IBM Software Group

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

Miguel Rodriguez (mrod@us.ibm.com)
Angel Rivera (rivera@us.ibm.com)
WebSphere MQ Unix® Level 2 Support
28 June 2011

WebSphere® Support Technical Exchange

Agenda

- Introduction: simple scenario, no advanced features
- Using WAS 7 and MQ 7
- Self-signed certificates
- Distributed Platforms: Unix and Windows®
- Starting with baseline of non-SSL connection of MQ JMS client under WAS with MQ queue manager
- Then SSL configuration is performed in both WAS and MQ queue manager

Presenting the players (1)

- Simple example of the sequence of events and players for delivering a message from MQ V7 to an MDB in WAS V7
- The scenario is basically the same for non-SSL and SSL configurations
- Assumptions:
  - All necessary components are installed and properly configured.
  - There is an unread message in the Queue
  - There is a Listener Port or an Activation Specification active in WAS

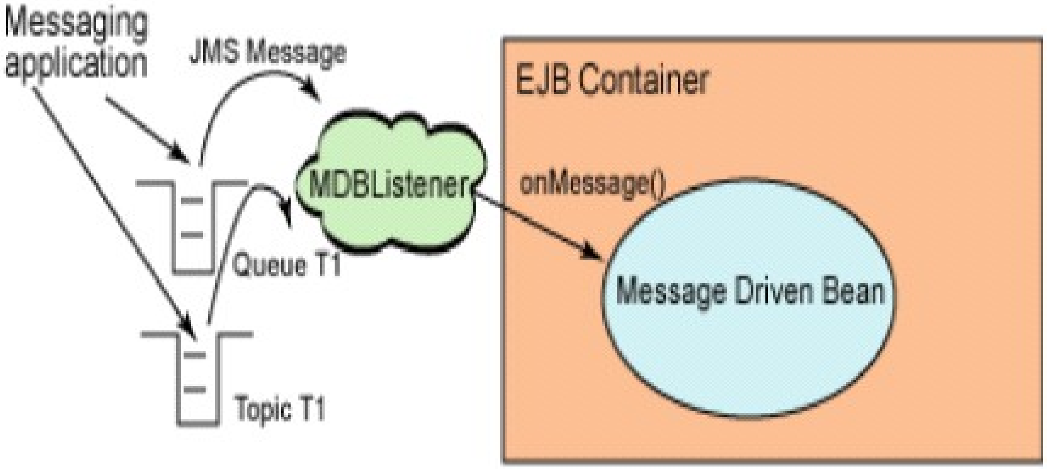
Presenting the players (2)

- WAS Server: a “building” that provides services to business tenants (EJBs): utilities, security, etc.
  - An MDB is a specialized EJB.

- MQ Queue Manager: another “building” that provides messaging/packaging services to WAS Servers and other users: assured delivery of messages and delivering a message only once (no duplicates).

Presenting the players (3)

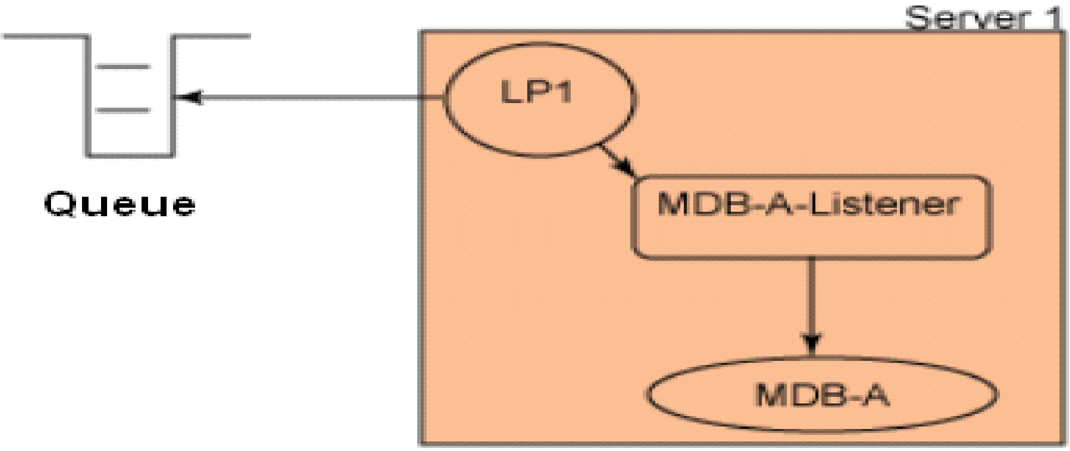
- The Enterprise Java™ Bean (EJB) occupies a “floor” in the WAS Server building and is operational.
- It has a specialized “secretary” that ONLY receives and handles messages:
  - Message Driven Bean (MDB) via the method: onMessage()
- The MDB waits until it is contacted by the Listener Port or Activation Spec



Presenting the players (4)

- The WAS Listener Port (LP) or the Activation Specification (AS) which resides in the “mail room”, delivers the messages from the mail room to the “tenants”.
- From now on, the generic LP represents both LP and AS.

- LP interacts with an MQ “representative” to get messages.



