEFINE CHANNEL('SYSTEM.ADMIN.SSL') +
CHLTYPE(SVRCONN) TRPTYPE(TCP) +
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) +
SSLCAUTH(REQUIRED) REPLACE
```

Type end to exit mqsc.

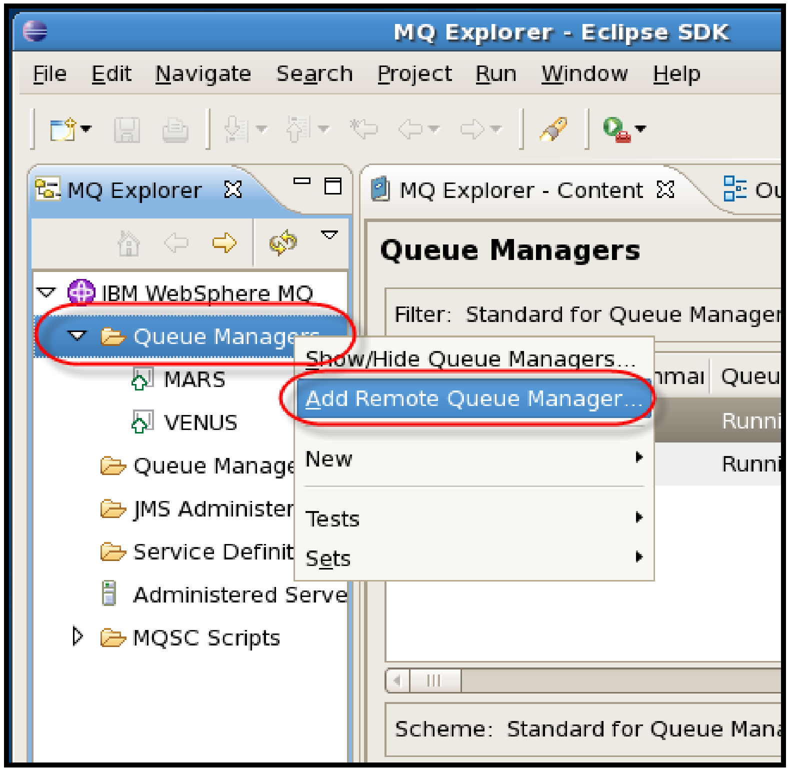
27) ____ Type **`runmqsc VENUS`** to start mqsc. Enter the following channel definition:

