Using SSL to Connect to a WebSphere Application Server with a WebSphere MQ Queue Manager

IBM Techdoc: 7021934

http://www.ibm.com/support/docview.wss?rs=171&uid=swg27021934

Date last updated: 06-Jul-2011

Miguel Rodriguez – mrod@us.ibm.com
Angel Rivera – rivera@us.ibm.com
IBM WebSphere MQ Support

+++ Objective

The objective of this technical document is to describe in detail how to configure the connection between a WebSphere Application Server V7 with a WebSphere MQ Queue Manager V7 using Secured Sockets Layer (SSL).

The focus of this techdoc is to provide the steps and the commands that you need to perform to configure the secured connection, and using self-signed certificates which you can generate for your testing.
The target platforms are these distributed ones: Unix and Windows.

It is not the intention of this document to provide the background and the explanation of what is SSL. Also, this document does not cover advanced features, such as certificate revocation lists or Online Certificate Status Protocol (OCSP), nor other platforms (z/OS, Open VMS, etc).

It is recommended that you perform the tasks in 2 phases because it is easier to narrow down the scope of the problem determination tasks in case that there are problems:
Phase 1) Connect your MDB in WAS using a non-SSL connection with the MQ queue manager.
Phase 2) Once the MDB is able to receive messages successfully, then you can configure the connection to add SSL.

For the Phase 1, the Sample MDB and deployment and testing instructions mentioned in the following techdoc were performed successfully (non SSL connection).

IBM Techdoc: 7016505
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505
Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7

The Sample MDB is a small but fully functional MDB which is very helpful for testing the connection between WAS and MQ. If the message that is placed in the queue has the text "TESTING", then the MDB will write in the WAS SystemOut.log the following:

```
+++ SAMPLE MDB: Text Message => TESTING
```

This document covers all the necessary steps for "Phase 2", in which the successful non-SSL connection is transformed into an SSL connection.

This document concentrates on Activation Specifications, which is the preferred mechanism in WAS v7. For completeness, information is provided also for Listener Ports which use information from Connection Factories.

The documentation mentioned in the "References" section provide excellent background on what is SSL but these resources do not offer a comprehensive step-by-step procedure that you can easily follow. Thus, the purpose of this techdoc is to fill the gap between the "theory" of those references and the "practice".

+++ Configuration

The software used for this techdoc is listed below:

WebSphere Application Server 7.0.0.17
WebSphere MQ 7.0.1.5
Both products running on an SLES 11 x86 machine

+++ Recommended SupportPacs regarding SSL

There are 2 SupportPacs related to SSL which are recommended by the following MQ consultant:

T.Rob Wyatt, Senior Managing Consultant, from IBM, mentions the following 2 SupportPacs regarding SSL in his article.
http://www.ibm.com/developerworks/websphere/techjournal/0909_mismes/0909_mismes.html
Mission: Messaging: Ten WebSphere MQ SupportPacs I can't live without

In the above article, T.Rob mentions the following about 2 SupportPacs:

< begin quote >

MH03: WebSphere MQ SSL Configuration Checker
http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24014179

This SupportPac will examine the SSL configuration settings for a queue manager and, optionally, for a client as well, and report on any issues that it finds. There is no complicated installation to perform, just drop the self-contained executable on the host where the queue manager lives and run it. The program comes compiled for AIX®, HP-UX, Linux®, Solaris™, and Windows.
When the program finds an issue, a detailed report is printed, which typically includes a short description of the problem, an "advice" section with a more detailed description of the problem, suggested possible resolutions, and any known exceptions.

MO04: WebSphere MQ SSL Wizard
http://www-1.ibm.com/support/docview.wss?uid=swg24010367

The MO04 SSL Wizard is one of my "most used" SupportPacs that I don't actually use that much. The way I use it these days is to give it to someone who is just getting started learning WebSphere MQ SSL so that they can learn from it.
MO04 is a Java-based GUI that walks you through an interview process to collect the requirements for connecting two queue managers (or a client and a queue manager) over SSL-enabled channels. Once the details for both sides of the SSL connection are collected, the SSL Wizard generates a very comprehensive process, including both narrative description and the necessary commands. The commands include the actual values for things like the queue manager name, channel names, and certificate details, and are intended to be run as-is.
The output produced is extremely helpful for understanding the process and answering questions like, am I supposed to export or extract that certificate?

< end of quote >

+++ Resources

The following resources provide excellent background information on SSL and the overall tasks to be done.

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp
WebSphere MQ Information Center Version 7

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp
WebSphere Application Server Information Center Version 7

+++ Table of Contents

The chapters of this document are:

++
+++ Chapter 1: Configuration for MQ - create queue manager and objects
++

Create a Queue Manager, channels, and queue. The Queue Manager and its objects can be created using the WebSphere MQ Explorer GUI or by using Control commands and the MQSC shell (runmqsc).

a. Create a Queue Manager with the name of your choosing using the control command crtmqm. It is a good practice to define a Dead Letter Queue.

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM_MDB
```

NOTE: This command creates the Queue Manager with the default values for the size and number of primaries/secondaries of the Queue Manager transaction logs. Please review the WeSphere MQ Information Center for more details and discuss with your team the messaging load to optimize the configuration.

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.amqzag.doc/fa15650_.htm
Command: crtmqm

b. Start the Queue Manager.

```
strmqm QM_MDB
```

c. Open the MQSC shell.

```
runmqsc QM_MDB
```

Note: For information regarding MQSC commands please review the following page.
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaj.doc/sc10340_.htm
The MQSC commands

d. For the non-SSL connection, this default server connection channel will be used. This command will display its attributes:

```
  DISPLAY CHANNEL(SYSTEM.DEF.SVRCONN)
```

e. For the SSL connection, define a server connection channel with a Cipher Specification to enable SSL.

A strong encryption is recommended. In addition we recommend that you implement client authentication as well.

```
DEFINE CHANNEL('WAS_SVRCONN') CHLTYPE(SVRCONN) TRPTYPE(TCP) +
  SSLCIPH(TRIPLE_DES_SHA_US) SSLCAUTH(REQUIRED)
```

f. Define a local queue.

```
DEFINE QLOCAL(Q_MDB)
```

g. Define and start an MQ Listener:

```
DEFINE LISTENER(TCP.LISTENER) TRPTYPE(TCP) CONTROL(QMGR)
PORT(1420)

START LISTENER(TCP.LISTENER)
```

h. (Optional but very useful!) Define a channel to be used with the MQ Explorer:

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

i. Exit runmqsc:

```
 END
```

++
+++ Chapter 2: Configuration for WAS - non-SSL connection and deploy Sample MDB
++

This chapter shows the main steps for configuring the JMS administrative objects stored in the JNDI from WAS, for deploying the Sample MDB and for testing it.
This chapter deals with a non-SSL connection and it provides a baseline for testing.

The full steps are described in the mentioned technical document:
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505
Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7

Here is a summary:

Login as a WAS Administrator.

Use the WebSphere Application Server Administrative Console with server: server1
http://localhost:9060/ibm/console/unsecureLogon.jsp

Create the following JMS objects:

Connection Factory:
Name: SampleMDBConnectionFactory
JNDI Name: jms/SampleMDBConnectionFactory
Transport: Bindings, then client
Hostname: localhost
Port: 1420
Server connection channel: SYSTEM.DEF.SVRCONN

Destination: Queue
Name: SampleMDBQueue
JNDI Name: jms/SampleMDBQueue
Queue Name in MQ: Q_MDB
Queue Manager: QM_MDB

Activation Specification for a Queue:
Name: SampleMDBQueueActivationSpec
JNDI Name: jms/SampleMDBQueueActivationSpec
Destination JNDI name: jms/SampleMDBQueue
Destination Type: Queue
Queue manager: QM_MDB
Transport : Bindings, then client
Hostname: localhost
Port: 1420
Server connection channel: SYSTEM.DEF.SVRCONN

Create a Listener Port for a Queue:

Name: SampleMDBQueueLP
Initial State: Started
Connection factory JNDI name: jms/SampleMDBConnectionFactory
Destination JNDI name: jms/SampleMDBQueue

Note:
Any additions/deletions to the JNDI directory service of the App Server require a reboot of the server: thus, logout from the Administrative Console.

Then login as root and stop and restart the server:

```
stopServer.sh server1

startServer.sh server1
```

Look at the SystemOut.log and SystemErr.log files to see if there are any warnings or error related to the JMS objects or the listener ports that we defined. The actual location of these logs will depend upon the directories chosen during the installation of your WebSphere Application Server. In our example these logs are found at the following location.
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1

Proceed to deploy the sample EAR:
SampleMDBEJB.ear

The result is that the following EJB, which as an MDB, will be deployed:
SampleMDBEJBEAR

From the console, in the left panel, select:
Applications > Application Types > WebSphere enterprise applications

Ensure that the Activation Specification is used instead of the default Listener Port:

Go to the screen: Enterprise Applications > SampleMDBEJBEAR > Message Driven Bean listener bindings
Uncheck “Listener Port”.
Check “Activation Specification” and specify the Activation Spec for Queues.
Targer Resource JNDI Name: jms/SampleMDBQueueActivationSpec

Click on OK then click on Save.

From the “Enterprise Applications” window, select “SampleMDBEJBEAR” and click on the box to the left of the name.

Then click “Start”.

++ Testing the MDB (using a non-SSL connection)

Let's test the MDB. We will need 2 UNIX command prompt windows:

+ Window 1: One for watching the recent entries in the WebSphere Application Server SystemOut.log. We can watch for the output from the MDB that indicates that a message was received.

+ Window 2: The other for entering an MQ command that places one message into the queue that is monitored by the Activation Specification and which passes the message to the MDB.

Step A: From Window 1:
Change to the directory where the WebSphere Application Server server logs reside. In this case it is:

```
cd /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1
```

Issue the command to watch constantly the recent lines into the SystemOut.log file.