Presenting the players (5)

- If SSL is enabled, then during the startup of the Listener Port or the Activation Specification is when the SSL Handshake takes place with the MQ channel.

- For more details, consult:
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzas.doc/sy10660_.htm
- **An overview of the SSL handshake**
- The SSL handshake enables the SSL client and SSL server to establish the secret keys with which they communicate.

Presenting the players (6)

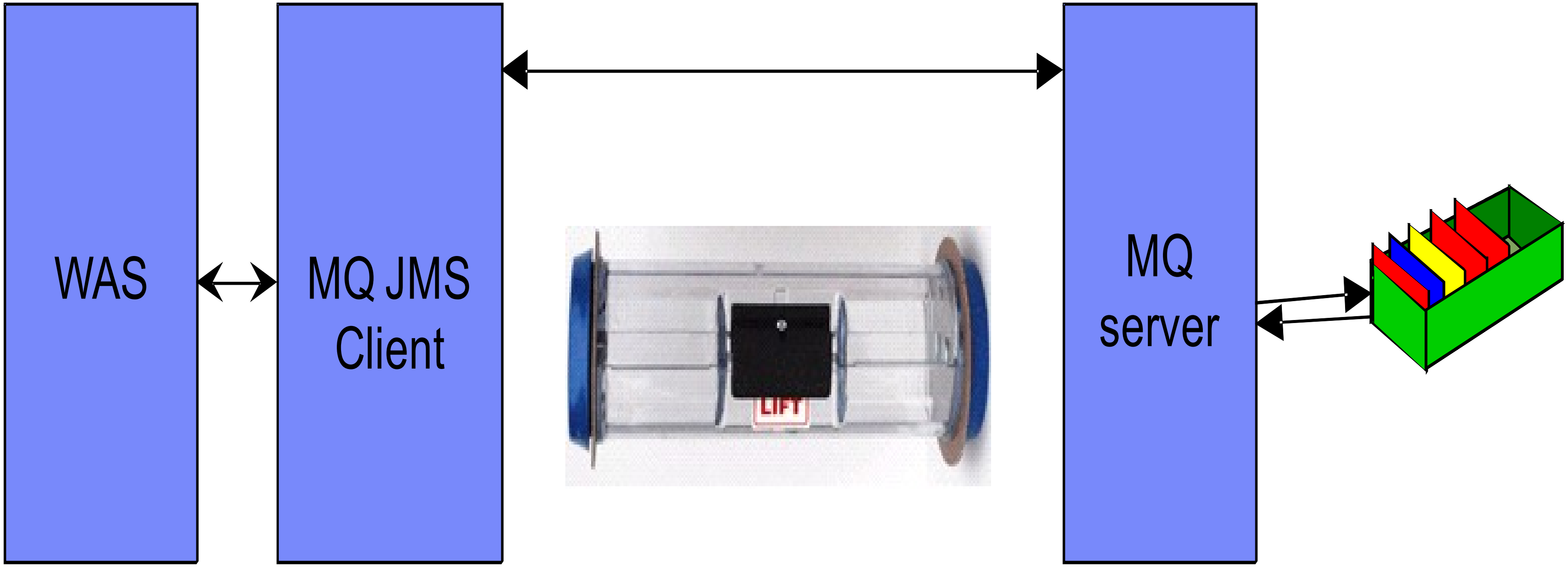
- The MQ representative is the MQ JMS Client which also resides in the mail room.

- It contacts the MQ queue manager via a server-connection channel.

- If SSL is enabled, then the whole package is encrypted (Message Descriptor and Message Contents)

Presenting the players (7)

- Let's visualize the channel as a single pneumatic tube which sends a “capsule” (like the ones used in drive-thru bank tellers)



WAS

MQ JMS Client

MQ server

Scenario: OK delivery of 1 message (1)

- The LP starts its shift by calling the MQ JMS client to ask if there are messages in the specified queue.
- LP remains on the line waiting to hear from MQ JMS client.
- The MQ JMS client writes a request into a note, puts the note in the capsule and sends it to the MQ queue manager (MQ server) via the pipe.
- The MQ server receives the capsule and reads the request.
- The MQ server checks if there is a message in the queue.
    - Yes, there is one.

Scenario: OK delivery of 1 message (2)

- The MQ server puts the copy of the message in the capsule and sends it back to the MQ JMS client.

- The MQ JMS client receives the capsule, gets the message out and tells the LP that there is a message.

- The LP goes to the mail room and gets the message, then it goes to where the MDB resides and gives the message to the MDB.

- The LP waits for confirmation from the MDB to confirm if the delivery is successful.

Scenario: OK delivery of 1 message (3)

- The MDB opens the message and reads its contents.
- If everything is fine then it handles the message and then tells the LP that everything is OK.
- The LP returns to the mail room and tells the MQ JMS client that the delivery was OK.
- In turn, the MQ JMS client tells the MQ server that the delivery was OK.
- The MQ server proceeds to destroy the message from the queue.

Sample MDB that you can experiment with - 1

- It is recommended that you start with a non-SSL connection as a baseline.
- Then you can add the SSL configuration.

