IBM Product Connectivity Scenarios information center
Version 1 Release 0

Securing connection between WebSphere Application Server and WebSphere MQ

IBM

Note

Before using this information and the product it supports, read the information in “Notices” on page 43.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Chapter 1. Scenario: Securing WebSphere MQ connection to WebSphere Application Server

In a business environment where Java™ EE applications on WebSphere® Application Server consume and work with messages from WebSphere MQ, it is usually desirable to transmit data over the connection securely. Starting with an existing, unsecured, environment of WebSphere MQ connected to WebSphere Application Server, this scenario leads you through the key tasks that are required to make the connection secure.

This scenario was developed with the sample environment that is created in the getting started scenario as the starting point, Connecting WebSphere Application Server to WebSphere MQ. This scenario uses sample applications to demonstrate the effect of securing the connection between WebSphere Application Server and WebSphere MQ.

Notes

This scenario applies to the use of either WebSphere Application Server Version 7 or Version 8 with WebSphere MQ version 7.

This scenario was developed and tested with WebSphere Application Server Version 7 and WebSphere MQ version 7. If you use WebSphere Application Server Version 8, you will notice some differences like using the IBM Installation Manager instead of the installation wizard described in this scenario.

Optional information to help you learn while implementing the solution

The scenario contains blocks of optional information marked by Why? or What else?. You do not need to read this information to complete the scenario, but might choose to learn more:

Why? Describes *why* you are instructed to do something. For example:

> **Why am I doing this?**
>
> You use the **Scope** property to set the level at which the activation specification is visible. The cell scope is the highest level, giving the activation specification the greatest visibility.

What else?

Describes *what else* you might do, or want to learn about, related to what you are reading in the main window. For example:

> **What else might I do or be interested in?**
>
> You can also create activation specifications at other levels. For example, if you have multiple servers you might create an activation specification for each server, using the server scope, so that you can specify different settings to be used for each server.

Tip: Some “Why?” and “What else?” information provides links that would take you to information resources outside the scenario. To complete a scenario, you do not need to follow such links; they are provided only as optional aids for your learning.

Related information:

Scenarios and Patterns
This scenario in the Scenarios and Patterns information center

Product web page
WebSphere Application Server product web page

Library page
WebSphere Application Server library page

Product web page
WebSphere MQ product web page

Library page
WebSphere MQ library page

Chapter 2. Scenario: Securing WebSphere MQ connection to WebSphere Application Server

In a business environment where Java EE applications on WebSphere Application Server consume and work with messages from WebSphere MQ, it is usually desirable to transmit data over the connection securely. Starting with an existing, unsecured, environment of WebSphere MQ connected to WebSphere Application Server, this scenario leads you through the key tasks that are required to make the connection secure.

This scenario was developed with the sample environment that is created in the getting started scenario as the starting point, Connecting WebSphere Application Server to WebSphere MQ. This scenario uses sample applications to demonstrate the effect of securing the connection between WebSphere Application Server and WebSphere MQ.

Notes

This scenario applies to the use of either WebSphere Application Server Version 7 or Version 8 with WebSphere MQ version 7.

This scenario was developed and tested with WebSphere Application Server Version 7 and WebSphere MQ version 7. If you use WebSphere Application Server Version 8, you will notice some differences like using the IBM Installation Manager instead of the installation wizard described in this scenario.

Optional information to help you learn while implementing the solution

The scenario contains blocks of optional information marked by Why? or What else?. You do not need to read this information to complete the scenario, but might choose to learn more:

Why? Describes *why* you are instructed to do something. For example:

> **Why am I doing this?**
>
> You use the **Scope** property to set the level at which the activation specification is visible. The cell scope is the highest level, giving the activation specification the greatest visibility.

What else?

Describes *what else* you might do, or want to learn about, related to what you are reading in the main window. For example:

> **What else might I do or be interested in?**
>
> You can also create activation specifications at other levels. For example, if you have multiple servers you might create an activation specification for each server, using the server scope, so that you can specify different settings to be used for each server.

Tip: Some “Why?” and “What else?” information provides links that would take you to information resources outside the scenario. To complete a scenario, you do not need to follow such links; they are provided only as optional aids for your learning.