```
tail –f SystemOut.log
```

Step B: From Window 2:
Login as user "mqm" (or another user who has access to MQ).
Enter the command to put a message into the queue Q_MDB from the queue manager QM_MDB.

```
amqsput Q_MDB QM_MDB
```

Enter a text that you could easily identify from the SystemOut.log, such as:
TESTING MDB
Press Enter to end amqsput.

Step C: From Window 1:
Notice the text at the bottom of the SystemOut.log file:

+++ SAMPLE MDB: Text Message => TESTING MDB

Press Ctrl-C to end the "tail" command on the SystemOut.log.

Now that the non-SSL connection was successfully tested, we can proceed to perform the necessary configuration steps to enable an SSL connection.

++
+++ Chapter 3: Configuration for MQ - create key database and certificates
++

++ Server connection channel enabled for SSL

We need to have a server connection channel that is enabled for SSL.
The following channel was defined earlier.
A strong encryption is recommended.
In addition we recommend that you implement client authentication as well.

```
DEFINE CHANNEL('WAS_SVRCONN') CHLTYPE(SVRCONN) TRPTYPE(TCP) +
  SSLCIPH(TRIPLE_DES_SHA_US) SSLCAUTH(REQUIRED)
```

+++ Create key database and certificates

Create a key database and certificates for the queue manager using the Global Security Kit (GSKit) provided with the WebSphere MQ product.

a. Login as user "mqm".

Note: It is mandatory to perform these tasks as user "mqm". If you login as another MQ Administrator (belonging to the "mqm" group) and create the key database and related files, you will get runtime errors during the SSL handshake due to required files not owned by the userid "mqm".

b. Set the environment variable JAVA_HOME either in the profile or at the session level.

For Linux is:

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

Consult:
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.amqzag.doc/fa16120_.htm
Preparing to use the gsk7cmd and gsk7capicmd commands

AIX®:	export JAVA_HOME=/usr/mqm/ssl/jre
HP-UX:	export JAVA_HOME=/opt/mqm/ssl/jre
Linux:	export JAVA_HOME=/opt/mqm/ssl/jre
Solaris:	export JAVA_HOME=/opt/mqm/ssl

Windows:
set PATH=%PATH%;C:\Program Files\IBM\gsk7\bin;C:\Program Files\IBM\gsk7\lib

c. Create a key database for the queue manager by using gsk7ikm to launch the GUI or use the gsk7cmd (or gsk7capicmd) command.

Note: Due to the large amount of text required by the GSKit commands and the limitation on the width of this page, the actual command is shown in 2 or more lines, but you must enter the whole text in a single command:

```
gsk7cmd -keydb -create -db /var/mqm/qmgrs/QM_MDB/ssl/key.kdb
  -pw passw0rd -type cms -expire 365 -stash
```

NOTES:
- The WebSphere MQ queue manager requires a key database of type CMS.
- The password for this database must be stashed (-stash option).
- A password expiration is recommended and the value is in number of days.
- You can create the key database in the directory of your choosing but the default path is: /var/mqm/qmgrs/QMgrName/ssl

For more information regarding key database creation and certificate management please review the following article.
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzas.doc/sy11560_.htm
Setting up communications for SSL or TLS on UNIX systems or Windows

d. Generate a self-signed certificate for the Queue Manager. Self-signed certificates contain both the public and private certificate keys.
This certificate must be labeled with the format:
ibmwebspheremq**queuemanagername**
Notice that the name of the queue manager MUST be in lowercase. In this case the queue manager name is QM_MDB and thus, the label is:
ibmwebspheremq**qm_mdb**

Remember that although the command is shown here using 3 lines, you must enter all the tokens in a single line:

```
gsk7cmd -cert -create -db /var/mqm/qmgrs/QM_MDB/ssl/key.kdb
 -pw passw0rd  -label ibmwebspheremqqm_mdb
 -dn "CN=QM_MDB,O=IBM,C=US,OU= MQ Support,ST=North Carolina"
 -size 1024
```

e. Run the following command to list the certificates in the key database:

```
gsk7cmd -cert -list -db /var/mqm/qmgrs/QM_MDB/ssl/key.kdb
  -pw passw0rd
```

NOTE: Self-signed certificates should only be used for testing and is not intended to be used in production environments. Please use a Certificate Authority, such as VeriSign, in order to maximize security.

f. Extract the "signer certificate" from the self-signed certificate into a file. The signer certificate contains the public key and is distributed to trusted parties for authentication.

You can label the signer certificate with a name of your choosing.

```
gsk7cmd -cert -extract -db /var/mqm/qmgrs/QM_MDB/ssl/key.kdb
  -pw passw0rd -label MQ_Signer_Cert
  -target /var/mqm/qmgrs/QM_MDB/ssl/qm_mdb_signer_cert.arm
```

g. Use ftp to copy the MQ Signer Certificate, "qm_mdb_signer_cert.arm" to the host where the WAS Server is located. In our example we placed the file into the following location:

/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/veracruzNode01Cell/nodes/veracruzNode01/qm_mdb_signer_cert.arm

The MQ queue manager Signer Certificate must be added to the WAS TrustStore in order for the WAS JMS client to authenticate the MQ queue manager.
In the same way, the signer certificate from the WAS Server must be sent to the MQ queue manager so that the WAS Signer Certificate can be added to the queue manager key database for client authentication to succeed.

h. Add the MQ Signer Certificate in the WAS TrustStore.
This step is described later on. See in Chapter 3:

f. Add the MQ Signer Certificate into the WAS TrustStore.

++
+++ Chapter 4: Configuration for WAS - create certificate stores and certificates
++

Login as root.

Create two certificate stores:
1) A KeyStore to contain the personal certificates and
2) A TrustStore to contain the signer certificates.

You can launch the Global Security Kit (GSKit) IKeyman GUI by issuing gsk7ikm command or use the gsk7cmd commands.

a. Create a KeyStore of type "jks" in the directory of your choice.
To make identification easier, the KeyStore was named WASKeyStore.jks

```
gsk7cmd -keydb -create -db /path/WASKeyStore.jks
  -pw passw0rd -type jks
```

The full path actually used in our test is:
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/veracruzNode01Cell/nodes/veracruzNode01/WASKeyStore.jks
… but for short, it will be denoted as: /path/WASKeyStore.jks

b. Generate a self-signed certificate for the WAS Server. This certificate can be labeled with a name of your choosing.

```
gsk7cmd -cert -create -db /path/WASKeyStore.jks -pw passw0rd
   -label WAS_Personal_Cert
   -dn "CN=server1,O=IBM,C=US,OU=Support,ST=North Carolina"
   -size 1024
```

c. Extract the "signer certificate" from the self-signed certificate into a file.

```
gsk7cmd -cert -extract -db /path/WASKeyStore.jks -pw passw0rd
  -label WAS_Signer_Cert -target /path/was_signer_cert.arm
```

d. Create a TrustStore of type "jks" in the directory of your choice.
For easy identification the TrustStore was named WASTrustStore.jks.

```
gsk7cmd -keydb -create -db /path/WASTrustStore.jks
  -pw passw0rd -type jks
```

The full path actually used in our test is:
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/veracruzNode01Cell/nodes/veracruzNode01/WASTrustStore.jks

… but for short, it will be denoted as: /path/WASTrustStore.jks

e. Add the WAS Signer Certificate into the WAS TrustStore.