- You can use a Sample MDB that you can download and experiment with.
- Such sample is provided with:

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

Sample MDB that you can experiment with - 2

- For the Non-SSL test, we use the following default server connection channel in the queue manager:
- SYSTEM.DEF.SVRCONN

- In WAS, we define JMS admin objects:

- Connection Factory: SampleMDBConnectionFactory
- Destination: Queue Q_MDB, SampleMDBQueue
- Activation Specification for a Queue:
- Name: SampleMDBQueueActivationSpec
- Destination JNDI name: jms/SampleMDBQueue
- Channel: SYSTEM.DEF.SVRCONN

Sample MDB that you can experiment with - 3

- Name of EJB: SampleMDBEJBEAR

- The onMessage() method displays the type of contents (payload) and an “eye catcher string” (+++ SAMPLE MDB) which can let you find quickly the output in the SystemOut.log file.

- If amqsput is used to enter the text: TESTING
- Then this MDB will display in the SystemOut.log

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

Techdoc that shows a complete SSL example

- We prepared a techdoc that shows a complete example of adding SSL to an established connection between a WAS V7 server with an MQ V7 queue manager.

- 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

Scenario: SSL is used

- When SSL is used, the WAS (MQ JMS client) will become the “SSL client”
- … and the MQ queue manager will be the “SSL server”

- Both will agree on which SSL Cipher Specification to use and perform the appropriate “SSL Handshake” to establish the proper credentials.

- The scenario when SSL is used is basically the same as the previous scenario, with the added step of the SSL Handshake at the beginning of the connection

SSL Configuration – what else is needed

- Certificates
  - signed personal certificate and its signer
  - for both client and server
- Keystores, to contain the certificates
  - Queue manager: type CMS
  - WAS: type JKS (at JVM level)
- Tools: GSKit commands, iKeyman GUI
- Configuration changes need at:
  - queue manager
  - WAS server, Connection Factory, Act Spec

MQ - create server connection channel SSL

- Define a server connection channel with a Cipher Specification to enable SSL.
- A strong encryption is recommended.
- We recommend that you implement client authentication as well.

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

MQ - create key database and certificates - 1

- Login as user "mqm"
- Set the environment variable JAVA_HOME
- Use gsk7ikm to launch the iKeyMan GUI or
- Use gsk7cmd / gsk7capicmd commands

- See MQ V7 Information Center:
- 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

MQ - create key database

- Create a key database for the queue manager
- Example of Queue Manager named: QM_MDB

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

- Note: The actual command is shown in 2 or more lines, but you must enter the whole text in a single command

MQ - Generate a self-signed certificate - 1

- Self-signed certificates contain both the public and private certificate keys.

- The personal certificate must be labeled with the following format:
- ibmwebspheremq**queuemanagername**

- The label MUST be written in lower-case.
- For example if the queue manager name is QM_MDB then the label must be:
- ibmwebspheremq**qm_mdb**

MQ - Generate a self-signed certificate - 2

-
  ```
  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
  ```

- To list the certificates in the key database, run:

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

MQ - Generate a self-signed certificate - 3

- Extract the "signer certificate".
- The signer certificate contains the public key and is distributed to trusted parties for authentication.

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

- A typical full path name looks like:
- /var/mqm/qmgrs/QM_MDB/ssl/qm_mdb_signer_cert.arm

MQ - Generate a self-signed certificate - 4

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

- A typical full path name of the certificate in the WAS server is:
- /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/veracruzNode01Cell/nodes/veracruzNode01/qm_mdb_signer_cert.arm

MQ - Generate a self-signed certificate - 5

- 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.

WAS - Create certificate stores - 1

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

- You can launch the Global Security Kit (GSKit) IKeyman GUI by issuing: gsk7ikm
- or use the gsk7cmd/gsk7capicmd commands.

WAS - Create certificate stores - 2

- Create a KeyStore of type "jks"
- For easy identification 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