Related information:

Scenarios and Patterns
This scenario in the Scenarios and Patterns information center

Product web page
WebSphere Application Server product web page

Library page
WebSphere Application Server library page

Product web page
WebSphere MQ product web page

Library page
WebSphere MQ library page

Chapter 3. Planning the solution

Review the topics in this section to understand what is covered in the scenario: the reasons why a business might want to follow the scenario, the user roles that are involved, and an overview of the solution that is proposed by the scenario.

Assumptions

This scenario makes several assumptions about your system, such as the version of the products that you are using.

This scenario assumes the following points:

- You are using the Windows operating system.
- You are using WebSphere MQ Version 7.5 with WebSphere Application Server Version 7.
- You have already created the delivered topology for the Getting started scenario. For instructions to create this environment, see the Getting started scenario.
- You use the graphical interfaces of WebSphere MQ, WebSphere Application Server, and IBM® Key Manager, rather than the command-line equivalents.
- You have no security mechanisms on your messaging infrastructure currently.

Note: This scenario creates and uses its own certificate authority to sign certificates. In a production environment, creating your own certificate authority is not as secure as using a recognized authority. To fully complete this scenario, use a recognized authority to sign certificate requests.

Business overview

A company wants to secure an existing IT environment that is provided by WebSphere MQ and WebSphere Application Server.

To date, company A has a business solution that uses a messaging infrastructure that is provided by WebSphere MQ. WebSphere MQ is connected to WebSphere Application Server so that Java EE applications on WebSphere Application Server can use and work with messages from WebSphere MQ.

The current solution does not promote identification, authentication, confidentiality, or data integrity. The current messaging infrastructure is insecure.

Company A decides to deploy a range of security mechanisms on their IT environment to gain business value that includes:

- Offering more assurances about their services.
- Preventing unauthorized leaking of data which can lead to negative publicity.
- Benefiting from a more enclosed system.

User roles and interactions

The user roles and interactions that are described are examples of what is involved in securing an environment.

The user roles focus on deploying the solution, and not the development.

Deploy the solution

Deploy the solution for production use, by installing, configuring, and testing the components that provide the solution.

Enterprise Architect
The Enterprise Architect is responsible for creating and maintaining the overall technical direction and infrastructure of an enterprise IT environment. Part of their responsibility is to establish the security strategy, choose which security products to use, and set policies, procedures, and guidelines for security. They must work with architects, administrators, and developers.

Administrator
The Administrator configures the components that support the solution, in this case WebSphere MQ and WebSphere Application Server. The Administrator must also configure the security certificates and ensure that they are properly signed. The administer must also be responsible for ensuring that certificates are still valid. The business might have multiple administrators for each of these tasks, or they might have one central administrator. If there are multiple administrators, they must communicate information between each other, such as the locations of the certificates or the queue manager's name.

Test Implementor
The Test Implementor runs the tests to verify that the solution is ready for use. For example, can the solution transfer messages properly and securely? Can unauthorized users not access the environment?

Develop for the solution

Developer
The Developer is responsible for creating and testing the software components that support the solution. The developer must get the certificate information from the Administrator. The Developer might be required to edit the source code of applications to enable the use of SSL.

Technical solution

This scenario describes securing the connection between WebSphere MQ and WebSphere Application Server. The scenario also describes securing the connection between clients and WebSphere MQ.

Overview: Initial IT configuration

The company uses a Java EE application on WebSphere Application Server to consume and work with messages from a messaging infrastructure that is provided by WebSphere MQ.

The initial IT configuration includes several components that an administrator configures or uses, as shown in Figure 1.

WebSphere MQ as a ***messaging provider*** **for WebSphere Application Server**
The WebSphere MQ messaging provider in WebSphere Application Server makes JMS messaging available to WebSphere Application Server applications by using the existing capabilities in the WebSphere MQ environment.

Why am I doing this?

WebSphere Application Server applications can interact with WebSphere MQ destinations to send and receive messages in the same way as any JMS application in the WebSphere MQ environment.

What else might I do or be interested in?

- You can connect a WebSphere MQ network to a service integration bus within WebSphere Application Server, by using WebSphere MQ links. A WebSphere MQ link provides support for sender-receiver channels between the service integration bus and a WebSphere MQ queue manager or queue-sharing group. This option requires more complex configuration in WebSphere Application Server; you must configure a service integration bus and messaging engines.
- You can add a WebSphere MQ server (representing a queue manager or queue-sharing group) as a member of a service integration bus within WebSphere Application Server. A WebSphere MQ server provides a direct client or bindings connection between a service integration bus and queues on a WebSphere MQ queue manager. This option also requires you to configure a service integration bus and messaging engines in WebSphere Application Server.