```
DEFINE CHANNEL('SYSTEM.ADMIN.SSL') +
CHLTYPE(SVRCONN) TRPTYPE(TCP) +
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) +
SSLCAUTH(REQUIRED) REPLACE
```

Type end to exit mqsc.

28) ____ In a later module we will illustrate channel authentication records, but for now will disable them until we get to that part of the lab. On each queue manager, start runmqsc and type the following command:

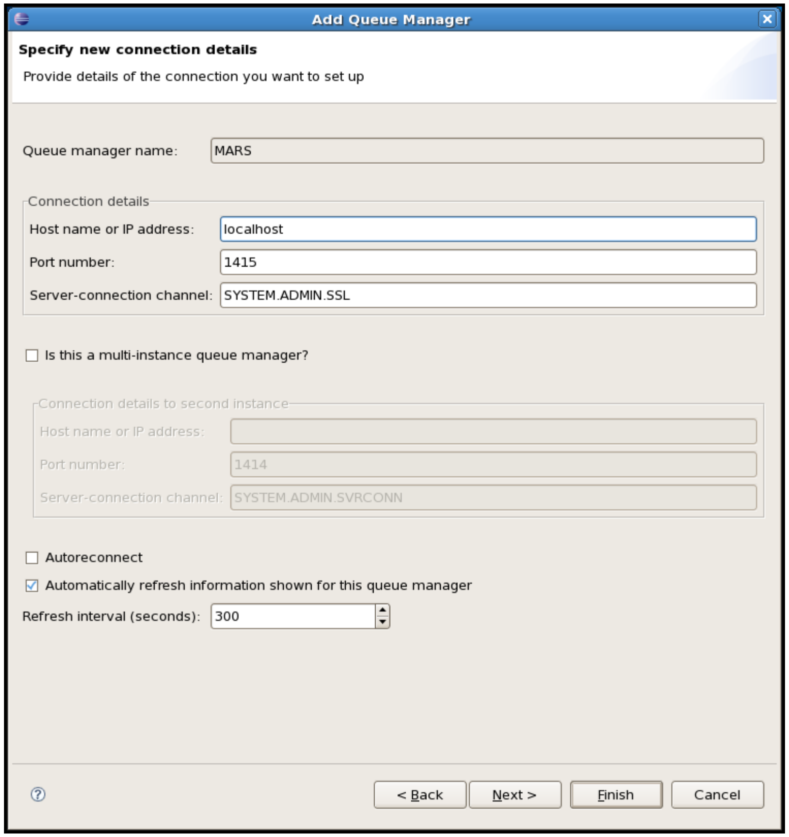
```
ALTER QMGR CHLAUTH(DISABLED)
```


MQ Explorer - Eclipse SDK
File Edit Navigate Search Project Run Window Help
MQ Explorer
MQ Explorer - Content
IBM WebSphere MQ
Queue Managers
MARS
VENUS
Queue Manage
JMS Administer
Service Definit
Administered Serve
MQSC Scripts
Queue Managers
Filter: Standard for Queue Manager
Show/Hide Queue Managers...
Add Remote Queue Manager...
New
Tests
Sets
Runni
Runni
Scheme: Standard for Queue Mana

29) ____ Now it's time to add the queue managers to Explorer.

Right-click on Queue Managers in the Explorer navigation tree and then select Add Remote Queue Manager.

30) ____ On the next dialog, type in the queue manager's name. Leave the selection as ‘Connect directly’ and click Next.


Add Queue Manager

Specify new connection details

Provide details of the connection you want to set up

Queue manager name: MARS

Connection details
Host name or IP address: localhost
Port number: 1415
Server-connection channel: SYSTEM.ADMIN.SSL

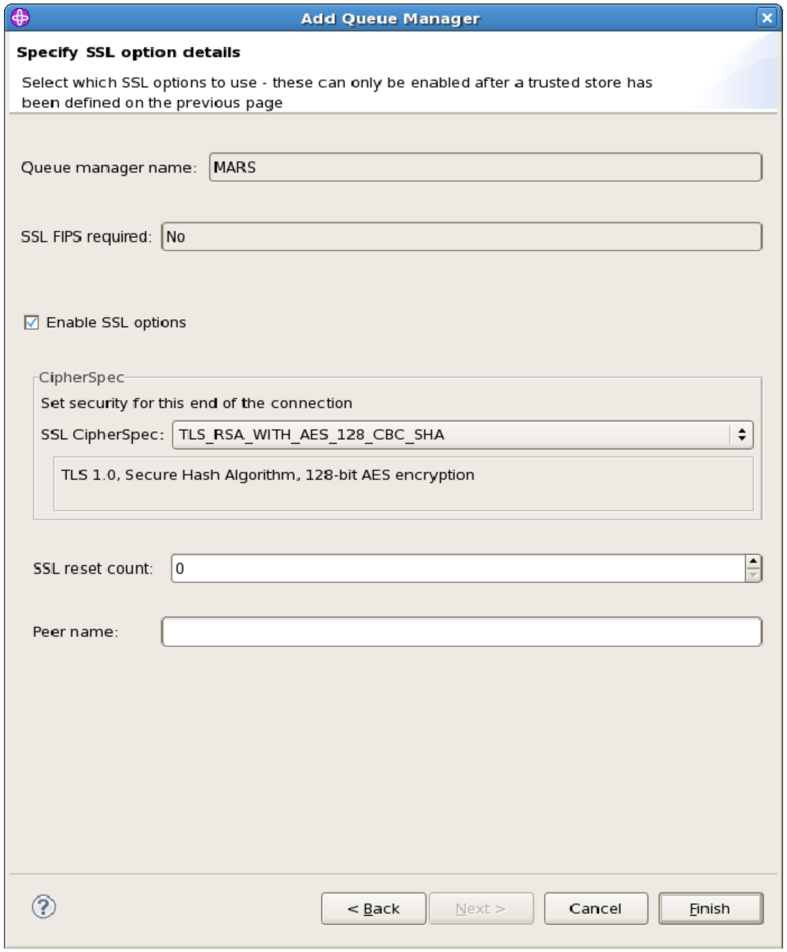
☐ Is this a multi-instance queue manager?

Connection details to second instance
Host name or IP address:
Port number: 1414
Server-connection channel: SYSTEM.ADMIN.SVRCONN

☐ Autoreconnect
☑ Automatically refresh information shown for this queue manager
Refresh interval (seconds): 300

31) ____ In the Connection Details dialog, enter `localhost` as the host name, the appropriate port (1414 for VENUS or 1415 for MARS) and `SYSTEM.ADMIN.SSL` as the channel name. Leave the remaining defaults and click Next.

32) ___ Click Next in the Security Exit and User Identification panels. Note that the fields on the Key Repository panel are pre-filled from the values previously entered in the Preferences. Check “Enabled SSL key repositories” and click Next to advance to the SSL Settings panel.



Specify SSL option details

Select which SSL options to use - these can only be enabled after a trusted store has been defined on the previous page

Queue manager name: MARS

SSL FIPS required: No

☑ Enable SSL options

CipherSpec

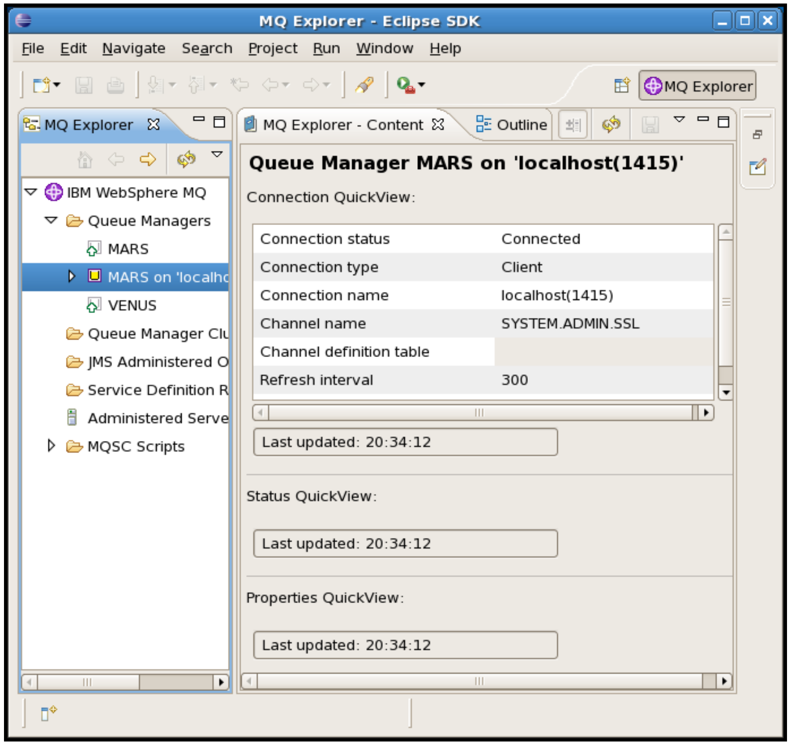
Set security for this end of the connection

SSL CipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA

TLS 1.0, Secure Hash Algorithm, 128-bit AES encryption

SSL reset count: 0

Peer name:



Queue Manager MARS on 'localhost(1415)'

Connection QuickView:

Connection status	Connected
Connection type	Client
Connection name	localhost(1415)
Channel name	SYSTEM.ADMIN.SSL
Channel definition table	
Refresh interval	300

Last updated: 20:34:12

Status QuickView:

Last updated: 20:34:12

Properties QuickView:

Last updated: 20:34:12

33) ____ On the SSL Details panel, select the checkbox to enable the SSL Options. Select TLS_RSA_WITH_AES_128_CBC_SHA from the CipherSpec drop-down. Leave the remaining fields at their defaults and click the Finish button.

34) ____ The Queue Managers list will now contain the new entry for MARS. Note that the new entry lists a host name next to the queue manager name. Clicking on it brings up the connection details in the content pane. Confirm that the connection is using the SSL channel.

35) ____ **Repeat steps 28 through 34 to add a connection for the VENUS queue manager on port 1414 before continuing.**

3.9 Module 3 Summary

This module demonstrated one way to configure administrative access from WebSphere MQ Explorer to queue managers. A dedicated administrative channel was created for this connection. Personal certificates were generated for the user and both of the queue managers and the public keys were exchanged. When self-signed certificates are used (as demonstrated here in this lab), the queue manager's public key must be in the user's Trust Store in order to make the connection. When CA-signed certificates are used it is not necessary for the queue manager’s public key to exist in the user’s trust store, however it is necessary that the CA root certificate and any intermediate signer certificates exist in the user’s trust store. This is because the SSL protocol always requires at least the server to authenticate.

When the client is also required to present a certificate, the configuration is said to be “mutually authenticated.” Setting the SVRCONN channel's SSLCAUTH attribute to REQUIRED results in a mutually authenticated connection. The queue manager will not allow the connection unless WMQ Explorer presents the user's certificate and the queue manager is able to validate it. The connection is only successful when the certificates of both parties are cross-validated and trusted.

This is one of the basic steps in hardening a queue manager, however additional configuration is required. Currently, there is no correlation between certificates in the Trust Store and the channels they are intended to be used with. In other words, the user's certificate that was configured in Module 3 to connect to SYSTEM.ADMIN.SSL would also validate and successfully connect to the RCVR channels on MARS and VENUS. The next module will configure the channels to be more selective about which certificates are allowed to connect.

TIP: The solution to Module 3 is provided in the LabFiles folder on the desktop of the mqm account under the directory for Module 03. In addition, there is a zip file which can be imported into WMQ Explorer and which contains SSL connections for the MARS and VENUS QMgrs.

4. Fine-grained authentication

4.1 Module Objectives

This module will configure fine-grained authentication on the SYSTEM.ADMIN.SSL channel using channel authentication records which are a new feature in WebSphere MQ V7.1. These will be configured to map specific certificate Distinguished Name values to corresponding user accounts.

4.2 Background

In the first three modules, certificates for MARS and VENUS queue managers and the MQLab user account were generated and exchanged. As currently configured, any of these certificates can connect to the SYSTEM.ADMIN.SSL channel. Recall that the RCVR channels on MARS and VENUS used the SSLPEER channel attribute to select specific certificates which would be allowed to connect. This technique can be used when there is only one certificate that is allowed to connect to a given channel and is well suited use with to RCVR channels.

Channels for interactive users present a much different use case, however. Usually there are many users grouped into a number of roles – administrator, project teams, operations, and so forth. It would be possible but impractical to provide a dedicated channel and fully-qualified SSLPEER value for each user. A more practical approach would be to provide a small number of SVRCONN channels and map users to the channels based on user roles.

One way to do this would be to specify a generic SSLPEER value. This method relies on having a certificate Distinguished Name standard that maps the company's organizational structure into multiple OU values. For example, if the Distinguished Name contains the department and team it might be possible to devise an SSLPEER value that selects only the WMQ administrators. If you are fortunate enough to have anticipated this need prior to generating all your certificates then SSLPEER may be sufficient. However, this is the exception rather than the rule. Many shops need to use existing certificates which do not have sufficient granularity in the DN, or else their organization topology changes enough to invalidate the existing DN names. In these cases channel authentication records can provide an explicit mapping of individual certificates to specific channels.

Another consideration when setting up channel authentication is that a generic SSLPEER selects for a subset of certificates but cannot be changed to revoke access to individual certificates within that subset. A certificate revocation list or OCSP (Online Certificate Status Protocol) can provide per-certificate revocation capabilities but there is nothing the WebSphere MQ administrator can do to a generic SSLPEER that does not affect all users. Channel authentication records can address this either by specifying which are the acceptable certificates, and/or which certificates are to be rejected.

Whichever method is selected, the goal is to restrict on a per-channel basis the certificates that are allowed to connect. Channels that allow administrative access must allow only administrators to connect. Channels that are dedicated to a specific application must not allow other applications to connect. Channels that are dedicated to an adjacent queue manager must not allow connections from other queue managers.

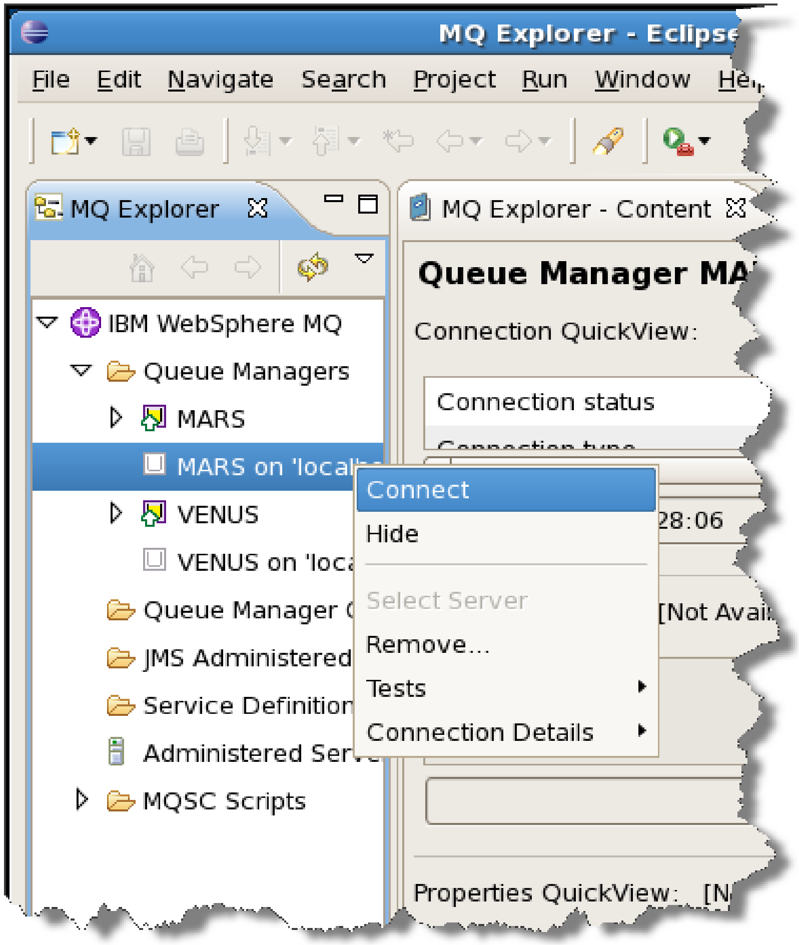
4.3 Without channel authentication records

Channel authentication records are a new feature in WebSphere MQ V7.1. They can be turned on and off with an ALTER QMGR command. In the previous module we turned channel authentication records off. Later in this module we will turn them on and make use of them.

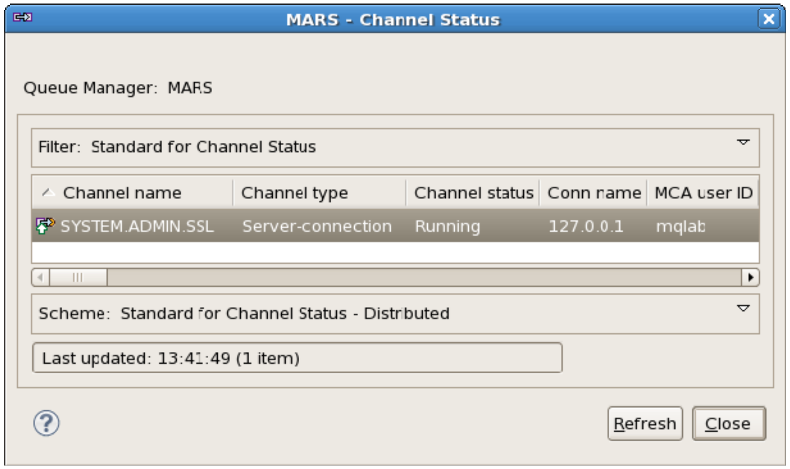
We will start by running without them to illustrate why your need to secure your queue manager. Brand new MQ V7.1 queue managers are created with channel authentication records enabled. Queue managers migrated from an earlier release start with them disabled.

1) ____ Ensure that channel authentication records are still turned off. Start runmqsc on the MARS queue manager and alter the queue manager as follows:

```
ALTER QMGR CHLAUTH(DISABLED)
```



2) ____ Make a connection to the SSL channel. Start MQ Explorer if it is not already running. Right-click on the disconnected MARS entry and select Connect.



MARS - Channel Status

Queue Manager: MARS

Filter: Standard for Channel Status

Channel name	Channel type	Channel status	Conn name	MCA user ID
SYSTEM.ADMIN.SSL	Server-connection	Running	127.0.0.1	mqlab

Scheme: Standard for Channel Status - Distributed

Last updated: 13:41:49 (1 item)

3) ____ Right-click on Channels and select Channel Status. The SYSTEM.ADMIN.SSL channel will be running with the mqlab user ID since that is the user ID presented by the client (i.e. the user ID the MQ Explorer is running under).

Remember to click the icon to show SYSTEM objects if you can't see any channels listed.

This user ID is in the mqm group and has privileges to do anything to the queue manager which is not what we want to allow.

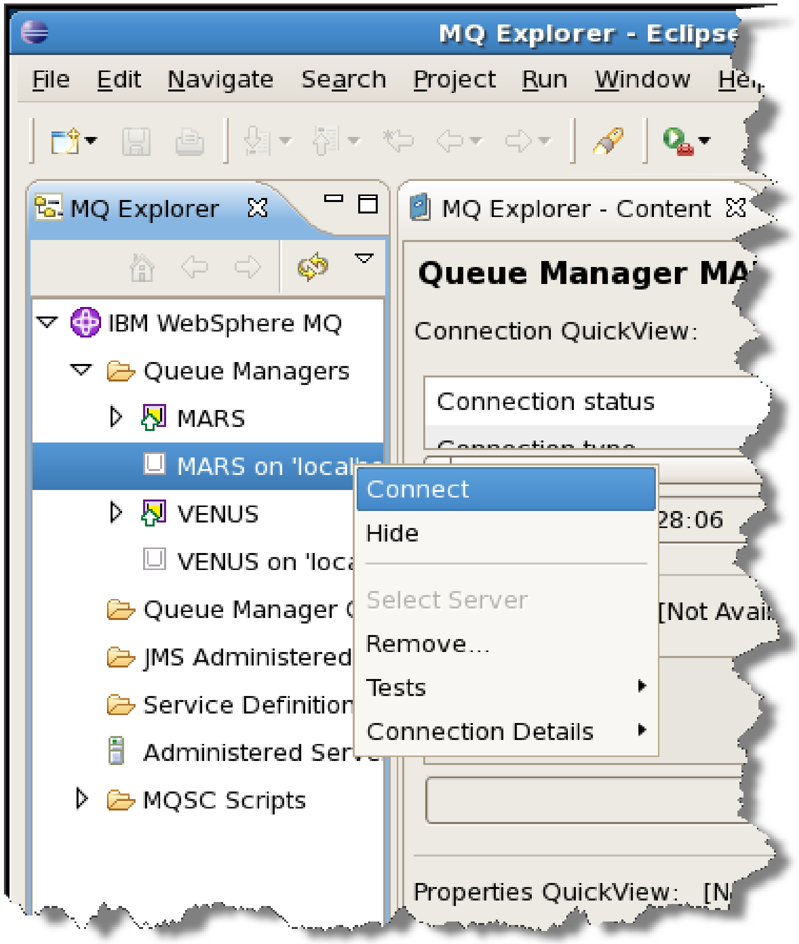
Now disconnect from MARS.

4.4 Default channel authentication rules

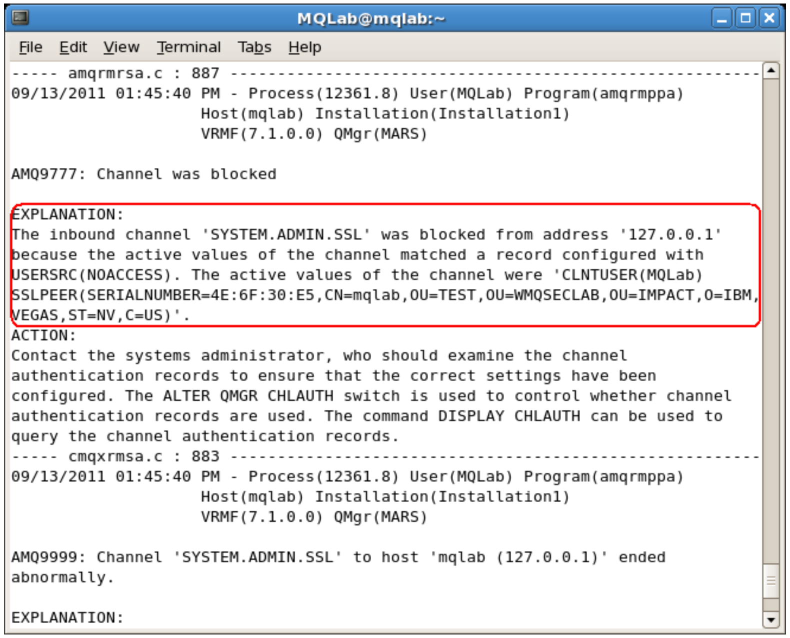
With channel authentication records enabled, WebSphere MQ V7.1 queue managers come with a few default rules. We will now enable channel authentication records and investigate the default rules.

1) ____ **On both MARS and VENUS queue managers**, start runmqsc and alter the queue manager as follows:

```
ALTER QMGR CHLAUTH(ENABLED)
```



2) ____ Make a connection to the SSL channel. Start MQ Explorer if it is not already running. Right click on the disconnect MARS entry and select connect. This connection will fail reporting that you are not authorized and we will now investigate why.

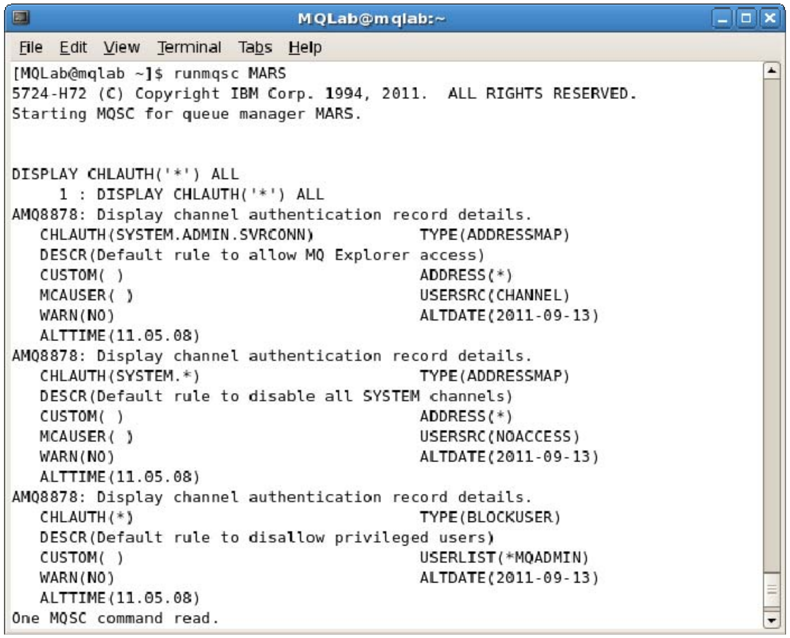


```
----- amqrmrsa.c : 887 ------------------------------------------------------
09/13/2011 01:45:40 PM - Process(12361.8) User(MQLab) Program(amqrmppa)
                    Host(mqlab) Installation(Installation1)
                    VRMF(7.1.0.0) QMgr(MARS)

AMQ9777: Channel was blocked

EXPLANATION:
The inbound channel 'SYSTEM.ADMIN.SSL' was blocked from address '127.0.0.1'
because the active values of the channel matched a record configured with
USERSRC(NOACCESS). The active values of the channel were 'CLNTUSER(MQLab)
SSLPEER(SERIALNUMBER=4E:6F:30:E5,CN=mqlab,OU=TEST,OU=WMQSECLAB,OU=IMPACT,O=IBM,
VEGAS,ST=NV,C=US)'.
ACTION:
Contact the systems administrator, who should examine the channel
authentication records to ensure that the correct settings have been
configured. The ALTER QMGR CHLAUTH switch is used to control whether channel
authentication records are used. The command DISPLAY CHLAUTH can be used to
query the channel authentication records.
----- cmqxrmsa.c : 883 ------------------------------------------------------
09/13/2011 01:45:40 PM - Process(12361.8) User(MQLab) Program(amqrmppa)
                    Host(mqlab) Installation(Installation1)
                    VRMF(7.1.0.0) QMgr(MARS)

AMQ9999: Channel 'SYSTEM.ADMIN.SSL' to host 'mqlab (127.0.0.1)' ended
abnormally.

EXPLANATION:
```

3) ____ Start by listing the AMQERR01.LOG file in the MARS queue manager directory. In the command terminal type:

```
cat /var/mqm/qmgrs/MARS/errors/AMQERR01.LOG
```

The output should look like the screen shot above. The error tells us that the connection has been blocked due to matching one of the channel authentication rules.



```
[MQLab@mqlab ~]$ runmqsc MARS
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager MARS.