WAS - Generate a self-signed certificate

- 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
  ```

WAS - Extract the "signer certificate"

- 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

WAS - Create a TrustStore

- Create a TrustStore of type "jks".
- 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

WAS – Add WAS Signer Cert into TrustStore

- 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: To remove a certificate issue:
-
  ```
  gsk7cmd -cert -delete -db /path/WASTrustStore.jks
  -label WAS_Signer_Cert
  -pw passw0rd
  ```

WAS – Add MQ Signer Cert into TrustStore

- Add the MQ Signer Certificate into the WAS TrustStore

- Note: The MQ Signer Certificate is the one that was copied via ftp in a previous slide.

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

MQ – Add WAS Signer Cert into MQ Key DB

- Ftp the WAS Signer Certificate to the MQ Server. In our example we placed the certificate in the keydatase directory:
- /var/mqm/qmgrs/QM_MDB/ssl/

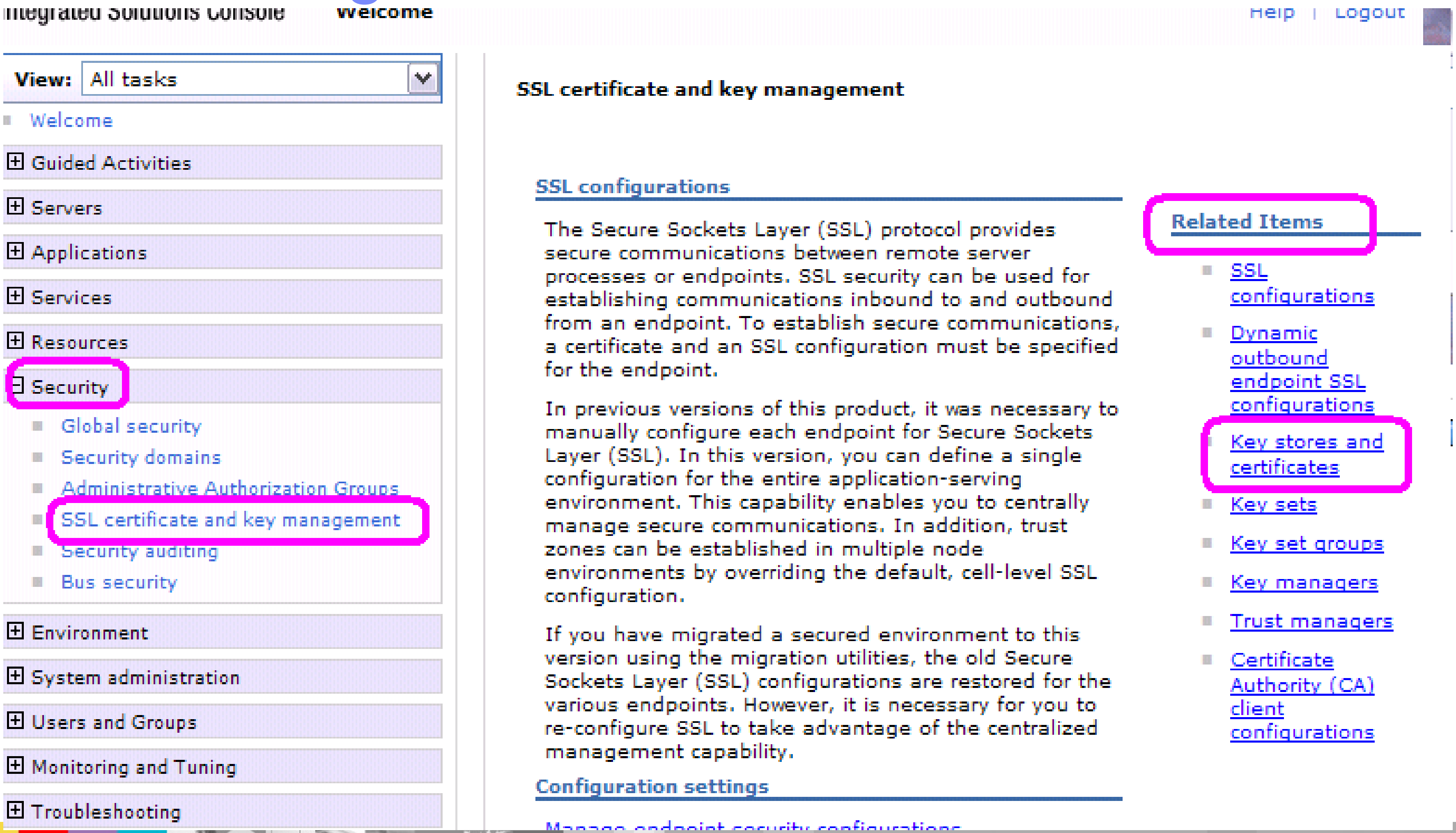
- On the host of the MQ Server, add the WAS Signer Certificate to the Queue Manager's key database.

- 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

WAS - Configure SSL Certificate Stores - 1

- Login to the WAS Administrative Console.
- Open Security -> SSL certificate and key management
- Under "Related Items" select Key stores and certificates.

WAS - Configure SSL Certificate Stores - 2