```
gsk7cmd -cert -add -db /path/WASTrustStore.jks
  -label WAS_Signer_Cert
  -file /path/was_signer_cert.arm -pw passw0rd
```

Note: In case that you need to remove the above certificate (for example, in case of a typo and you need to replace the certificate and redo the 'add' operation), this is the command:

```
gsk7cmd -cert -delete -db /path/WASTrustStore.jks
  -label WAS_Signer_Cert
  -pw passw0rd
```

f. Add the MQ Signer Certificate into the WAS TrustStore.

Note: The MQ Signer Certificate is the one that was copied via ftp in the previous chapter:

g. Use ftp to copy the MQ Signer Certificate, "qm_mdb_signer_cert.arm" to the host where the WAS Server is located.

```
gsk7cmd -cert -add -db /path/WASTrustStore.jks
  -label MQ_Signer_Cert
  -file /path/qm_mdb_signer_cert.arm -pw passw0rd
```

g. Ftp the WAS Signer Certificate to the MQ Server, into the directory:
/var/mqm/qmgrs/QM_MDB/ssl/

h. On the host of the MQ Server add the WAS Signer Certificate to the Queue Manager's key database.

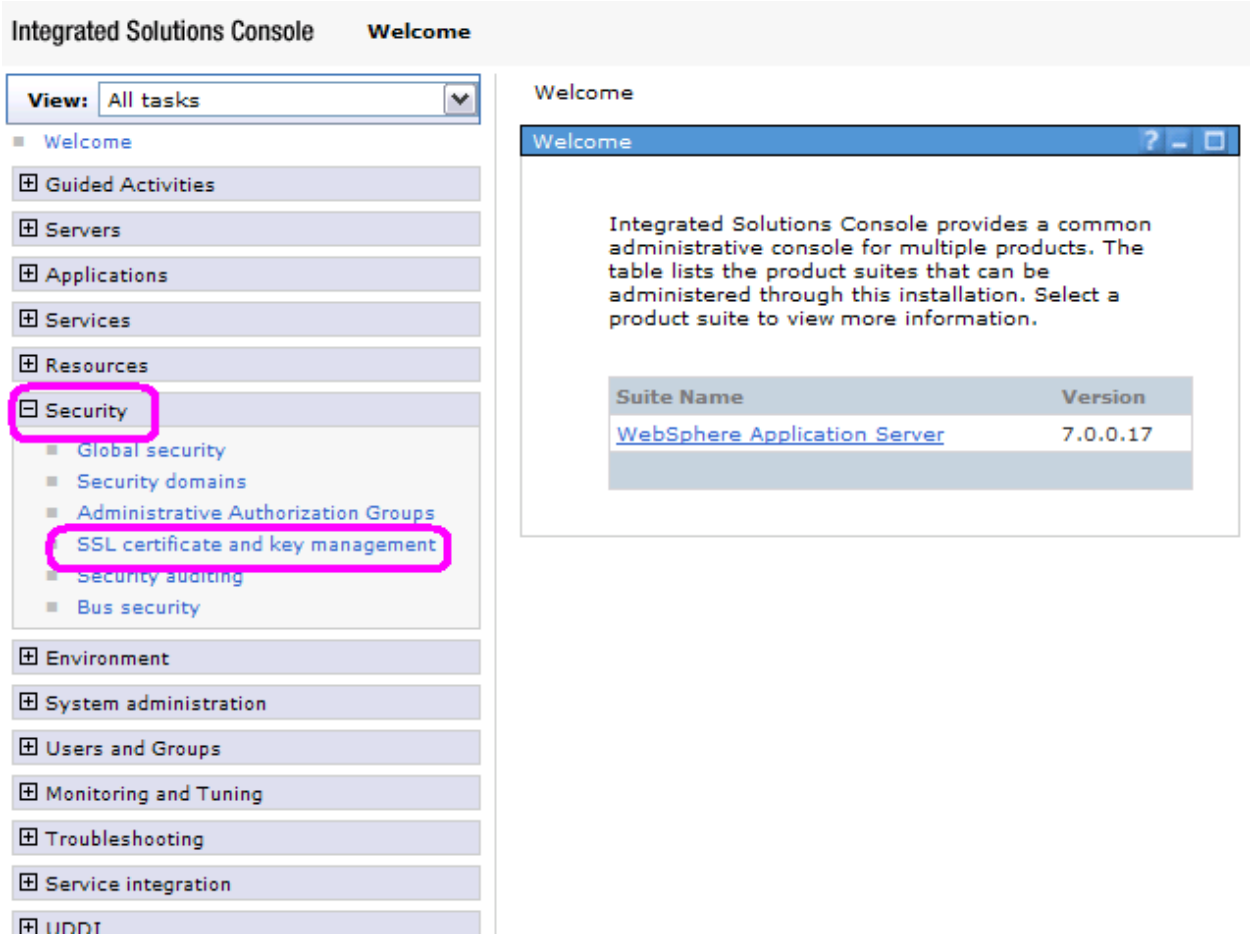
```
cd /var/mqm/qmgrs/QM_MDB/ssl