DISPLAY CHLAUTH('*') ALL
     1 : DISPLAY CHLAUTH('*') ALL
AMQ8878: Display channel authentication record details.
   CHLAUTH(SYSTEM.ADMIN.SVRCONN)           TYPE(ADDRESSMAP)
   DESCR(Default rule to allow MQ Explorer access)
   CUSTOM( )                               ADDRESS(*)
   MCAUSER( )                              USERSRC(CHANNEL)
   WARN(NO)                                ALTDATE(2011-09-13)
   ALTTIME(11.05.08)
AMQ8878: Display channel authentication record details.
   CHLAUTH(SYSTEM.*)                       TYPE(ADDRESSMAP)
   DESCR(Default rule to disable all SYSTEM channels)
   CUSTOM( )                               ADDRESS(*)
   MCAUSER( )                              USERSRC(NOACCESS)
   WARN(NO)                                ALTDATE(2011-09-13)
   ALTTIME(11.05.08)
AMQ8878: Display channel authentication record details.
   CHLAUTH(*)                              TYPE(BLOCKUSER)
   DESCR(Default rule to disallow privileged users)
   CUSTOM( )                               USERLIST(*MQADMIN)
   WARN(NO)                                ALTDATE(2011-09-13)
   ALTTIME(11.05.08)
One MQSC command read.
```

4) ____ Start runmqsc on the MARS queue managers and list all the channel authentication rules as follows:

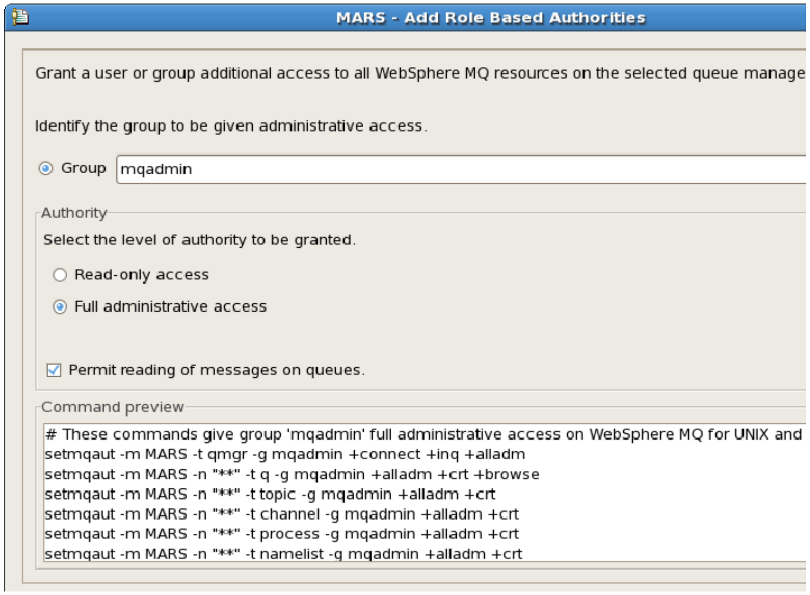
```
DISPLAY CHLAUTH(‘*’) ALL
```

We have matched the default rule that bans the use of SYSTEM channels. All the SYSTEM channels are disabled for use by the 2nd rule. The channel which the MQ Explorer uses by default, SYSTEM.ADMIN.SVRCONN, is enabled for use by the 1st rule, but if a privileged user id is asserted, such as our MQLab user id, this will be blocked by the 3rd rule. The special value of *MQADMIN equates to any privileged user id on this platform.

4.5 Configuring channel authentication records

Now that we have channel authentication records enabled we will configure a rule to allow our MQ Explorer to connect. We are going to allow the MQ Explorer to run with a user that is not in the mqm group, but that we will grant administrative rights to directly. First we grant that user the access we want, then we will map the SSL certificate MQ Explorer is using to that user ID.

1) ____ Start MQ Explorer If it is not already running. Right-click on the MARS queue manager and select 'Object Authorities; and then 'Add Role Based Authorities…'.



Grant a user or group additional access to all WebSphere MQ resources on the selected queue manage

Identify the group to be given administrative access.

● Group: mqadmin

Authority

Select the level of authority to be granted.

○ Read-only access

● Full administrative access

☑ Permit reading of messages on queues.

Command preview

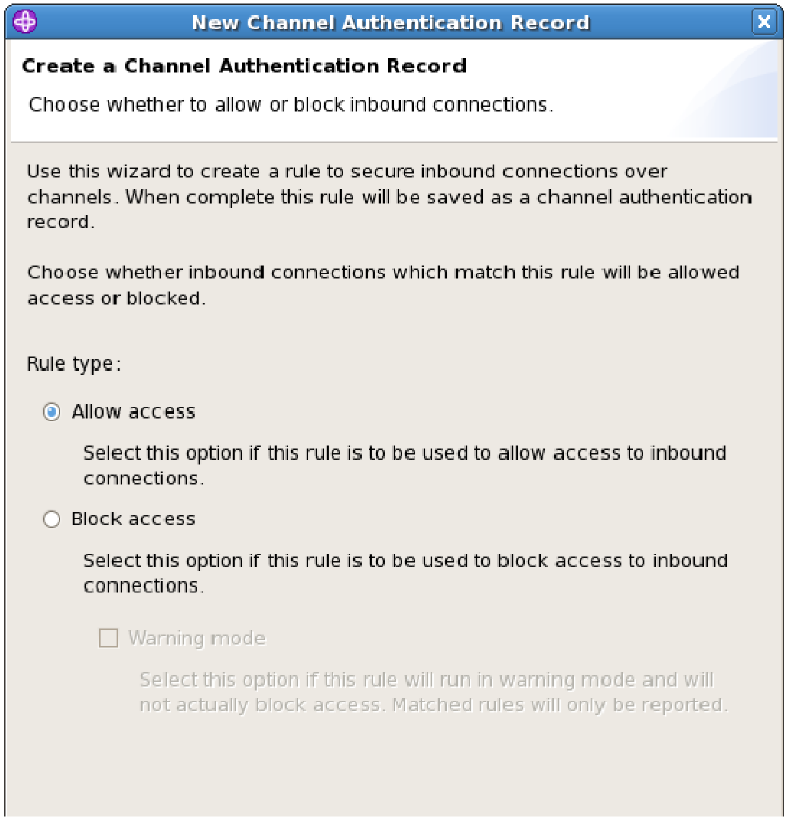
```
# These commands give group 'mqadmin' full administrative access on WebSphere MQ for UNIX and
setmqaut -m MARS -t qmgr -g mqadmin +connect +inq +alladm
setmqaut -m MARS -n "**" -t q -g mqadmin +alladm +crt +browse
setmqaut -m MARS -n "**" -t topic -g mqadmin +alladm +crt
setmqaut -m MARS -n "**" -t channel -g mqadmin +alladm +crt
setmqaut -m MARS -n "**" -t process -g mqadmin +alladm +crt
setmqaut -m MARS -n "**" -t namelist -g mqadmin +alladm +crt
```

2) ____ We will give user id 'mqadmin' full admin access. Select Group and type in mqadmin and choose the 'Full administrative access' option. Also check 'Permit reading of messages on a queue'. Review the commands that are going to be issued in the box at the bottom of the wizard before pressing OK.

3) ____ **Repeat steps 1 and 2 for the VENUS queue manager before continuing**

3) ____ Now we will map the certificate DN to the mqadmin user id using a channel authentication record. Open the Channels folder, right-click on the Channel Authentication Records folder and select New Channel Authentication Record.

The wizard that is shown will walk you through the process of creating the mapping that we need.



Create a Channel Authentication Record

Choose whether to allow or block inbound connections.

Use this wizard to create a rule to secure inbound connections over channels. When complete this rule will be saved as a channel authentication record.

Choose whether inbound connections which match this rule will be allowed access or blocked.

Rule type:

● Allow access

Select this option if this rule is to be used to allow access to inbound connections.

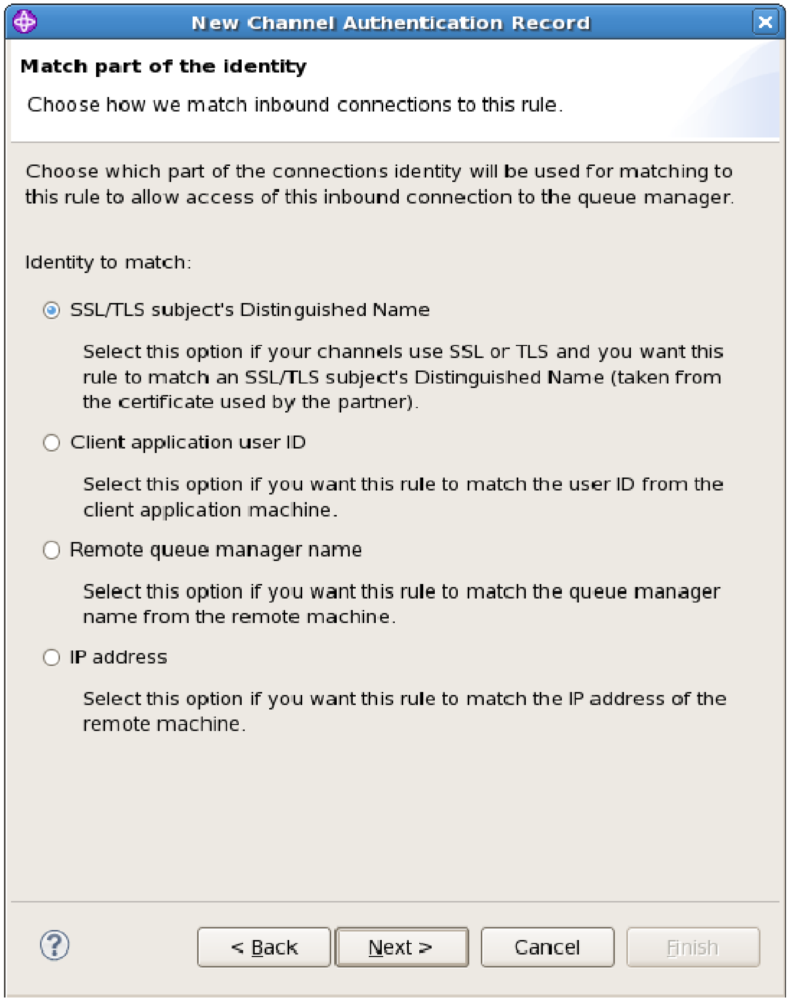
○ Block access

Select this option if this rule is to be used to block access to inbound connections.

□ Warning mode

Select this option if this rule will run in warning mode and will not actually block access. Matched rules will only be reported.

4) ____ On the first panel, choose a Rule Type of 'Allow access', and press Next.



New Channel Authentication Record

Match part of the identity

Choose how we match inbound connections to this rule.

Choose which part of the connections identity will be used for matching to this rule to allow access of this inbound connection to the queue manager.

Identity to match:

● SSL/TLS subject's Distinguished Name