SSL certificate and key management

SSL configurations

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

Configuration settings

Manage endpoint security configurations

Related Items
SSL configurations
Dynamic outbound endpoint SSL configurations
Key stores and certificates
Key sets
Key set groups
Key managers
Trust managers
Certificate Authority (CA) client configurations

WAS - Configure SSL Certificate Stores - 3

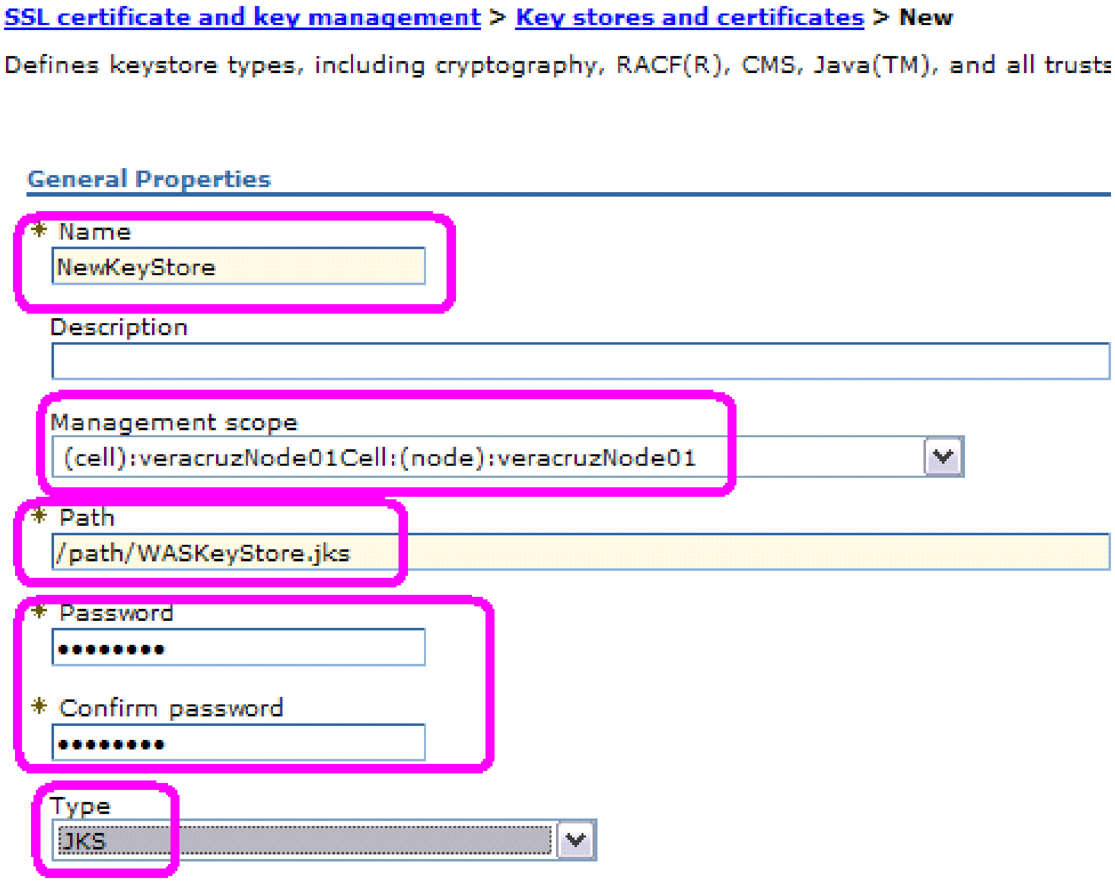
- Click the "New" button for the KeyStore:

- Name NewKeyStore
- Management scope (Select proper scope)
- Path /path/WASKeyStore.jks
- Type: JKS

- Repeat the operation for the TrustStore:
- Name NewTrustStore
- Management scope Select proper scope)
- Path /path/WASTrustStore.jks
- Type: JKS

WAS - Configure SSL Certificate Stores - 4

- This is what NewKeyStore looks like:



SSL certificate and key management > Key stores and certificates > New

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all trusts

General Properties

* Name
NewKeyStore

Description

Management scope
(cell):veracruzNode01Cell:(node):veracruzNode01

* Path
/path/WASKeyStore.jks

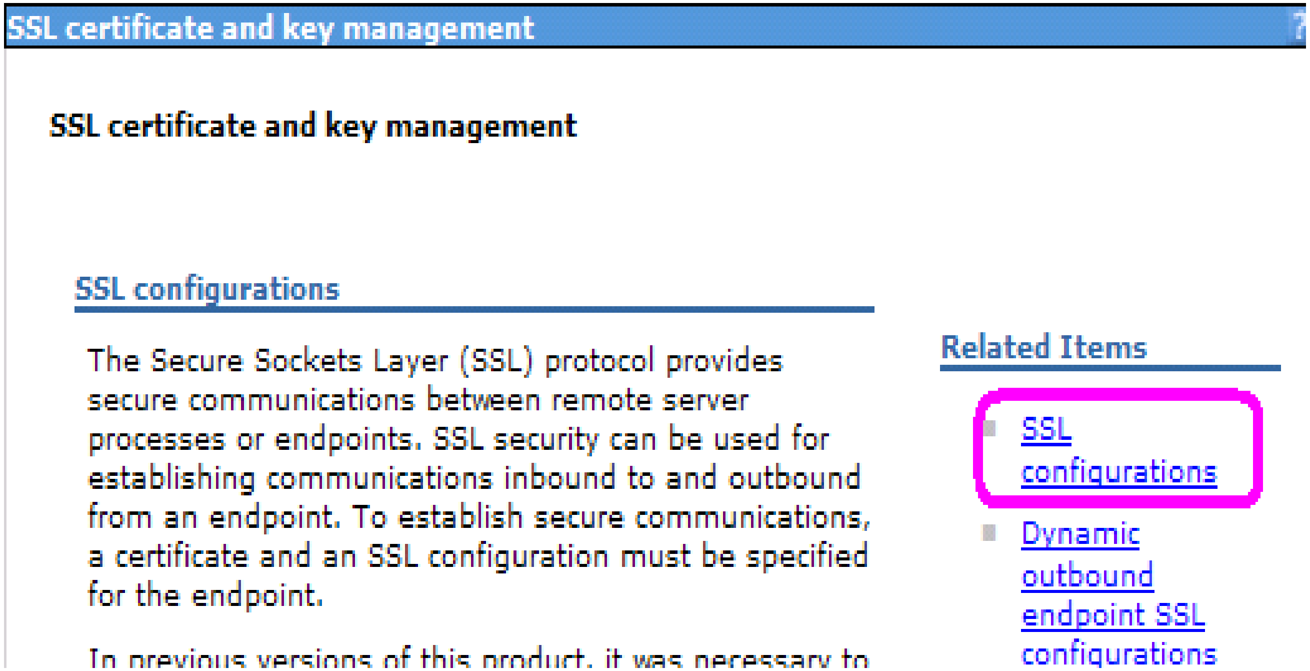
* Password
●●●●●●●●●

* Confirm password
●●●●●●●●●

Type
JKS

WAS - SSL Configuration - 1

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



SSL certificate and key management

SSL certificate and key management

SSL configurations