gsk7cmd -cert -add -db /var/mqm/qmgrs/QM_MDB/ssl/key.kdb
  -label WAS_Signer_Cert -file /path/was_signer_cert.arm
  -pw passw0rd
```

++
+++ Chapter 5: Configuration for WAS - server SSL configuration
++

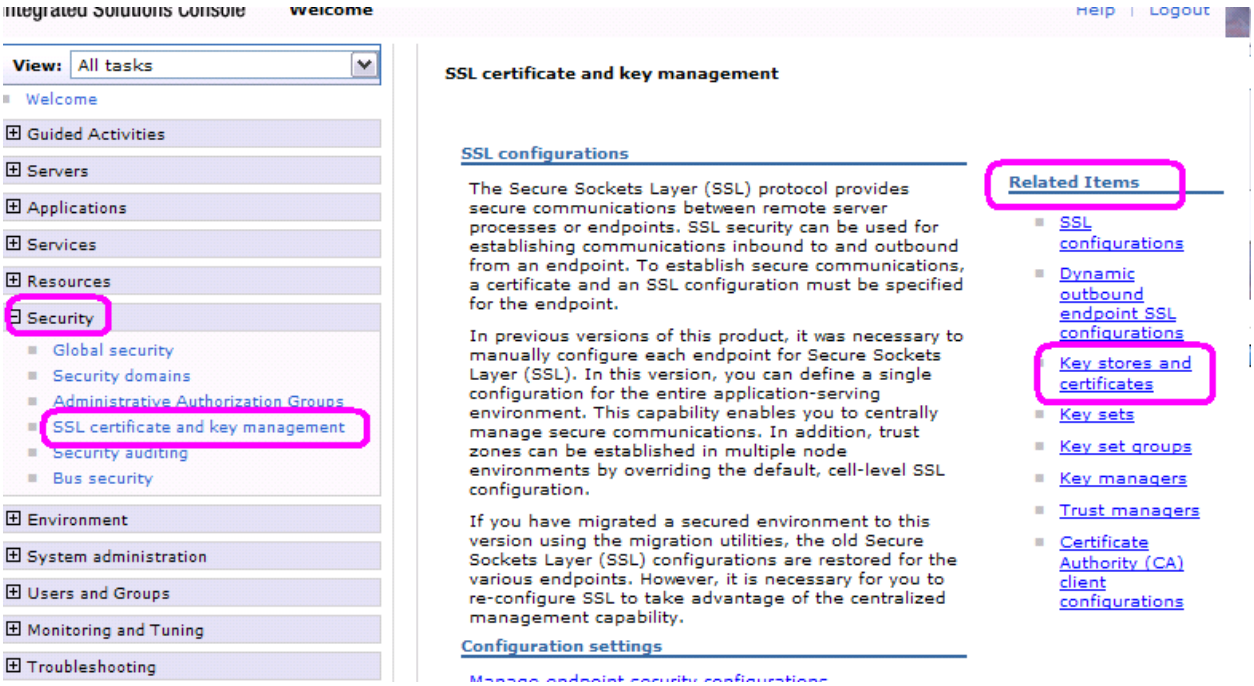
++ Section 1: Configure SSL Certificate Stores

a. Login to the WAS Administrative Console.

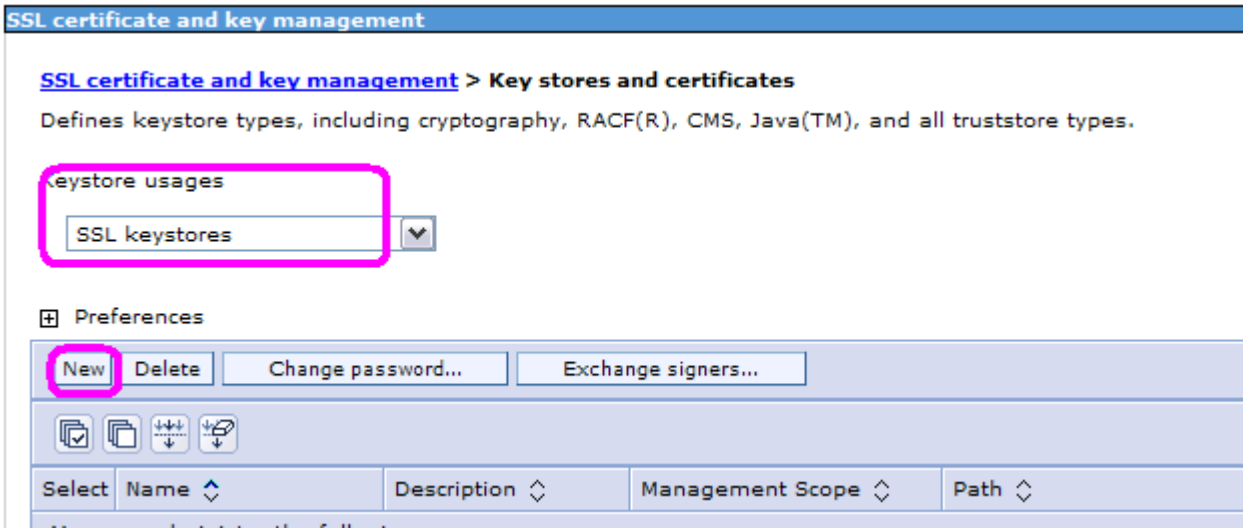
b. Open Security -> SSL certificate and key management



c. Under "Related Items" select Key stores and certificates.

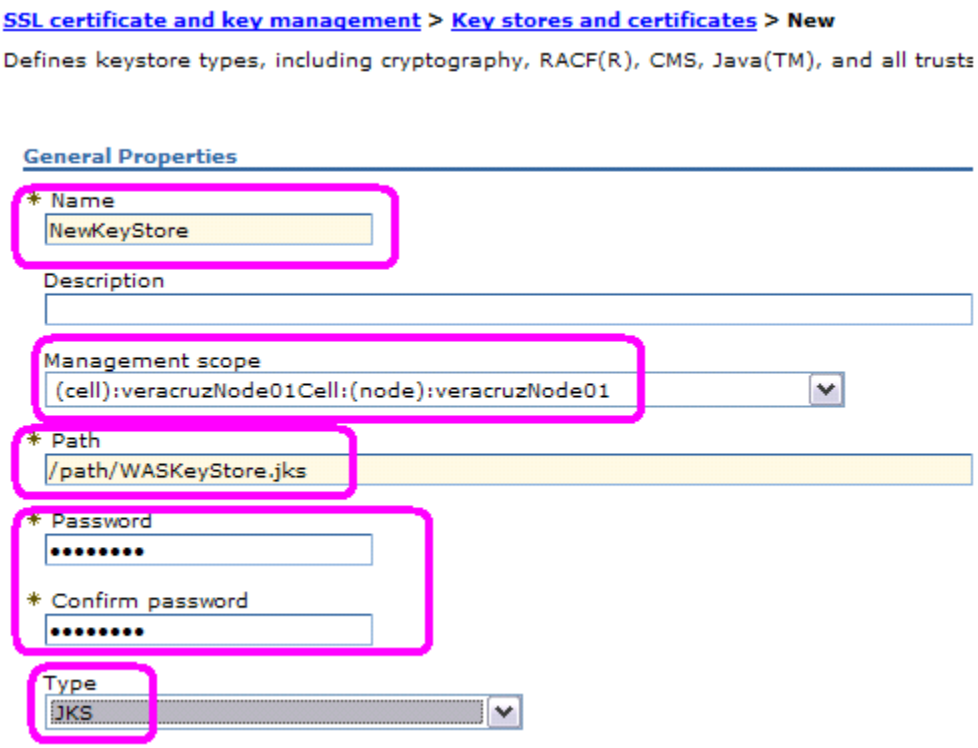


You will see the following window.
Notice that the default "Keystore usages" is: SSL keystores



d. Click the "New" button and fill out the following fields for the KeyStore:

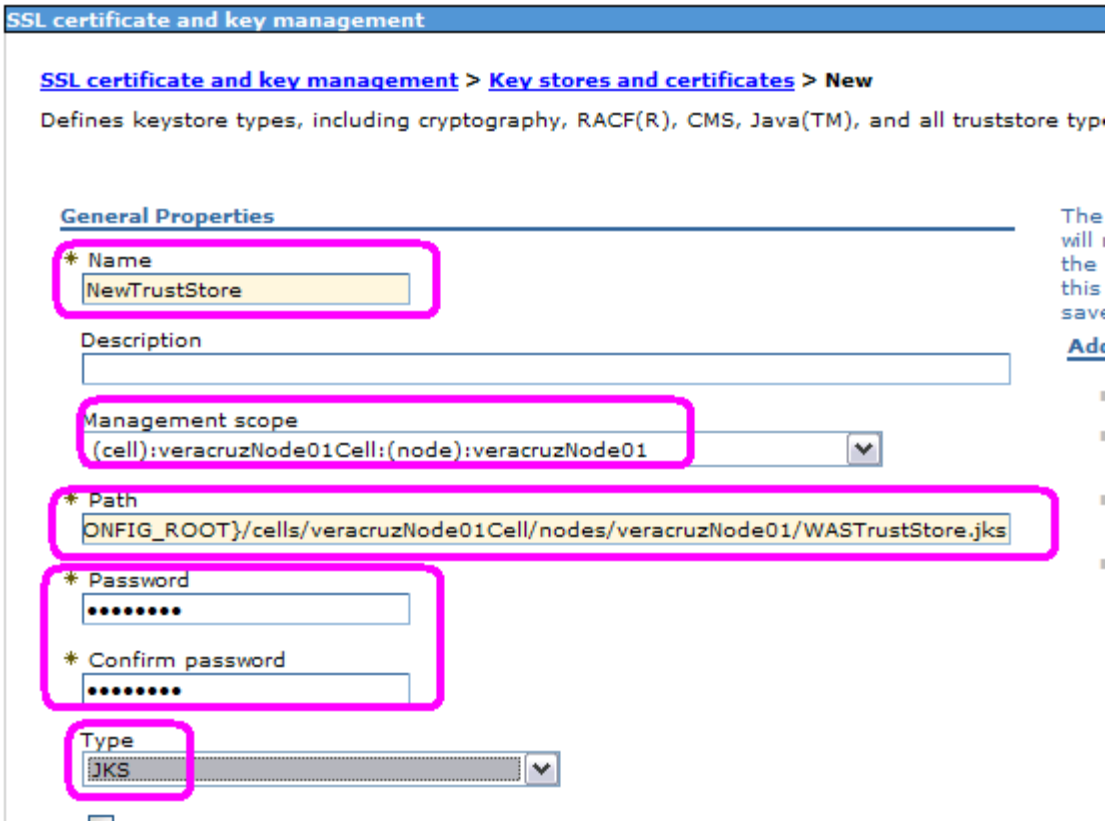
Field	Value
Name	NewKeyStore (name of your choosing)
Management scope	(Select proper scope from drop down menu, such as node)
Path	/path/WASKeyStore.jks
Password	passw0rd
Confirm password	passw0rd
Type:	JKS



Click OK to save configuration to local the master configurations.

e. Repeat the operation for the TrustStore. Click the "New" button and fill out the following fields for the TrustStore:

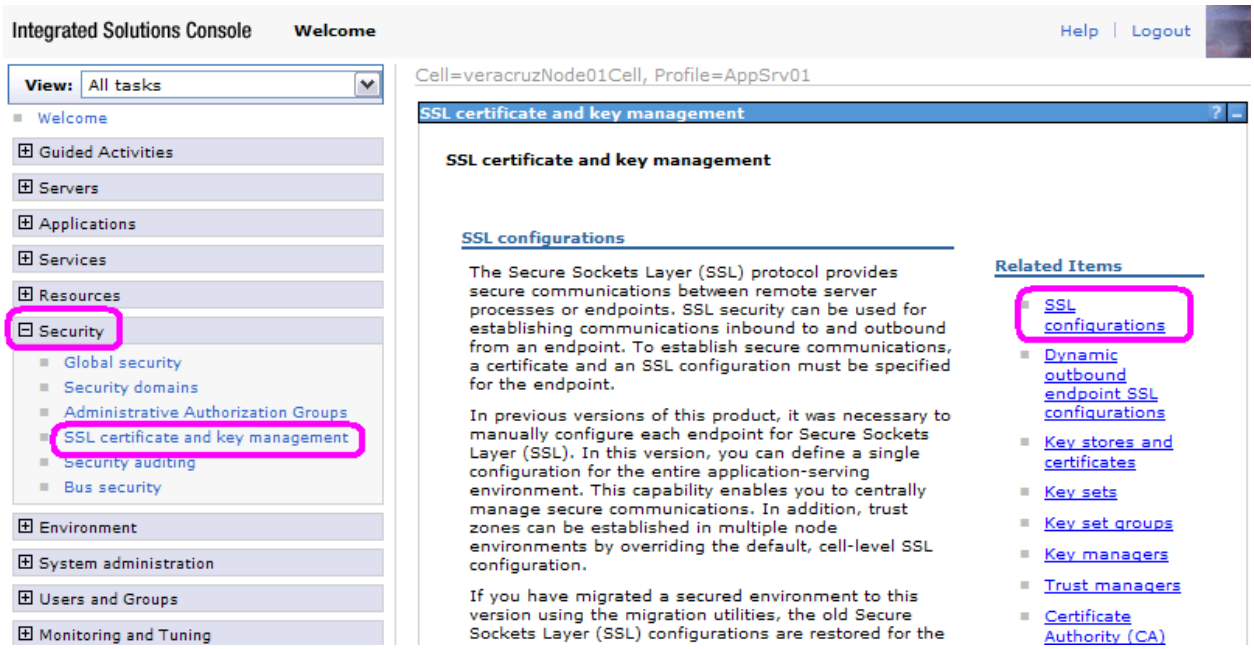
Field	Value
Name	NewTrustStore (name of your choosing)
Management scope	(Select proper scope from drop down menu, such as node)
Path	/path/WASTrustStore.jks
Password	passw0rd
Confirm password	passw0rd
Type:	JKS