Select this option if your channels use SSL or TLS and you want this rule to match an SSL/TLS subject's Distinguished Name (taken from the certificate used by the partner).

○ Client application user ID

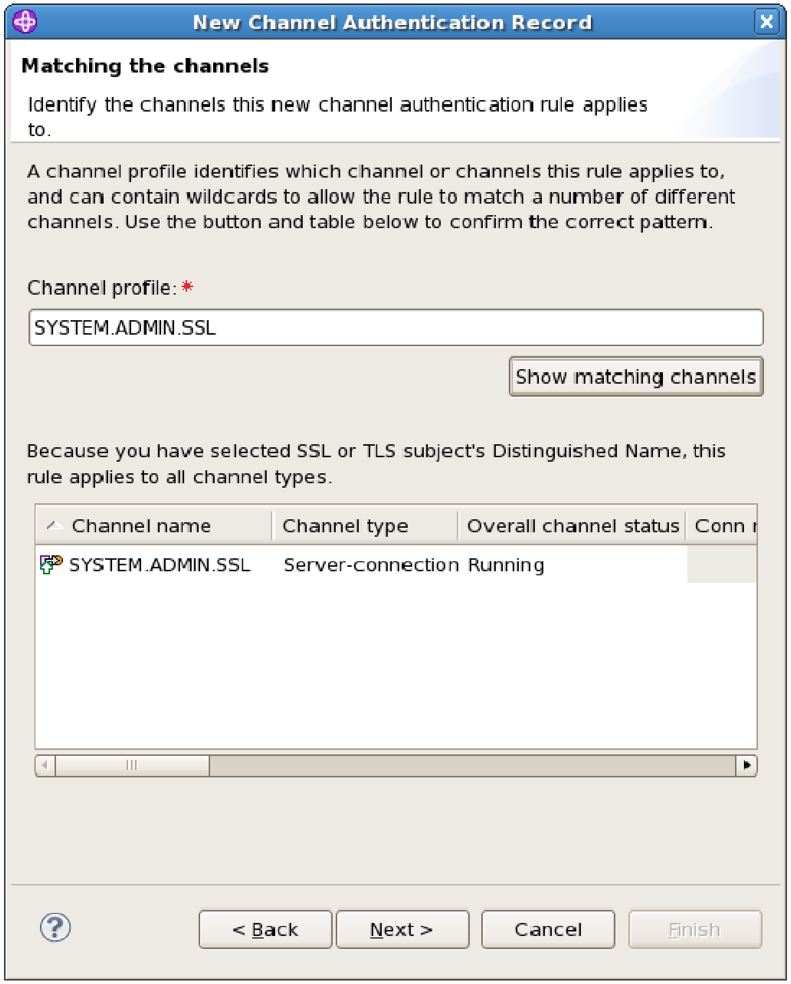
Select this option if you want this rule to match the user ID from the client application machine.

○ Remote queue manager name

Select this option if you want this rule to match the queue manager name from the remote machine.

○ IP address

Select this option if you want this rule to match the IP address of the remote machine.



New Channel Authentication Record

Matching the channels

Identify the channels this new channel authentication rule applies to.

A channel profile identifies which channel or channels this rule applies to, and can contain wildcards to allow the rule to match a number of different channels. Use the button and table below to confirm the correct pattern.

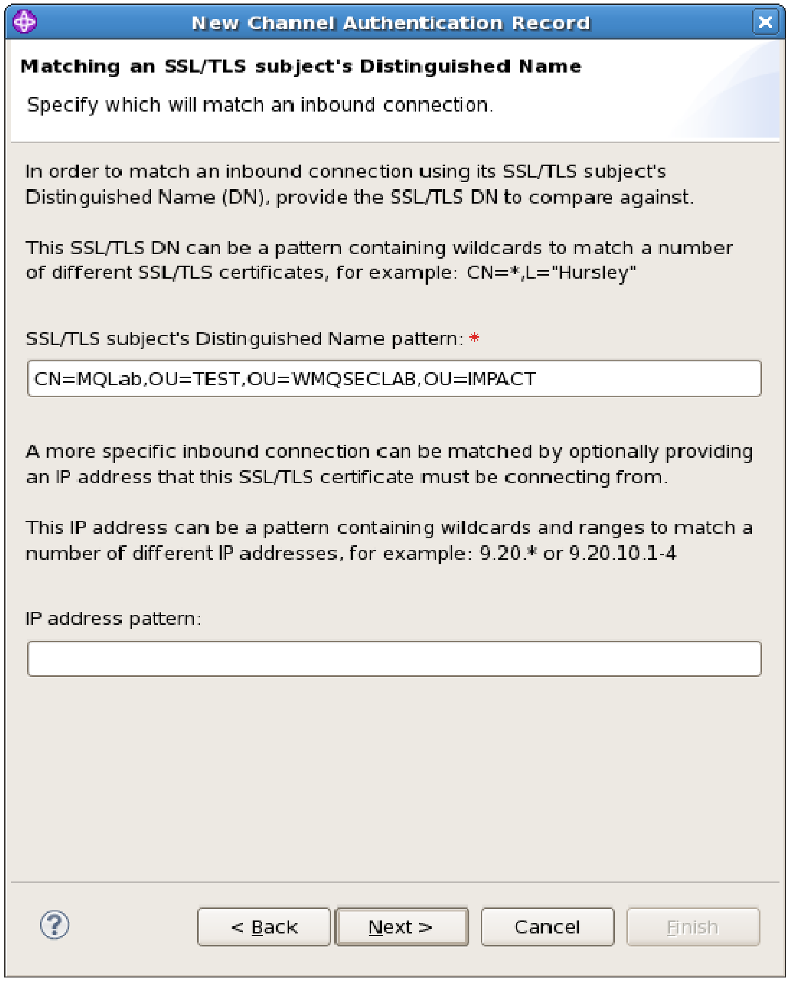
Channel profile: * SYSTEM.ADMIN.SSL

Because you have selected SSL or TLS subject's Distinguished Name, this rule applies to all channel types.

Channel name	Channel type	Overall channel status	Conn r
SYSTEM.ADMIN.SSL	Server-connection	Running	

5) ____ The part of the client's identity we want to use for this mapping is the SSL/TLS subject's Distinguished Name, so ensure that option is selected on the next panel and press Next.

6) ____ Now we select the channel or channels we want this rule to apply to. In our example, we will use the specific channel name of 'SYSTEM.ADMIN.SSL'. Feel free to try out some patterns, and use the 'Show matching channels' button to see how that works before moving on. Press the Next button to move to the next panel.



Matching an SSL/TLS subject's Distinguished Name

Specify which will match an inbound connection.

In order to match an inbound connection using its SSL/TLS subject's Distinguished Name (DN), provide the SSL/TLS DN to compare against.

This SSL/TLS DN can be a pattern containing wildcards to match a number of different SSL/TLS certificates, for example: CN=*,L="Hursley"

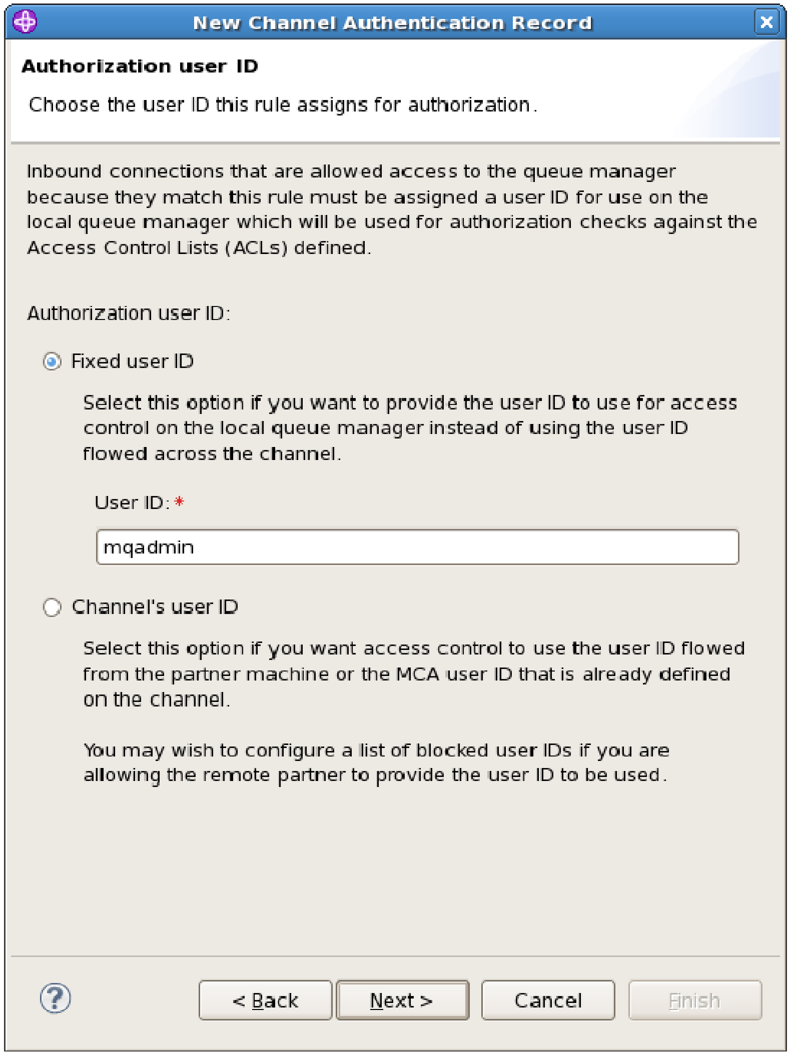
SSL/TLS subject's Distinguished Name pattern: *

CN=MQLab,OU=TEST,OU=WMQSECLAB,OU=IMPACT

A more specific inbound connection can be matched by optionally providing an IP address that this SSL/TLS certificate must be connecting from.

This IP address can be a pattern containing wildcards and ranges to match a number of different IP addresses, for example: 9.20.* or 9.20.10.1-4

IP address pattern:



Authorization user ID

Choose the user ID this rule assigns for authorization.

Inbound connections that are allowed access to the queue manager because they match this rule must be assigned a user ID for use on the local queue manager which will be used for authorization checks against the Access Control Lists (ACLs) defined.

Authorization user ID:

● Fixed user ID

Select this option if you want to provide the user ID to use for access control on the local queue manager instead of using the user ID flowed across the channel.

User ID: *

mqadmin

○ Channel's user ID

Select this option if you want access control to use the user ID flowed from the partner machine or the MCA user ID that is already defined on the channel.

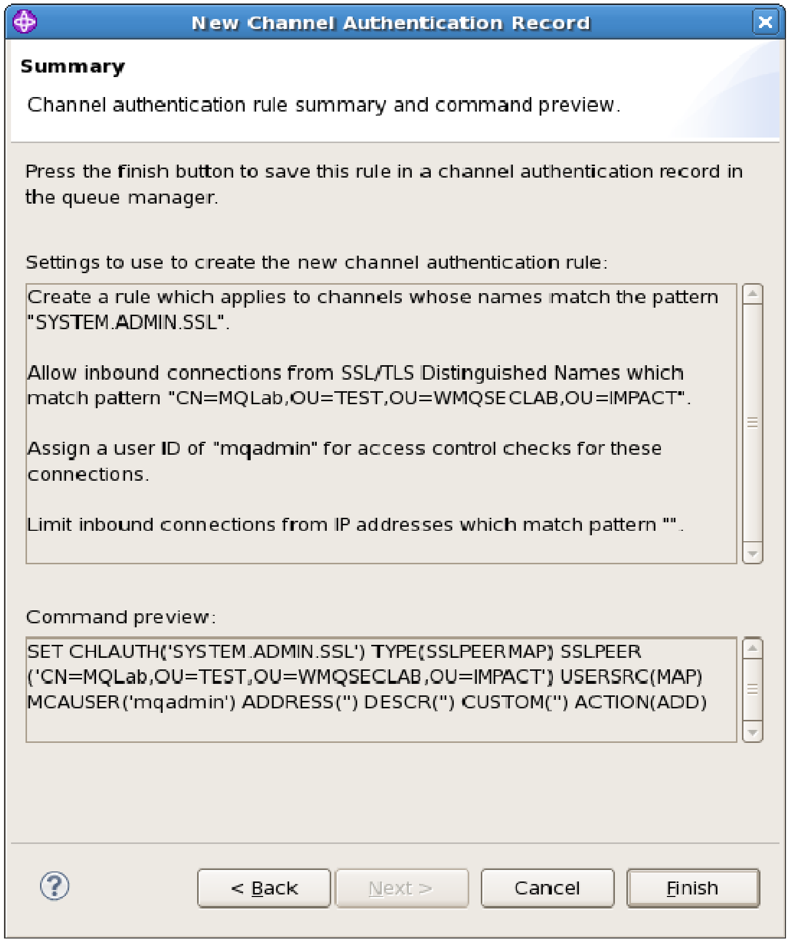
You may wish to configure a list of blocked user IDs if you are allowing the remote partner to provide the user ID to be used.

7) ____ Enter the DN we wish to allow into the first box on this panel:

`CN=MQLab,OU=TEST,OU=WMQSECLAB,OU=IMPACT`

Note, you can restrict this rule further by mandating that it only matches if the client also comes from a specific IP address. We will not be using this in the lab today.

8) ____ Now we indicate that this certificate DN will be mapped to a user id. Provide mqadmin as the user id to be used.



Summary

Channel authentication rule summary and command preview.

Press the finish button to save this rule in a channel authentication record in the queue manager.

Settings to use to create the new channel authentication rule:

Create a rule which applies to channels whose names match the pattern "SYSTEM.ADMIN.SSL".

Allow inbound connections from SSL/TLS Distinguished Names which match pattern "CN=MQLab,OU=TEST,OU=WMQSECLAB,OU=IMPACT".

Assign a user ID of "mqadmin" for access control checks for these connections.

Limit inbound connections from IP addresses which match pattern "".

Command preview:

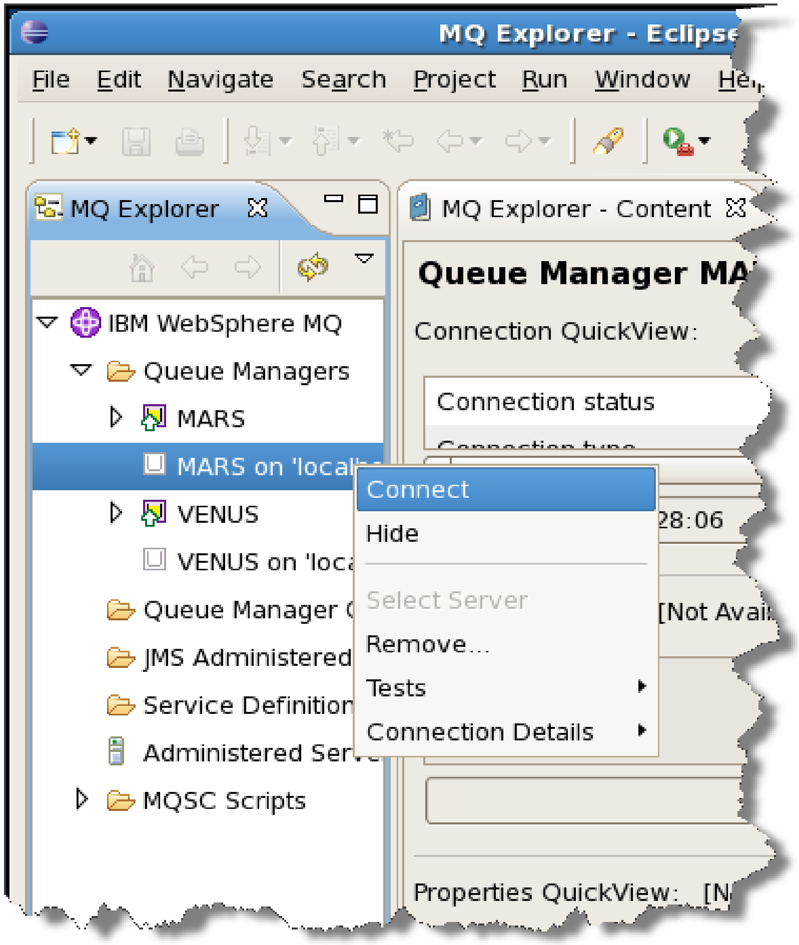
```
SET CHLAUTH('SYSTEM.ADMIN.SSL') TYPE(SSLPEERMAP) SSLPEER
('CN=MQLab,OU=TEST,OU=WMQSECLAB,OU=IMPACT') USERSRC(MAP)
MCAUSER('mqadmin') ADDRESS('') DESCR('') CUSTOM('') ACTION(ADD)
```

9) ____ Press Next until you get to the Summary page. Review the description of the rule you are creating and the MQSC equivalent command in the lower box. Then press Finish.

10) ____ We can do the same task via an MQSC command as indicated in the summary panel. We will use this method for the VENUS queue manager.

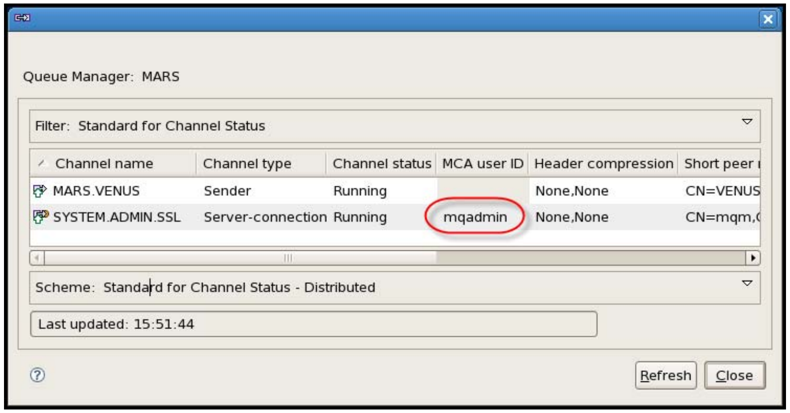
Start runmqsc on the VENUS queue manager and create a channel authentication record as follows:

```
SET CHLAUTH(SYSTEM.ADMIN.SSL) TYPE(SSLPEERMAP) +
SSLPEER(‘CN=MQLab,OU=TEST,OU=WMQSECLAB,OU=IMPACT’) +
MCAUSER(‘mqadmin’)
```



11) ____ Test the configuration by making a connection to the SSL channel. Start WMQ Explorer if it is not already running. Right-click on the disconnected MARS entry and select Connect

12) **Repeat Step 11 for the VENUS queue manager before continuing**.


Queue Manager: MARS

Filter: Standard for Channel Status

Channel name	Channel type	Channel status	MCA user ID	Header compression	Short peer
MARS.VENUS	Sender	Running		None,None	CN=VENUS
SYSTEM.ADMIN.SSL	Server-connection	Running	mqadmin	None,None	CN=mqm,(

Scheme: Standard for Channel Status - Distributed

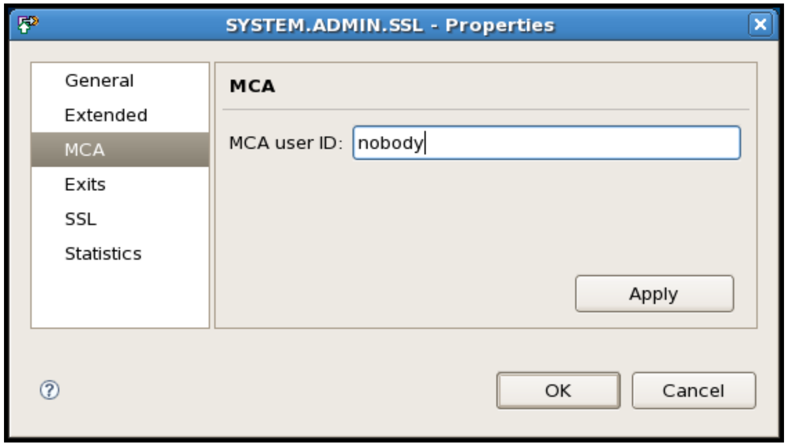
Last updated: 15:51:44

05) ____ The channel now accepts only the certificate specified in the channel authentication records we set earlier and maps certificate name to a specific account.

Reconnect to MARS, right-click on Channels and select Channel Status. The SYSTEM.ADMIN.SSL channel will now be running with the mqadmin user ID configured in WMQ Explorer. The channel authentication record has overridden the ID presented by WebSphere MQ Explorer. Channel authentication records can be set for each user. Some shops may find it convenient to map users to roles. For example, all administrators may be mapped to mqadmin, all users in Project XYX may be mapped to a prj_xyz account, and so on.

Alternatively, it is possible to map each user's certificate to their actual user ID on the local system. Although this incurs more administrative overhead, it is useful when regulatory or internal requirements dictate that connections be traceable back to specific users.

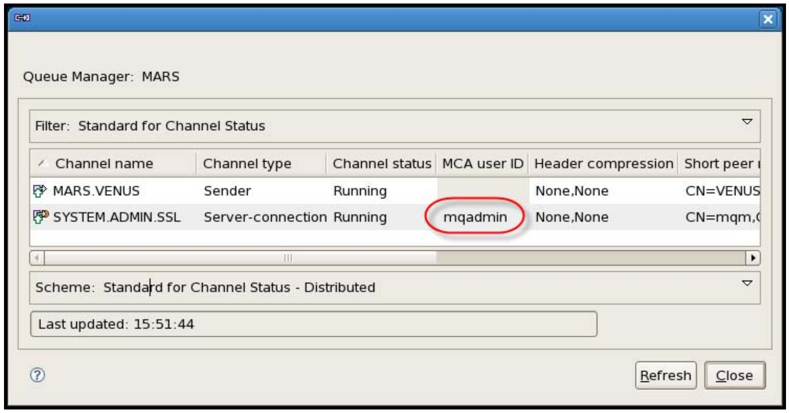
Whichever method is chosen, creating channel authentication records to set the MCAUSER of the channel at run time allows a single channel to serve many users while still enforcing authorizations on a per-user basis.



MCA user ID: nobody

11) ____ By default the MCAUSER of all channels is blank. As we have seen, this allows the user ID to be specified in the connection request, but if that ID is a privileged administrative ID it is blocked by the default rules. To address this, it is necessary to set the MCAUSER value of the channel to the desired user ID. Now that channel authentication rules are dynamically setting the value at connect time, it is no longer necessary to have a useable value in the channel's object definition. In fact, it is useful to have a non-working value in the object definition so that if the channel authentication records are configured improperly, the channel fails to a safe state.

From WMQ Explorer, connect to MARS and select the Channels panel from the navigation tree. Double-click the SYSTEM.ADMIN.SSL channel and select MCA from the navigation tree. Enter the value nobody and click the OK button.



Queue Manager: MARS

Filter: Standard for Channel Status

Channel name	Channel type	Channel status	MCA user ID	Header compression	Short peer
MARS.VENUS	Sender	Running		None,None	CN=VENUS
SYSTEM.ADMIN.SSL	Server-connection	Running	mqadmin	None,None	CN=mqm,(

Scheme: Standard for Channel Status - Distributed

Last updated: 15:51:44

12) ____ Disconnect and reconnect the localhost connection to the MARS queue manager. Verify that the channel's MCAUSER is set to nobody in the channel definition. Then verify that the MCAUSER in the channel status reports mqadmin.

13) ____ Change the MCAUSER of the SYSTEM.ADMIN.SSL channel on VENUS to nobody before continuing.

4.6 Module 4 Summary

This module used channel authentication records to map certificate distinguished names to user IDs. This is the means by which authentication can be meaningfully tied to authorization.

As an added security measure, the MCAUSER of the channel was set to nobody. This insures that the channel can only be activated if channel authentication rules are correctly configured and validate the requestor's distinguished name.

5. Configure authorizations

5.1 Module objectives

The goal of this module is to configure the channels in a way that enforces authorizations for non-administrative users. This will include MCA channels (those of type RCVR, RQSTR and CLUSRCVR) as well as MQI channels (those of type SVRCONN).

5.2 Background

Authorization is the aspect of WebSphere MQ security that is usually performed first and, unfortunately, is frequently the only security configuration that is performed. As we have seen in the previous lab modules, enforcing authorization without authentication does not effectively protect the queue manager. When authorization is configured without authentication a security façade is created. It provides the appearance of security because legitimate users who connect using the prescribed credentials are locked out if they connect to the wrong channel or queue manager. However, a malicious attacker would simply connect to one of the unauthenticated channels or present an administrative user ID and gain access.

A system configured without authentication is arguably worse than one that is wide open. When a network has no controls by design or due to a lack or resources to secure it, users are wary of trusting the network with valuable data. However when the network has ineffective controls, users believing it to be secure do not exercise the same level of caution. Similarly, a network believed to be secure may not be subject to the same level of testing or review as one that is known to be vulnerable.

It is for this reason that the lab is structured to configure the authentication steps in the previous modules. Authentication should be the first priority when hardening the MQ network. This module will build on that foundation by enforcing authorization profiles which will restrict the actions which may be executed by an authenticated user ID..

5.3 Authorizing adjacent queue managers

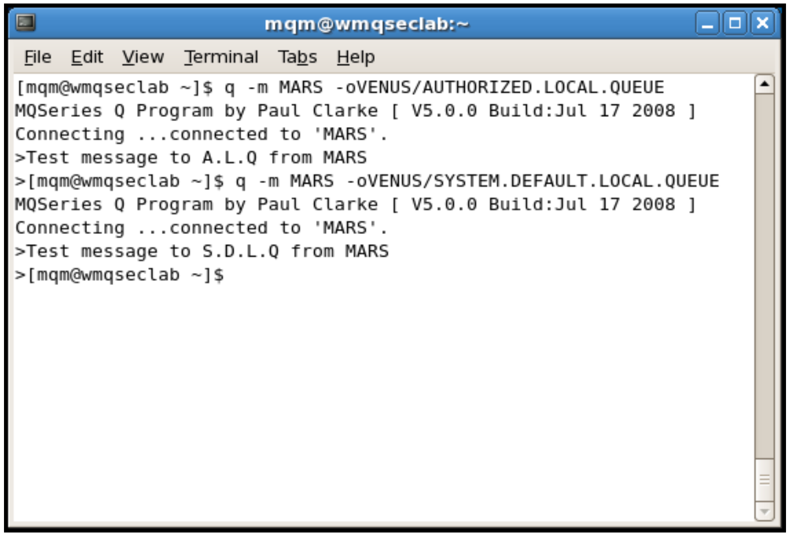
By default, an adjacent queue manager can put messages to any local queue and administer the local queue manager. The most basic security improvement is to continue to allow messages to be put to any queue, with the exception of those queues that confer administrative rights. The set of commands below will restrict the queues on the VENUS queue manager to which the MARS queue manager is allowed to put messages.