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to

Related Items

- SSL configurations
- Dynamic outbound endpoint SSL configurations

WAS - SSL Configuration - 2

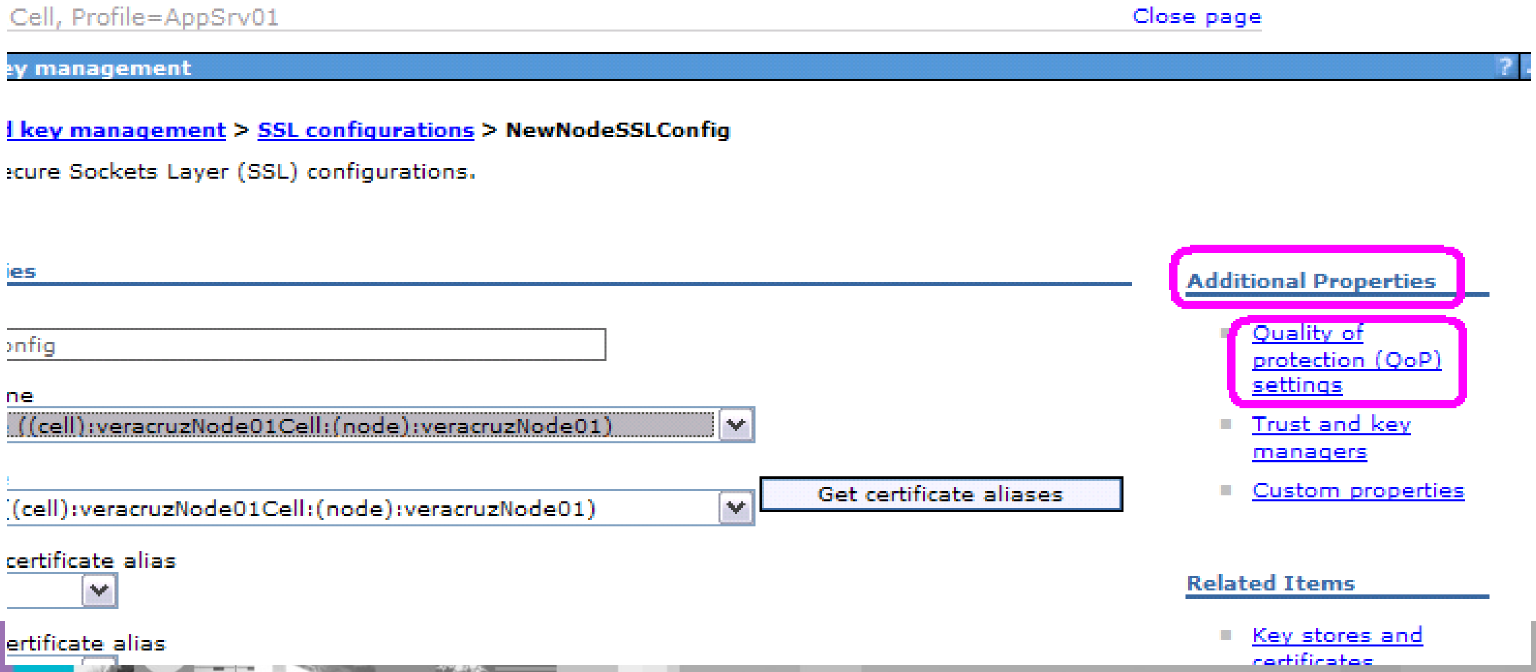
- Click the "New" button and fill out the following fields:

- Name NewNodeSSLConfg
- Trust store name NewTrustStore (*)
- Keystore name NewKeyStore (*)
- Management scope (*)

- (*) selected through drop down menu

WAS - SSL Configuration - 3

- Click on the name of the new SSL configuration, example: NewNodeSSLConfig
- Under "Additional Properties" Click Quality of protection (QoP) settings



Cell, Profile=AppSrv01 Close page

ey management

l key management > SSL configurations > NewNodeSSLConfig

ecure Sockets Layer (SSL) configurations.

ies

onfig

ne

.((cell):veracruzNode01Cell:(node):veracruzNode01)

(cell):veracruzNode01Cell:(node):veracruzNode01)

certificate alias

ertificate alias

WAS - SSL Configuration - 4

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

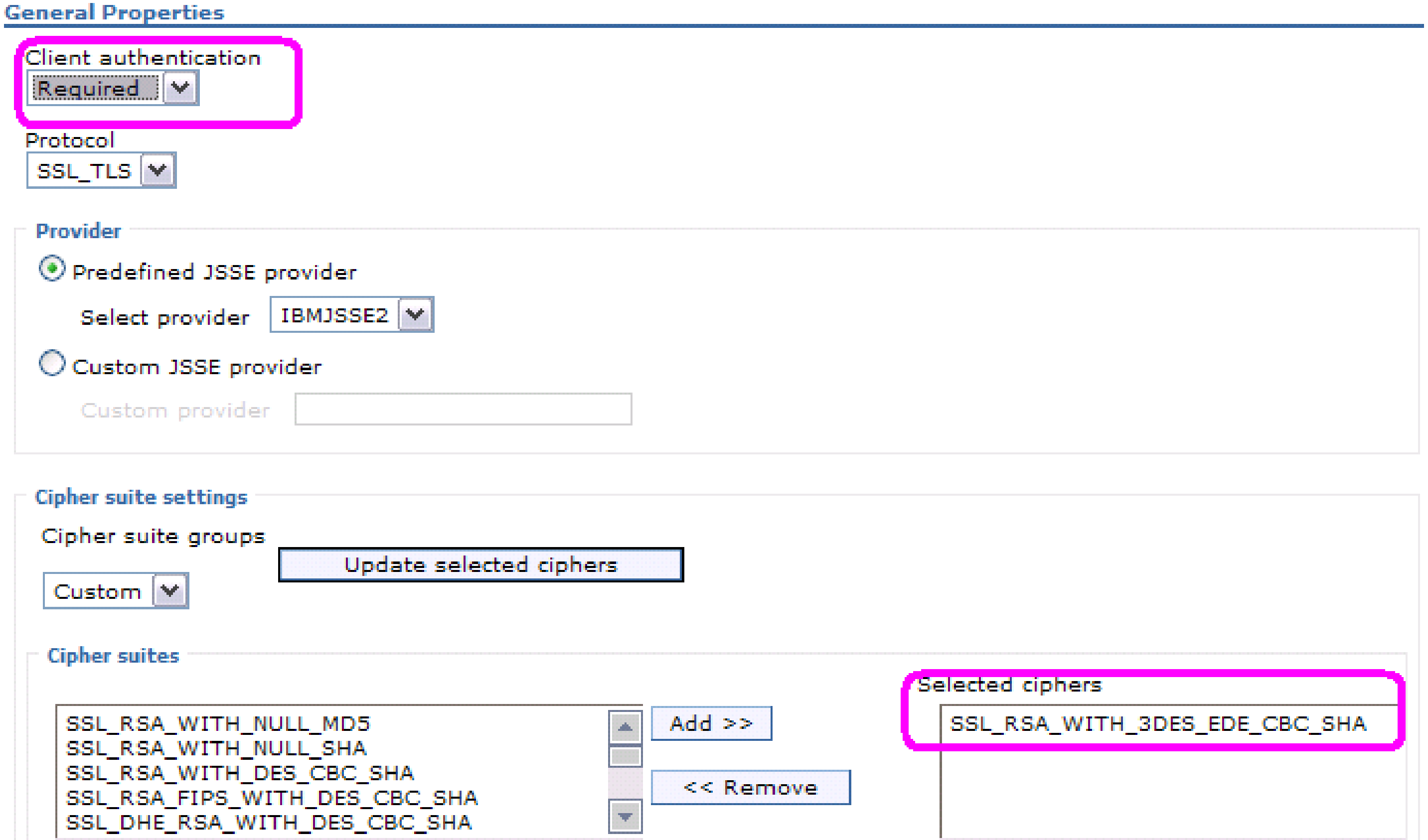
WAS - SSL Configuration - 5

- 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."

- The MQ channel was configured with strong encryption, TRIPLE_DES_SHA_US, thus, we chose the strong "Cipher suite groups", kept the matching Cipher Suite called SSL_RSA_WITH_3DES_EDE_CBC_SHA and removed all others.

WAS - SSL Configuration - 6

- This is what the screen looks like:



General Properties