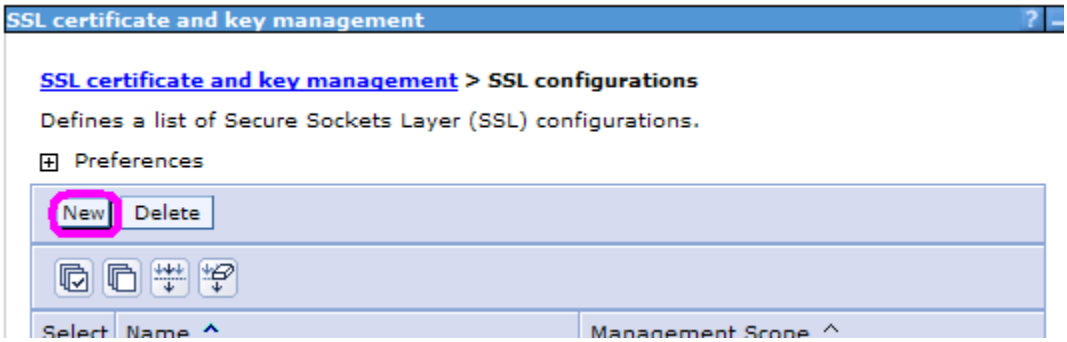
Click OK to save configuration to local the master configurations.

++ Section 2. SSL Configuration

a. Open Security -> Select SSL certificate and key management.
Under "Related Items" select SSL configurations

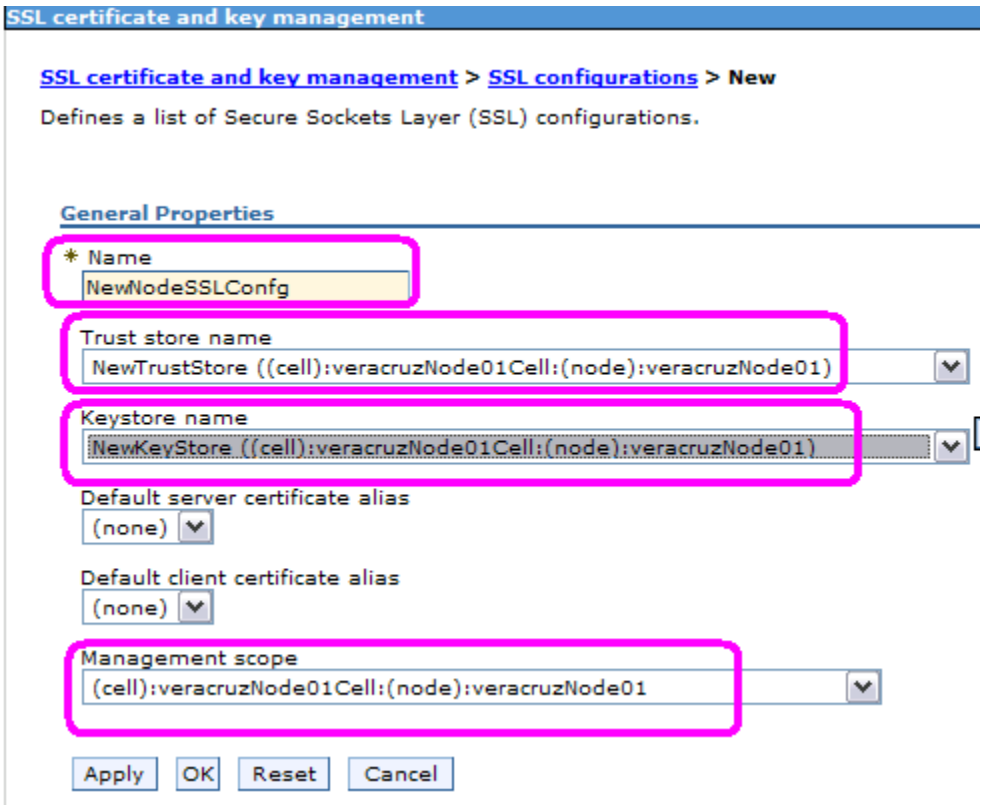


You will see a new panel:



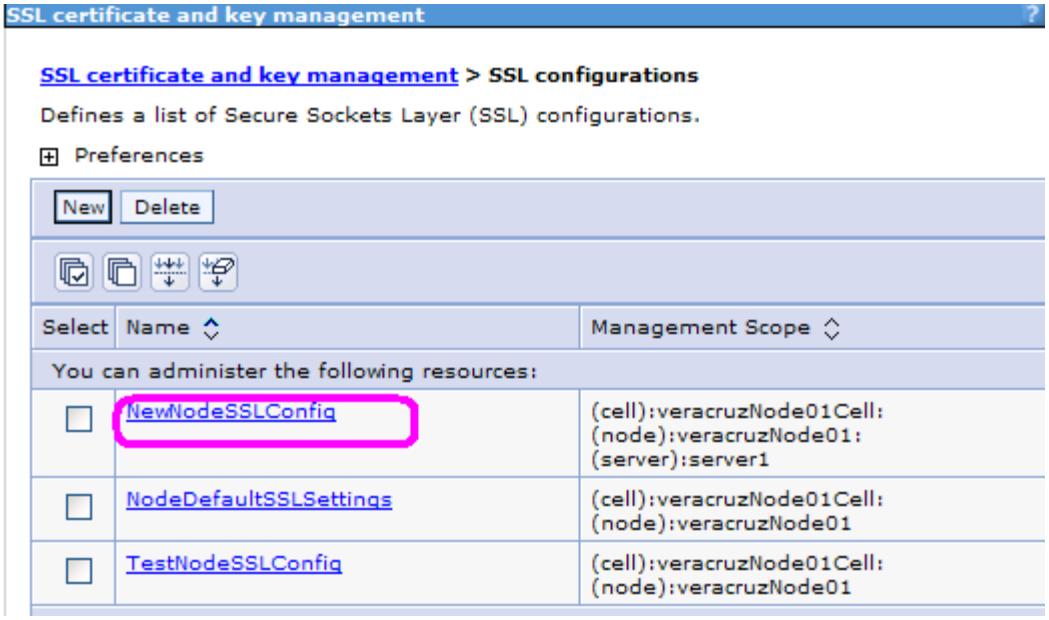
b. Click the "New" button and fill out the following fields:

Field	Value
Name	NewNodeSSLConfg (name of your choosing)
Trust store name	NewTrustStore (selected through drop down menu)
Keystore name	NewKeyStore (selected through drop down menu)
Management scope	Select scope from drop down menu



c. Click OK and Save configuration to local the master configurations.

d. Click on the name of the new SSL configuration, example NewNodeSSLConfig



You will need to scroll to the right to see the rest of the options.

e. Under "Additional Properties" Click Quality of protection (QoP) settings

f. Under "Client authentication" select None, Supported or Required, per your Security requirements. "Required" is the recommendation.

g. In Cipher suite settings under the "Cipher suite groups" select the strength of encryption from the drop down menu and click on the "Update selected Ciphers" button.

The Cipher Suites families will be populated under "Selected Ciphers"