```
* Allow MCAUSER to connect.  Needs setall per IBM docs.
SET AUTHREC OBJTYPE(QMGR) GROUP('mqmmca') AUTHADD(CONNECT, INQ, SETALL)

* Grant MCAUSER default policy of "allow all" to all queues.  Channels
* just put messages so no need for get, browse, etc.  Also needs setall.
SET AUTHREC PROFILE('**') OBJTYPE(QUEUE) GROUP('mqmmca') AUTHADD(PUT, SETALL)

* Now deny access to SYSTEM.** queues
SET AUTHREC PROFILE('SYSTEM.**') OBJTYPE(QUEUE) GROUP('mqmmca') AUTHRMV(ALL)

* And to transmit queues
SET AUTHREC PROFILE('MARS') OBJTYPE(QUEUE) GROUP('mqmmca') AUTHRMV(ALL)

* Grant access to the DLQ so the channel doesn't stop on delivery errors
SET AUTHREC PROFILE('SYSTEM.DEAD.LETTER.QUEUE') OBJTYPE(QUEUE) GROUP('mqmmca') AUTHADD(PUT, SETALL)

* Activate the security by placing the mqmmca user ID into the channel's
* MCAUSER and restarting the channel:
ALTER CHL(MARS.VENUS) CHLTYPE(RCVR) MCAUSER('mqmmca')
```

01) ____ Right-click on the desktop to open a terminal. Start runmqsc on the VENUS queue manager and paste the lines above into the runmqsc window. Alternatively, pipe the MQSC script into runmqsc. It can be found in the LabFiles folder for this module. The LabFiles folder is located on the desktop, and on the command can be found in ~/scripts. The sub-folder for Module 5 contains several scripts including the one above which is filed as mqmmca_VENUS.mqs.

02) ____ Stop and restart the sender channel called MARS.VENUS on queue manager MARS.

5.4 Test the authorization



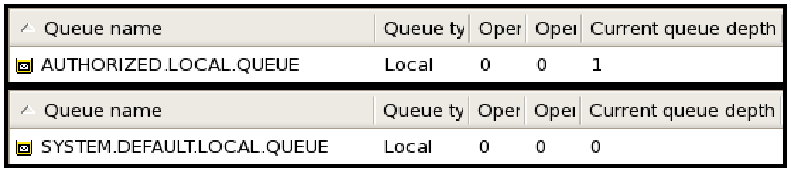
```
[mqm@wmqseclab ~]$ q -m MARS -oVENUS/AUTHORIZED.LOCAL.QUEUE
MQSeries Q Program by Paul Clarke [ V5.0.0 Build:Jul 17 2008 ]
Connecting ...connected to 'MARS'.
>Test message to A.L.Q from MARS
>[mqm@wmqseclab ~]$ q -m MARS -oVENUS/SYSTEM.DEFAULT.LOCAL.QUEUE
MQSeries Q Program by Paul Clarke [ V5.0.0 Build:Jul 17 2008 ]
Connecting ...connected to 'MARS'.
>Test message to S.D.L.Q from MARS
>[mqm@wmqseclab ~]$
```

01) ____ Use the Q program from SupportPac MA01 to send messages from MARS to VENUS. This exercises the authorization profiles associated with the mqmmca ID in the MARS.VENUS channel's MCAUSER. In a terminal window, type:

```
q -m MARS –oVENUS/AUTHORIZED.LOCAL.QUEUE
```

Type in a message and hit <ENTER> followed by #end to exit the program.

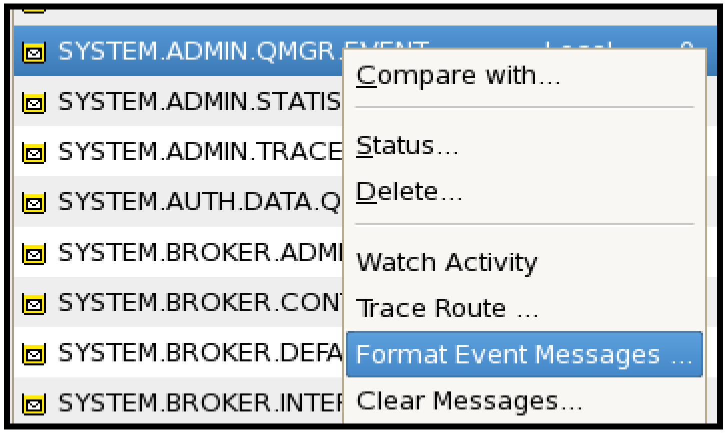
02) ____ Repeat the exercise above but this time send the message to the SYSTEM.DEFAULT.LOCAL.QUEUE on VENUS.



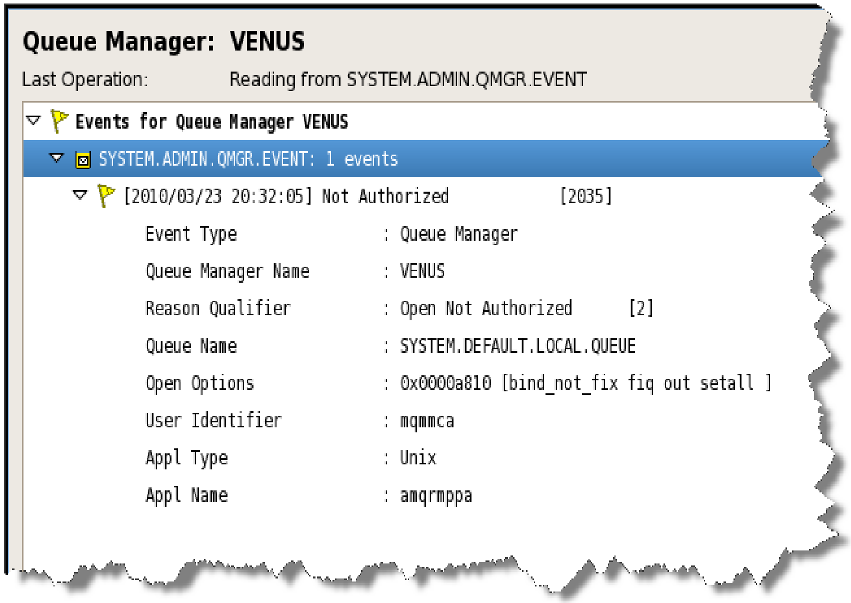
Queue name	Queue ty	Oper	Oper	Current queue depth
AUTHORIZED.LOCAL.QUEUE	Local	0	0	1

Queue name	Queue ty	Oper	Oper	Current queue depth
SYSTEM.DEFAULT.LOCAL.QUEUE	Local	0	0	0

03) ____ Open WebSphere MQ Explorer and navigate to the Queues panel for the VENUS queue manager. Note that the queue depth has increased in AUTHORIZED.LOCAL.QUEUE but that the queue depth of SYSTEM.DEFAULT.LOCAL.QUEUE remains at zero. This is because the channel, running as mqmmca, failed authorization on the put to the SYSTEM.DEFAULT.LOCAL.QUEUE. We can verify this using the MS0P Event Message Formatter plug-in.



04) ____ Right-click on the SYSTEM.ADMIN.QMGR.EVENT queue in WMQ Explorer. In the context menu, select Format Event Messages.

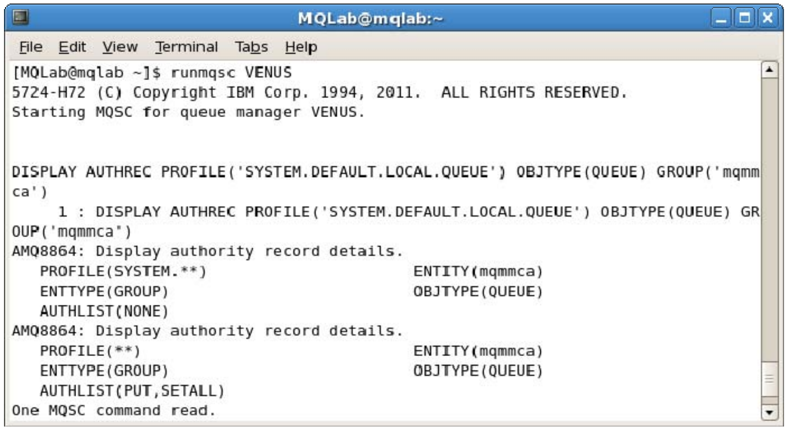


Queue Manager: VENUS

Last Operation: Reading from SYSTEM.ADMIN.QMGR.EVENT

```
▽ Events for Queue Manager VENUS
  ▽ SYSTEM.ADMIN.QMGR.EVENT: 1 events
    ▽ [2010/03/23 20:32:05] Not Authorized          [2035]
          Event Type                : Queue Manager
          Queue Manager Name        : VENUS
          Reason Qualifier          : Open Not Authorized      [2]
          Queue Name                : SYSTEM.DEFAULT.LOCAL.QUEUE
          Open Options              : 0x0000a810 [bind_not_fix fiq out setall ]
          User Identifier           : mqmmca
          Appl Type                 : Unix
          Appl Name                 : amqrmppa
```

05) ____ Click past the selection dialog to arrive at the message detail screen. Click the twistie to expand the event message. We can see that the event was generated by an authorization error on the open call for the SYSTEM.DEFAULT.LOCAL.QUEUE. We can also tell that the application that failed was the channel agent (amqrmppa), the user ID which that executed the failing API call and the exact options that were used in the call.



```
[MQLab@mqlab ~]$ runmqsc VENUS
5724-H72 (C) Copyright IBM Corp. 1994, 2011.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager VENUS.


DISPLAY AUTHREC PROFILE('SYSTEM.DEFAULT.LOCAL.QUEUE') OBJTYPE(QUEUE) GROUP('mqmm
ca')
     1 : DISPLAY AUTHREC PROFILE('SYSTEM.DEFAULT.LOCAL.QUEUE') OBJTYPE(QUEUE) GR
OUP('mqmmca')
AMQ8864: Display authority record details.
   PROFILE(SYSTEM.**)                      ENTITY(mqmmca)
   ENTTYPE(GROUP)                          OBJTYPE(QUEUE)
   AUTHLIST(NONE)
AMQ8864: Display authority record details.
   PROFILE(**)                             ENTITY(mqmmca)
   ENTTYPE(GROUP)                          OBJTYPE(QUEUE)
   AUTHLIST(PUT,SETALL)
One MQSC command read.
```

06) ____ It is often useful to know which authorization profiles contributed to an authorization failure. To display these, start runmqsc on the VENUS queue manager and issue the following command:

```
DISPLAY AUTHREC +
PROFILE('SYSTEM.DEFAULT.LOCAL.QUEUE') +
OBJTYPE(QUEUE) GROUP('mqmmca')
```

According to the event message, the queue was opened for output and setall. The result of the command shows that the generic profile of '**' granted these rights to the mqmmca group, but that they were overridden by the more specific profile which revokes all rights to SYSTEM.** objects for the mqmmca group.

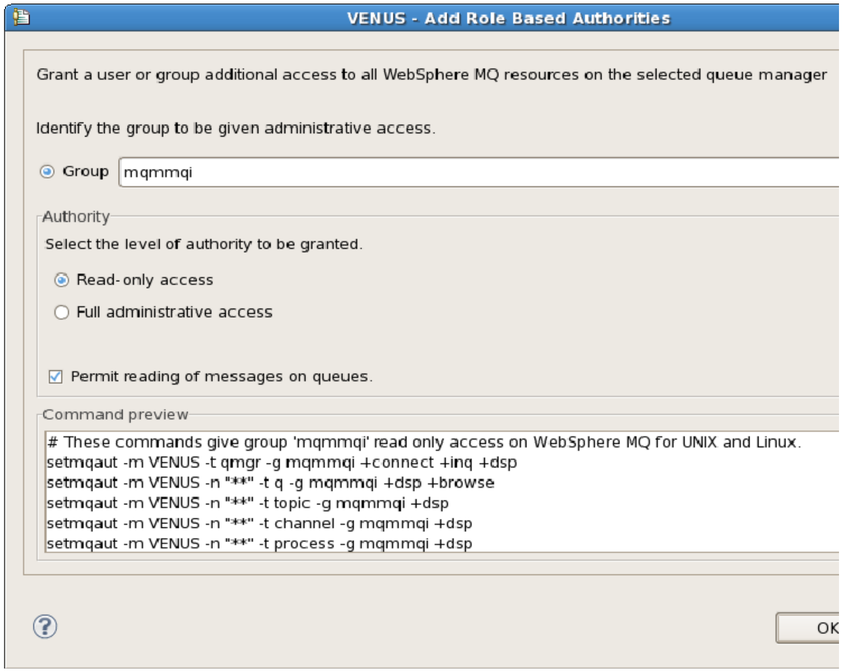
5.5 Authorizing WMQ Explorer

In earlier modules, the SYSTEM.ADMIN.SVRCONN channel was left allowing any non privileged user able to connect and a dedicated SVRCONN channel was created for the administrators. The administrative channel is authenticated and allows full access. The opposite end of the continuum would be a channel that allows anonymous connections but very little access. Since SYSTEM.ADMIN.SVRCONN is a well-known channel name, it is a good candidate for such a channel. We are going to use the Role Based Authorities wizard to grant a user read-only access to the queue manager and use that user ID on the SYSTEM.ADMIN.SVRCONN channel.

07) ____ Start runmqsc on the VENUS queue managers and alter the channel as follows:

```
ALTER CHANNEL(SYSTEM.ADMIN.SVRCONN) +
CHLTYPE(SVRCONN) MCAUSER('mqmmqi')
```

07) ____ Start MQ Explorer If it is not already running. Right-click on the VENUS queue manager and select ‘Object Authorities; and then ‘Add Role Based Authorities…’.



VENUS - Add Role Based Authorities

Grant a user or group additional access to all WebSphere MQ resources on the selected queue manager

Identify the group to be given administrative access.

● Group mqmmqi

Authority

Select the level of authority to be granted.

● Read-only access

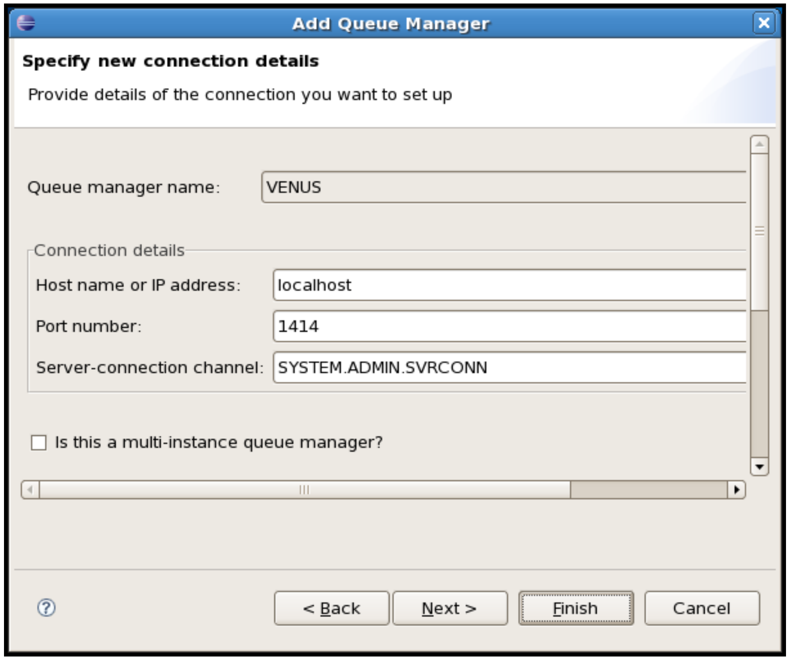
○ Full administrative access

☑ Permit reading of messages on queues.

Command preview

```
# These commands give group 'mqmmqi' read only access on WebSphere MQ for UNIX and Linux.
setmqaut -m VENUS -t qmgr -g mqmmqi +connect +inq +dsp
setmqaut -m VENUS -n "**" -t q -g mqmmqi +dsp +browse
setmqaut -m VENUS -n "**" -t topic -g mqmmqi +dsp
setmqaut -m VENUS -n "**" -t channel -g mqmmqi +dsp
setmqaut -m VENUS -n "**" -t process -g mqmmqi +dsp
```

08) ____ We will give user id ‘mqmmqi; read-only access. Select Group and type in mqmmqi and choose the ‘Read-only access’ option. Also check ‘Permit reading of messages on a queue’. Review the commands that are going to be issued in the box at the bottom of the wizard before pressing OK.



Add Queue Manager

Specify new connection details

Provide details of the connection you want to set up

Queue manager name: VENUS

Connection details

Host name or IP address: localhost

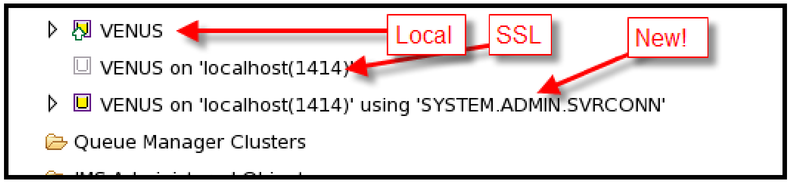
Port number: 1414

Server-connection channel: SYSTEM.ADMIN.SVRCONN

☐ Is this a multi-instance queue manager?

08) ____ To test the new profiles, make a new connection to the VENUS queue manager using the SYSTEM.ADMIN.SVRCONN channel. Right-click on Queue Managers and select Add Remote Queue Manager. In the dialog, enter VENUS and click Next. In the Details dialog, enter localhost as the host name and click Finish.

Of the three entries for the VENUS queue manager, the restricted one is identified by the channel name.

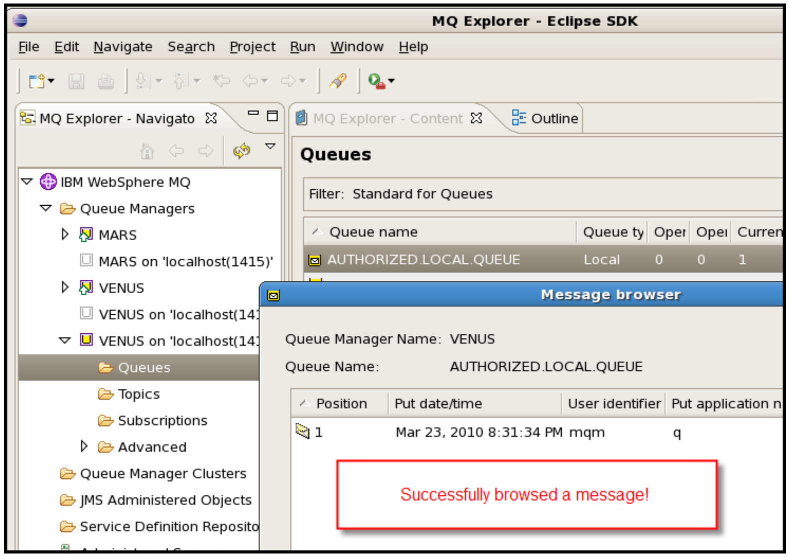


VENUS

VENUS on 'localhost(1414)'

VENUS on 'localhost(1414)' using 'SYSTEM.ADMIN.SVRCONN'

Queue Manager Clusters



MQ Explorer - Eclipse SDK

Queues

Filter: Standard for Queues

Queue name	Queue ty	Oper	Oper	Curren
AUTHORIZED.LOCAL.QUEUE	Local	0	0	1

Message browser

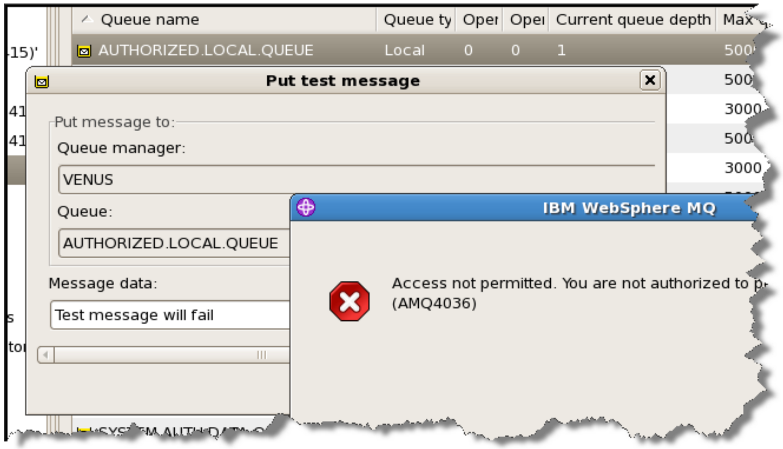
Queue Manager Name: VENUS

Queue Name: AUTHORIZED.LOCAL.QUEUE

Position	Put date/time	User identifier	Put application n
1	Mar 23, 2010 8:31:34 PM	mqm	q

Successfully browsed a message!

09) ____ Try to display various objects on the queue manager. Look at channel statuses. Browse the message that you left on the AUTHORIZED.LOCAL.QUEUE. Note that any attempt to put or destructively get a message or to change an object setting will fail.



Queue name	Queue ty	Oper	Oper	Current queue depth	Max
AUTHORIZED.LOCAL.QUEUE	Local	0	0	1	500

Put test message

Put message to:

Queue manager: VENUS

Queue: AUTHORIZED.LOCAL.QUEUE

Message data: Test message will fail

IBM WebSphere MQ

Access not permitted. You are not authorized to p
(AMQ4036)

6. Summary

This has been a brief overview of hardening a WebSphere MQ queue manager. The high-level steps involved were:

1. Configure SSL between queue managers.
2. Configure SSL for administrative SVRCONN connections.
3. Lock down any unused channels.
4. Map SSL Distinguished Names to user accounts using channel authentication records.
5. Map SSL Distinguished Names to user accounts using SSLPEER.
6. Provision a channel for non-administrative access.
7. Restrict authority of the MCA channels (adjacent queue managers).
8. Restrict authority of non-administrative SVRCONN channels.

Remediation of a WebSphere MQ network would include several more roles, such as developers, testers, instrumentation, end users, and so on. However the high-level process to implement the configurations would remain the same: provide authentication, then provide authorization.

The tools that were used included:

- SupportPac MO04 WebSphere MQ SSL Wizard
- SupportPac MS0P WMQ Explorer Configuration and Display Plug-In
- SupportPac MS0L LSB Init Scripts for WebSphere MQ
- A variety of scripts

You are encouraged to branch off from this point and extend the lab. The scripts and guide were designed to be extensible and portable to most UNIX platforms. The obvious next step is to fill in the middle ground between the administrative access that you configured and the read-only WebSphere MQ explorer. Just remember that the following rights implicitly grant administrative authority so do not grant them out to ordinary users or applications:

- The ability to create a queue other than from a model queue.
- The ability to set attributes of the queue manager.
- The +all +alladm +allapi +setid rights.
- The +setall right (except for RCVR, RQSTR or CLUSRCVR channels).
- The ability to read the SSL keystore files.
- The ability to write to the qm.ini file.
- The ability to write to the exits directory.

I hope that you found the lab enjoyable as well as informative. Please send suggestions, bug reports and comments to the author at t.rob.wyatt@us.ibm.com.

7. Legal Notices

IBM, the IBM logo, the e-business logo and other IBM products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

Product release dates and/or capabilities referenced in this publication may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

All other trademarks, company, products or service names may be trademarks, registered trademarks or service marks of others.