Client authentication
Required

Protocol
SSL_TLS

Provider

● Predefined JSSE provider

Select provider IBMJSSE2

○ Custom JSSE provider

Custom provider

Cipher suite settings

Cipher suite groups
Custom

Cipher suites

SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_FIPS_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA

Selected ciphers
SSL_RSA_WITH_3DES_EDE_CBC_SHA

WAS - JMS Act Spec SSL configuration - 1

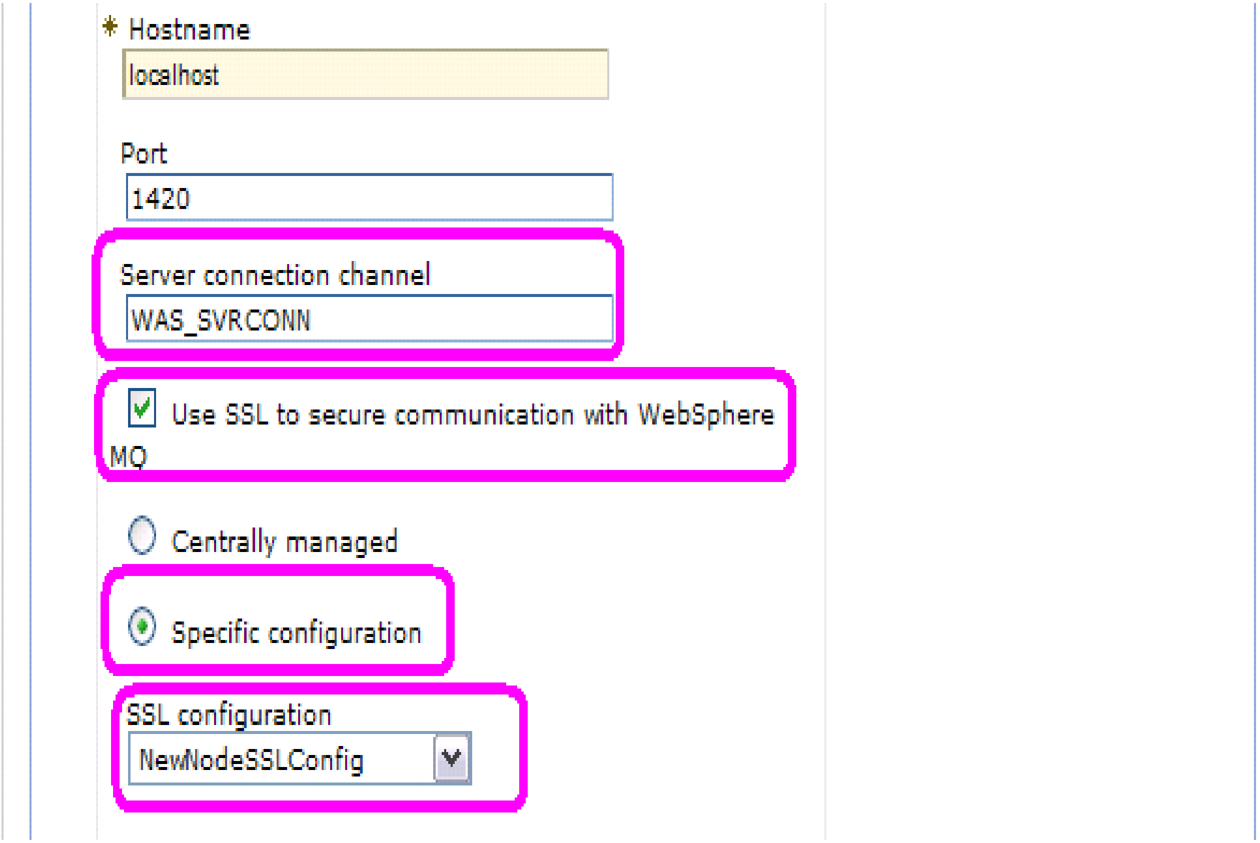
- Open Resources -> JMS -> Activation Specifications
- Click on: SampleMDBQueueActivationSpec

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

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

WAS - JMS Act Spec SSL configuration - 2

- This is what it looks like:



* Hostname
localhost

Port
1420

Server connection channel
WAS_SVRCONN

☑ Use SSL to secure communication with WebSphere MQ

○ Centrally managed

● Specific configuration

SSL configuration
NewNodeSSLConfig

Testing the SSL connection - 1

- Stop and restart the WAS Server so that all modifications will take effect.
- Start the application and verify the application is in a "Started" status.
- Repeat the testing of the Sample MDB for the original non-SSL scenario

Testing the SSL connection - 2

- Window 1: (MQ)

- Put message

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

Testing the SSL connection - 3

- Window 2: (WAS)

- View bottom of WAS SystemOut.log

- [6/17/11 1:32:19:230 EDT] 00000029 SystemOut   O
- +++ SAMPLE MDB: Text Message
  => TESTING MESSAGE WITH SSL CONNECTION

Troubleshooting

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

SupportPac: MO04: WebSphere MQ SSL Wizard

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

- 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.

References (1)

- Techdoc: Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7
- http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505
- It has a Sample MDB

- Techdoc: Using SSL to Connect to a WebSphere Application Server with a WebSphere MQ Queue Manager
- http://www.ibm.com/support/docview.wss?rs=171&uid=swg27021934
- All steps to perform the SSL configuration in both WAS and MQ

References (2)

- WebSphere MQ V7 Information Center:
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

- MQ V7: SSL CipherSpecs and CipherSuites
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.csqzaw.doc/jm34740_.htm

- WebSphere Application Server V7 Information Center:
- http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/

- Join the Global WebSphere Community: http://www.websphereusergroup.org

- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant

- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html

- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line "wste subscribe" to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter

Questions and Answers