Java EE application

The application consumes and works with messages on the WebSphere MQ queue, Q1. This application runs on an *application server* of the WebSphere Application Server product that is installed and connected with WebSphere MQ.

The sample application in this scenario provides a *message-driven bean (MDB)* as an asynchronous message consumer. When a message arrives at the queue, the MDB automatically processes the message without the application explicitly polling the queue.

Why am I doing this?

MDBs are activated by the EJB container in WebSphere Application Server on receipt of a message. A typical MDB performs messaging functions, and calls one or more session beans to perform business logic. Because of this separation of function, you can easily change and reuse units of business logic without affecting the messaging function of the application.

Application server

A server program in WebSphere Application Server that provides the execution environment for Java EE application programs.

WebSphere Application Server JNDI namespace

WebSphere Application Server includes a name server which provides access to the following JMS administered objects through the Java Naming and Directory Interface (JNDI). .

Activation specification, myActSpec

A JMS activation specification can be associated with one or more MDBs and provides the configuration necessary for them to listen for messages that arrive at a destination. Activation specifications process inbound messages to the MDB.

Why am I doing this?

Activation specifications are part of the Java EE Connector architecture (JCA) 1.5 standard. JCA 1.5 provides a standard way to integrate JMS providers, such as WebSphere MQ, with Java EE application servers such as WebSphere Application Server.

What else might I do or be interested in?

Use of listener ports is an older configuration method for MDBs to listen for messages that arrive at a destination. The use of listener ports is stabilized in WebSphere Application Server Version 7 and later. If you are using WebSphere Application Server for the first time avoid using listener ports.

Connection factory, myCF

A JMS connection factory object defines a set of standard configuration properties for connections. An application uses a connection factory to create a connection to WebSphere MQ.

Destination, myQueue

A JMS destination can be a topic or a queue. In this scenario the destination is a queue. The queue identifies the WebSphere MQ queue that applications send messages to, or from which an application receives messages, or both. An application looks up the destination in the JNDI namespace to create a connection to the WebSphere MQ queue.

In this scenario, when JMS objects are created in WebSphere Application Server, you specify full details rather than use a *client channel definition table (CCDT)* exported from WebSphere MQ.

Why am I doing this?

A CCDT makes configuring objects in WebSphere Application Server easier because it contains information that is required to connect to WebSphere MQ. If you do not specify a CCDT, you must enter this information yourself. This scenario does not use a CCDT so that you can see more clearly what information is required to make a connection.

What else might I do or be interested in?

A CCDT is useful if your client applications must connect to a number of alternative queue managers. This scenario does not cover that situation.

