

```
rem =====
rem   One possible approach to creating the SSL key repositories
rem   for two queue managers
rem
rem   Using personal certificates only.
rem
rem   The commands can be run (as written here) on a single machine
rem   and then the completed key repositories moved into locations
rem   accessible by the queue managers
rem
rem   Dale Lane (http://hursleyonwmq.wordpress.com/)
rem =====

REM *****
REM *** ENVIRONMENT
REM *****
rem *** path for WebSphere MQ
set MQBASE=C:\Program Files\IBM\WebSphere MQ

set PASSWORD=passw0rd

REM *** command name (gsk7cmd on UNIX, runmqckm on Windows)
set GSK7CMD=runmqckm

REM -----
REM   On Windows, using runmqckm acts as a wrapper for the GSKit command
REM   gsk7cmd in the correct environment. Using runmqckm means you do
REM   not need the following commands.
REM   Alternatively, you could use gsk7cmd, and use the following two
REM   commands to set the environment manually.
REM -----
rem *** Set the path to the GSKit programs used to create the repository ***
rem set PATH=%PATH%;C:\Program Files\IBM\gsk7\bin
rem *** Set the path to the JRE installed by WMQ for GSKit ***
rem set JAVA_HOME=%MQBASE%\gskit\jre
REM -----

REM *****
REM   lowercase!
REM   when used in label names, we need
REM   queue manager names in lowercase,
REM   regardless of the case of the qmgr
REM   names
REM *****
set QMGR1NAME=qmgr1
set QMGR2NAME=qmgr2
```

```
REM *****
REM create repositories for use by queue managers to store keys
REM
REM these should be moved to the SSL directory of the relevant
REM queue manager, or the queue manager SSLKEYR attribute
REM altered to point at this location
REM *****
%GSK7CMD% -keydb -create -db qmgr1.kdb -pw %PASSWORD% -type cms -stash
%GSK7CMD% -keydb -create -db qmgr2.kdb -pw %PASSWORD% -type cms -stash

REM *****
REM create certificates for use by queue managers
REM
REM once created, the public keys are exported
REM for adding to repositories for other queue
REM managers
REM *****
rem *** Create a certificate to be signed for QMGR1 ***
%GSK7CMD% -cert -create -db qmgr1.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR1NAME% -dn "CN=Qmgr1,O=IBM,OU=Hursley blog,L=Hursley,C=UK"
rem *** Extract the public key for QMGR1 for use with other queue managers ***
%GSK7CMD% -cert -extract -db qmgr1.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR1NAME% -target qmgr1cert.arm

rem *** Create a certificate to be signed for QMGR2 ***
%GSK7CMD% -cert -create -db qmgr2.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR2NAME% -dn "CN=Qmgr2,O=IBM,OU=Hursley blog,L=Hursley,C=UK"
rem *** Extract the public key for QMGR1 for use with other queue managers ***
%GSK7CMD% -cert -extract -db qmgr2.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR2NAME% -target qmgr2cert.arm

REM *****
REM add public keys for use by queue managers
REM
REM each queue manager needs the public key for
REM each other queue manager it will connect to
REM *****
rem *** add the public key for QMGR2 to the QMGR1 key repository ***
%GSK7CMD% -cert -add -db qmgr1.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR2NAME% -file qmgr2cert.arm

rem *** add the public key for QMGR1 to the QMGR2 key repository ***
%GSK7CMD% -cert -add -db qmgr2.kdb -pw %PASSWORD% -label ibmwebspheremq%QMGR1NAME% -file qmgr1cert.arm
```

```
REM -----  
REM THE FOLLOWING FILES REMAIN:  
REM -----  
REM QUEUE MANAGER KEY REPOSITORIES  
REM   qmgr1.kdb (and associated stash file qmgr1.sth)  
REM   qmgr2.kdb (and associated stash file qmgr2.sth)  
REM   are now ready for use by the queue managers  
REM -----  
REM   qmgr1cert.arm  
REM   qmgr2cert.arm  
REM   these are the queue manager certificates for  
REM   importing into each queue manager repository  
REM   and can now be deleted  
REM -----  
  
rem =====  
rem   END  
rem =====
```