NOTE: "You can only use one Cipher Suite in the SSL configuration for a WebSphere MQ messaging provider Connection Factory or Activation Specification. If you specify more than one Cipher Suite, ONLY the first one is used."

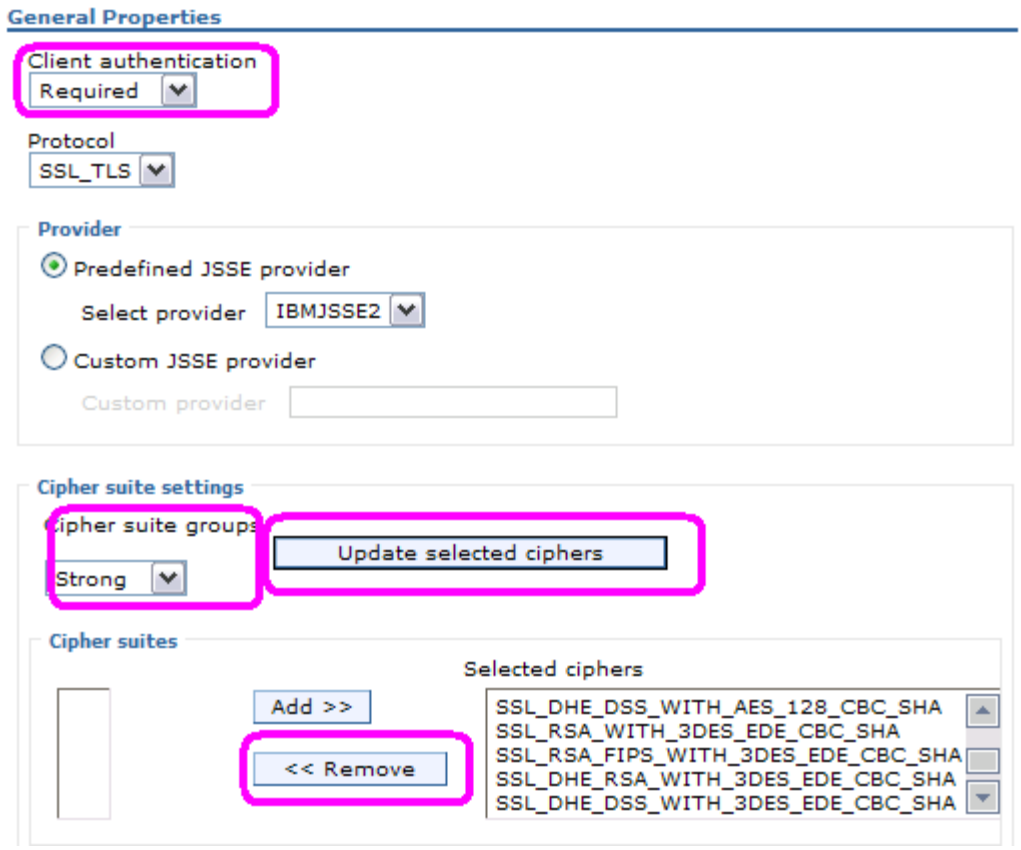
Per the encryption level configured for the server connection channel defined earlier, TRIPLE_DES_SHA_US, we chose the strong "Cipher suite groups", we kept the matching Cipher Suite called SSL_RSA_WITH_3DES_EDE_CBC_SHA and removed all others.

Please review the following article listing the Cipher Specifications and the equivalent Cipher Suites.

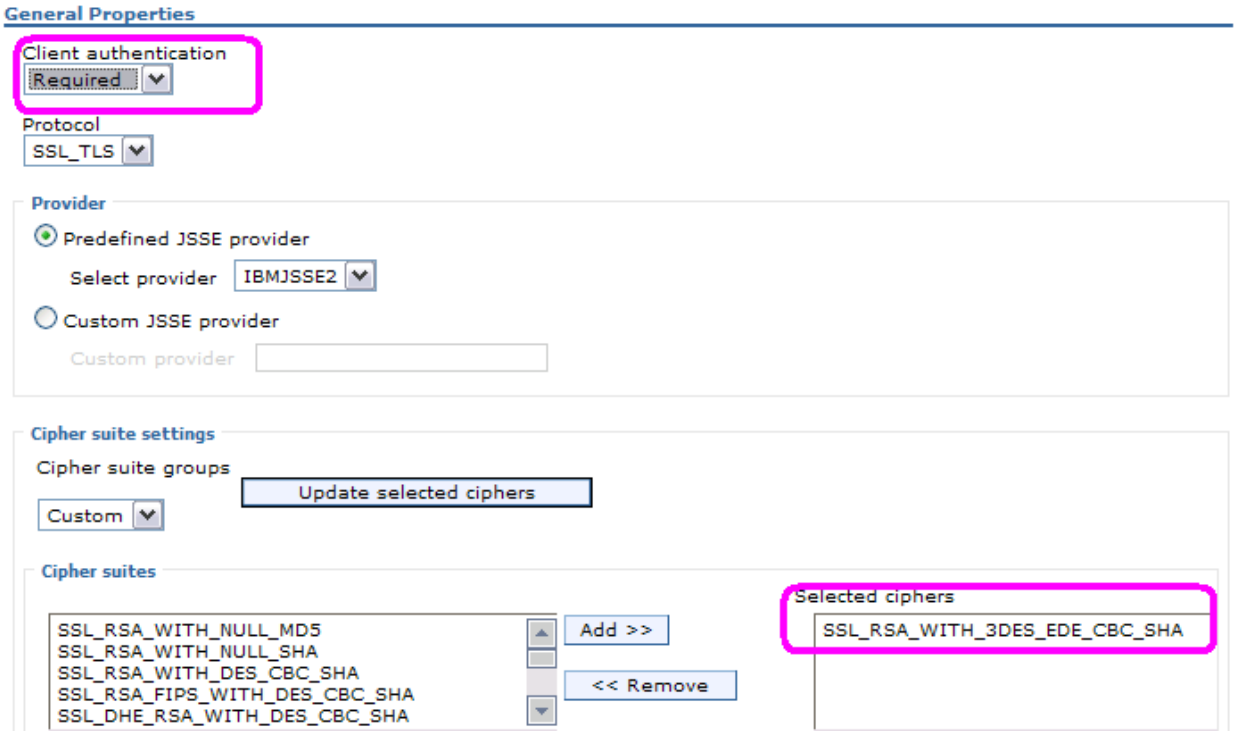
http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm34740_.htm
SSL CipherSpecs and CipherSuites

Recommendation: Per the restriction stated in the "NOTE" above, remove all Cipher Suites from the "Selected Ciphers" except the suite that will match what is configured on the Server connection channel of the WebSphere MQ Queue Manager.

This is what the original screen looked like:



After removing the other specs, the end result looks like this:



h. Click OK, to save configuration to local the master configurations.

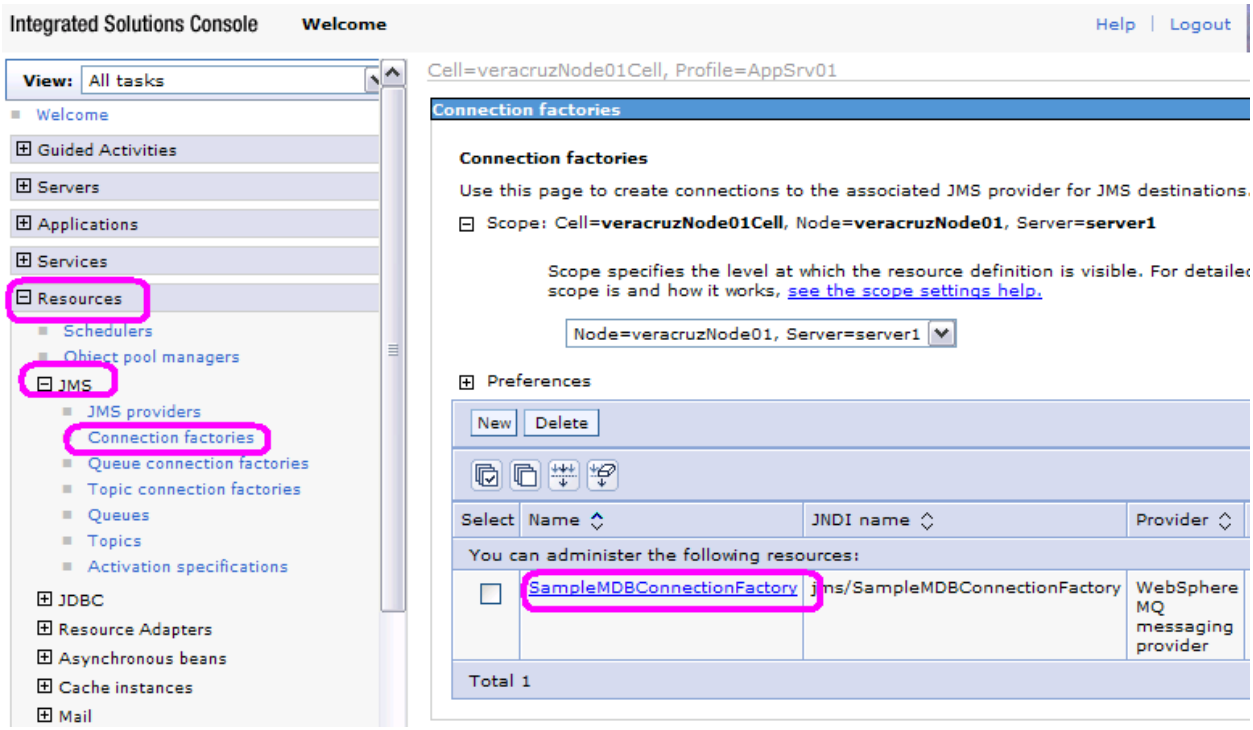
++
+++ Chapter 6: Configuration for WAS - JMS SSL configuration and Testing
++

Secure the Connection Factory and Activation Specification with SSL

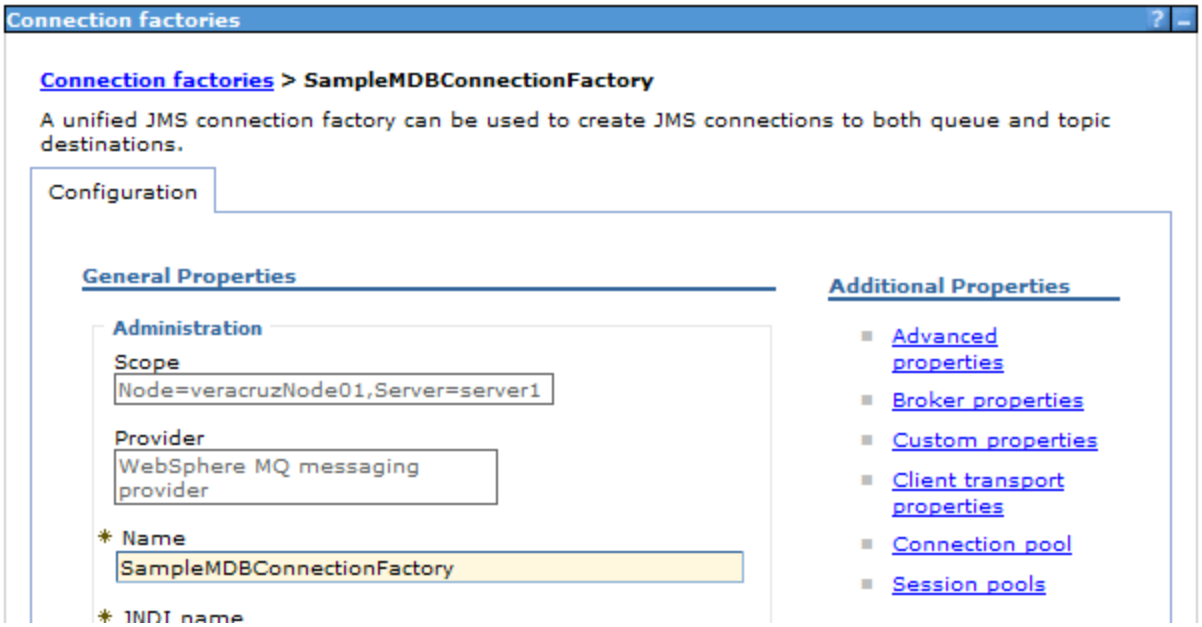
Note:
In WAS V7, the Activation Specifications are the recommended way to handle messages from MQ to pass them to the MDB.
It is possible to use Connection Factories that use Listener Ports.
Thus, we provide information on both approaches.

++ Section 1. Connection Factory

a. Open Resources -> JMS -> Connection Factories and click on the name of the Connection Factory.
In this example it is called: SampleMDBConnectionFactory

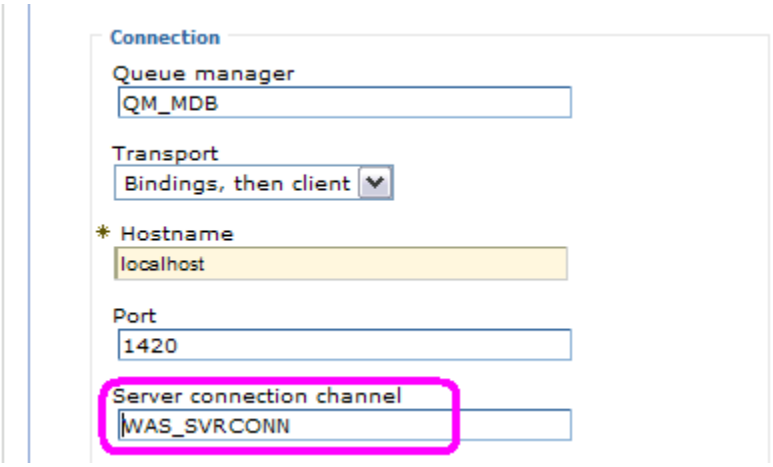


You will see a long vertical panel. You will need to scroll down to get to the SSL items.

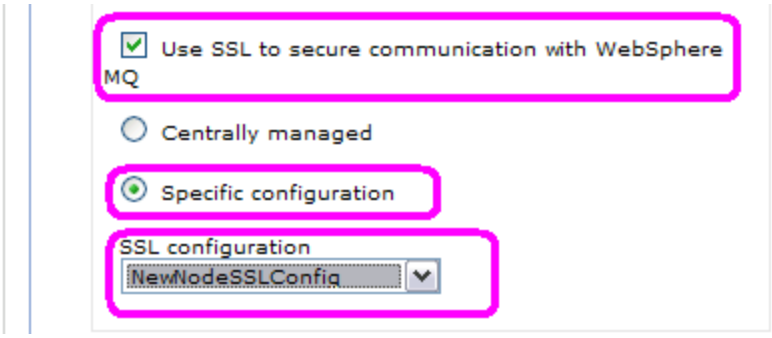


b. You need to specify the server connection channel that was defined earlier and which is enabled for SSL. In this example is called: WAS_SVRCONN

Notice that the name of the server connection channel used previously, SYSTEM.DEF.SVRCONN, did NOT use SSL.



c. Check the checkbox next to "Use SSL to secure communication with WebSphere MQ"
- Select "Specific configuration"
- Select "SSL configuration" that you created earlier. In this example is: NewNodeSSLConfig



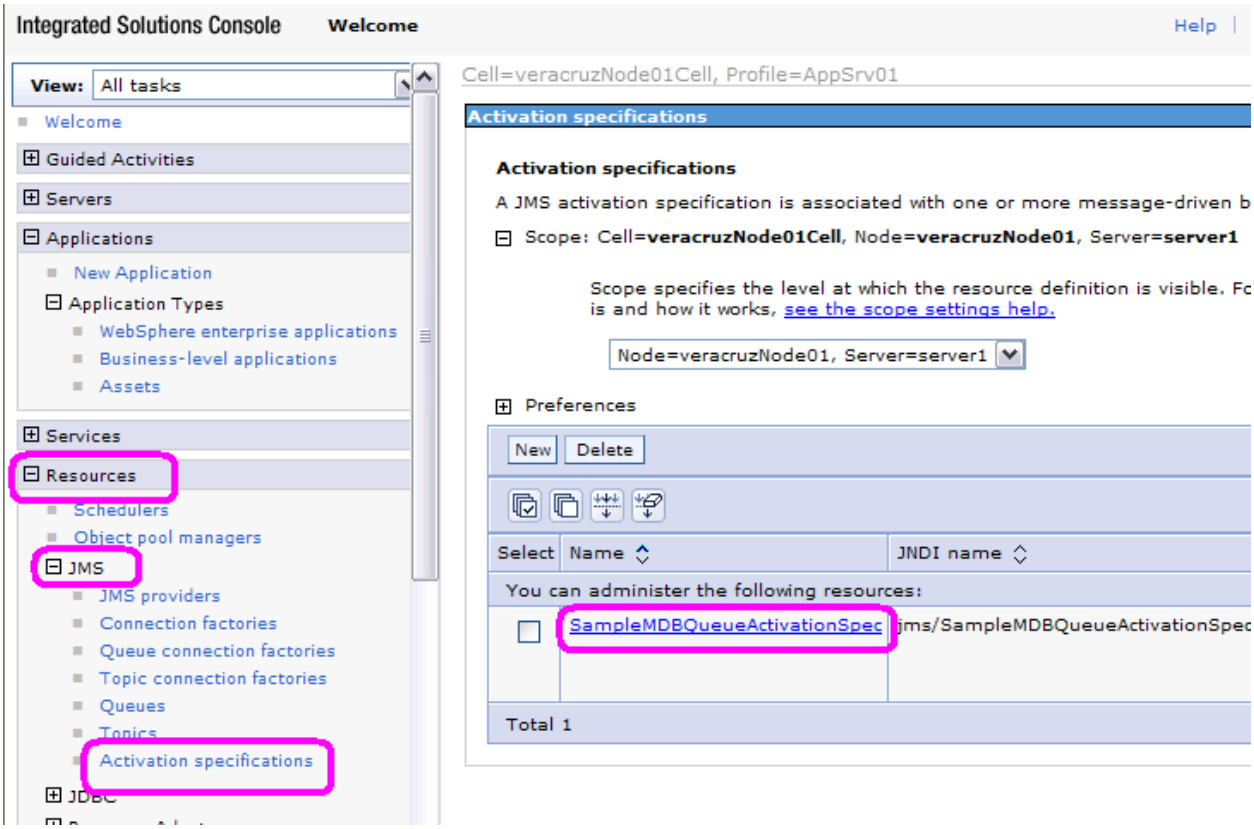
If you use a Connection Factory instead of an Activation Specification, then you will need to have a Listener Port, and associate that Listener Port to the MDB. In our example, we have:

Listener Port for Queue:
Name: SampleMDBQueueLP
Initial State: Started
Connection factory JNDI name: jms/SampleMDBConnectionFactory
Destination JNDI name: jms/SampleMDBQueue

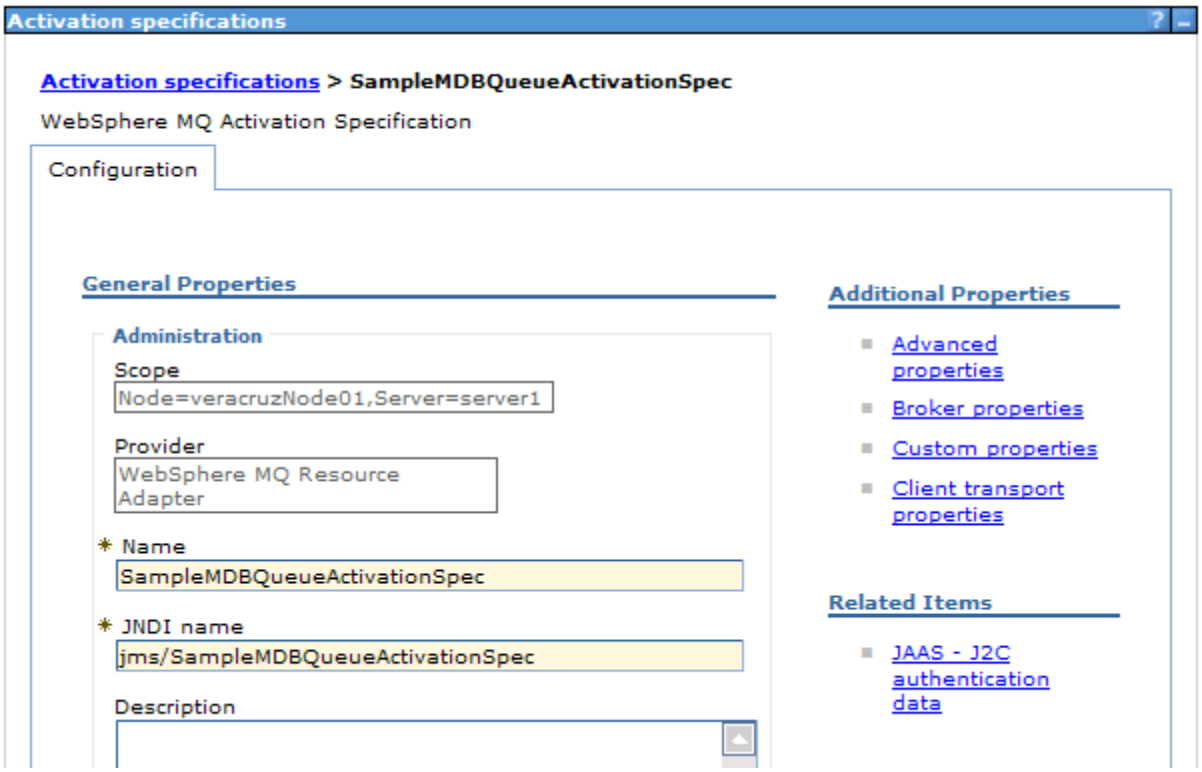
++ Section 2. Activation Specification

a. Open Resources -> JMS -> Activation Specifications and Click on the Name of the Activation Specification. In this case it is:

SampleMDBQueueActivationSpec



You will need to scroll down the long vertical panel to see the SSL related items.



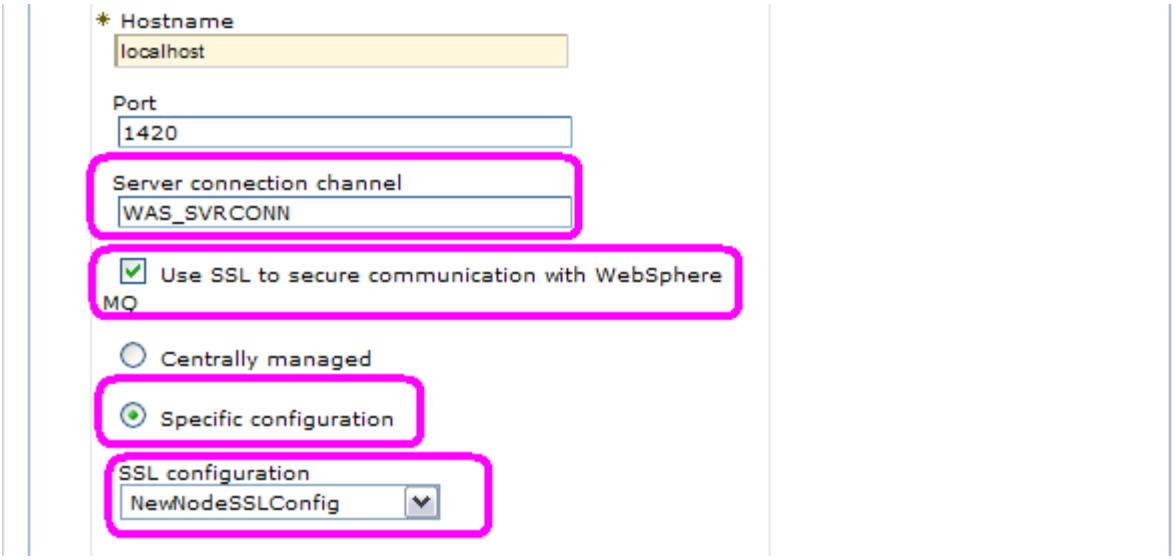
b. You need to specify the server connection channel that was defined earlier. In this example is called: WAS_SVRCONN

Notice that the name of the server connection channel used previously, SYSTEM.DEF.SVRCONN, did NOT use SSL.

c. Check the checkbox next to "Use SSL to secure communication with WebSphere MQ"

d. Select "Specific configuration"

e. Select "SSL configuration" that you created earlier.



++ Section 3: Testing the SSL connection

a. Stop and restart the WAS Server so that all changes will take effect.

b. Start the application and verify the application in a "Started" status.

c. Repeat the testing of the Sample MDB as described in Chapter 2:
    ++ Testing the MDB (using a non-SSL connection)

Window 1: put message

```
amqsput Q_MDB QM_MDB
Sample AMQSPUT0 start
target queue is Q_MDB
TESTING MESSAGE WITH SSL CONNECTION
Sample AMQSPUT0 end
```

Window 2: view bottom of SystemOut.log
[6/17/11 1:32:19:230 EDT] 00000029 SystemOut O
+++ SAMPLE MDB: Text Message => TESTING MESSAGE WITH SSL CONNECTION

The message was received by the MDB when securing the channel with SSL. Yeah!

+++ end +++