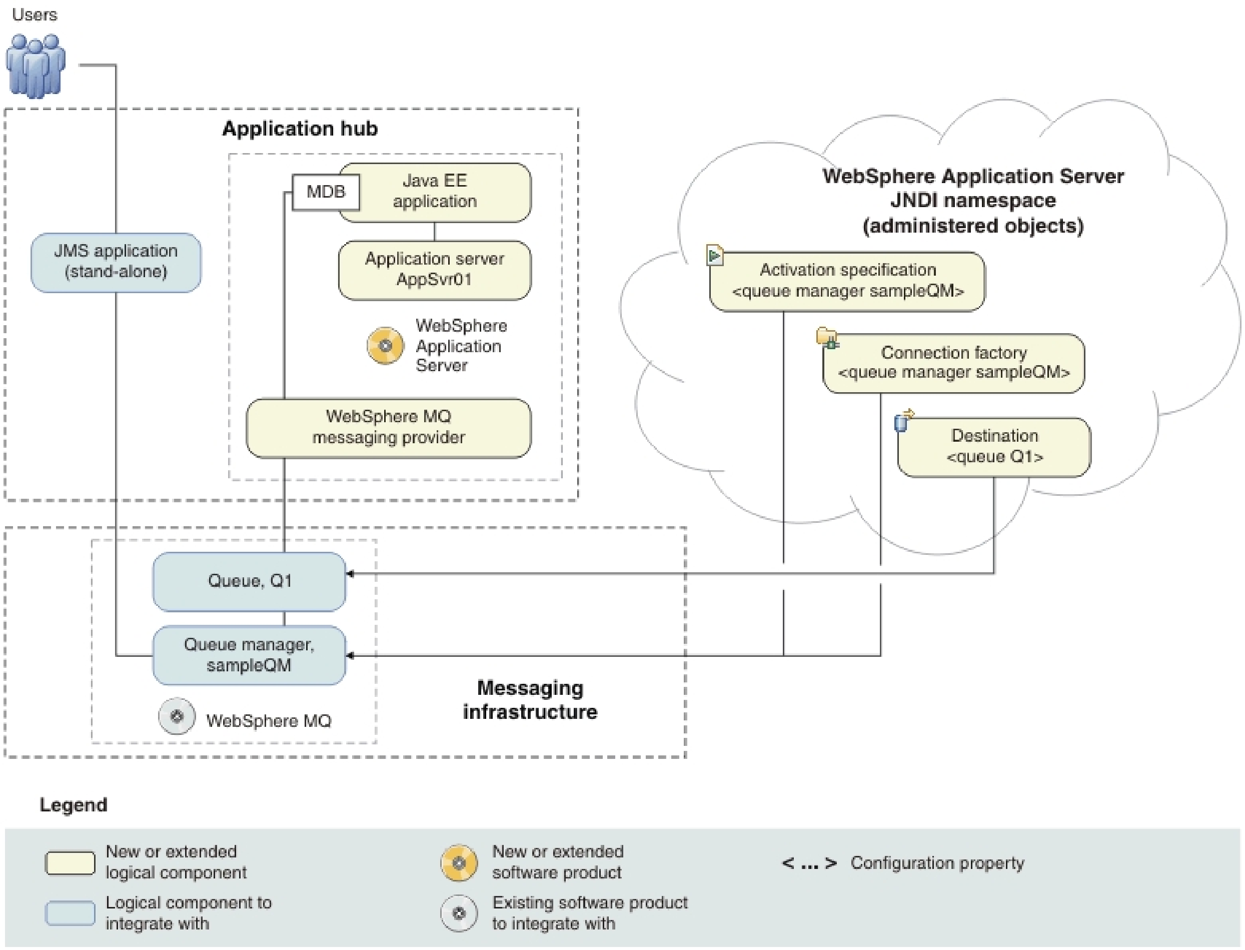
Users
Application hub
MDB
Java EE application
JMS application (stand-alone)
Application server AppSvr01
WebSphere Application Server
WebSphere MQ messaging provider
WebSphere Application Server JNDI namespace (administered objects)
Activation specification <queue manager sampleQM>
Connection factory <queue manager sampleQM>
Destination <queue Q1>
Queue, Q1
Queue manager, sampleQM
WebSphere MQ
Messaging infrastructure
Legend
New or extended logical component
Logical component to integrate with
New or extended software product
Existing software product to integrate with
< ... > Configuration property

Figure 1. Initial logical topology. The high-level logical topology diagram for software functions that are provided by WebSphere Application Server connected to WebSphere MQ. The new components that are indicated in the diagram were added when WebSphere Application Server was connected to WebSphere MQ in the “Getting Started” scenario.

Overview: The delivered logical topology

The company secures communication between WebSphere MQ and WebSphere Application Server with the use of an SSL connection.

The delivered IT configuration includes several components that an administrator configures or uses, as shown in Figure 1.

WebSphere MQ

WebSphere MQ provides the queue manager to manage the WebSphere MQ queue.

Queue manager key repository

The queue manager key repository stores the certificates that are used by WebSphere MQ; the queue manager's certificate signed by a certificate authority and the public certificate of the certificate authority.

Server-connection channel

The server-connection channel uses SSL encryption. The channel requires a specific CipherSpec to be used when a connection is made.

Application Server

A server program in WebSphere Application Server that provides the execution environment for Java EE application programs.

Keystore

The WebSphere Application Server truststore contains the WebSphere Application Server certificate that is signed by a certificate authority.

Truststore

The WebSphere Application Server truststore contains the public certificate of the certificate authority and the public certificate of the WebSphere Application Server.

SSL configuration

The SSL configuration defines what is needed by each endpoint in a communicating system. The SSL configuration points to the WebSphere Application Server truststore and keystore. The SSL configuration must use a cipher suite that is compatible with the CipherSpec defined on the WebSphere MQ channel. The WebSphere Application Server administered objects must use the SSL configuration to communicate with the WebSphere MQ queue.

Stand-alone JMS application

The JMS application must have a keystore and a truststore that contain; the public certificate from a certificate authority and the signed JMS application certificate.

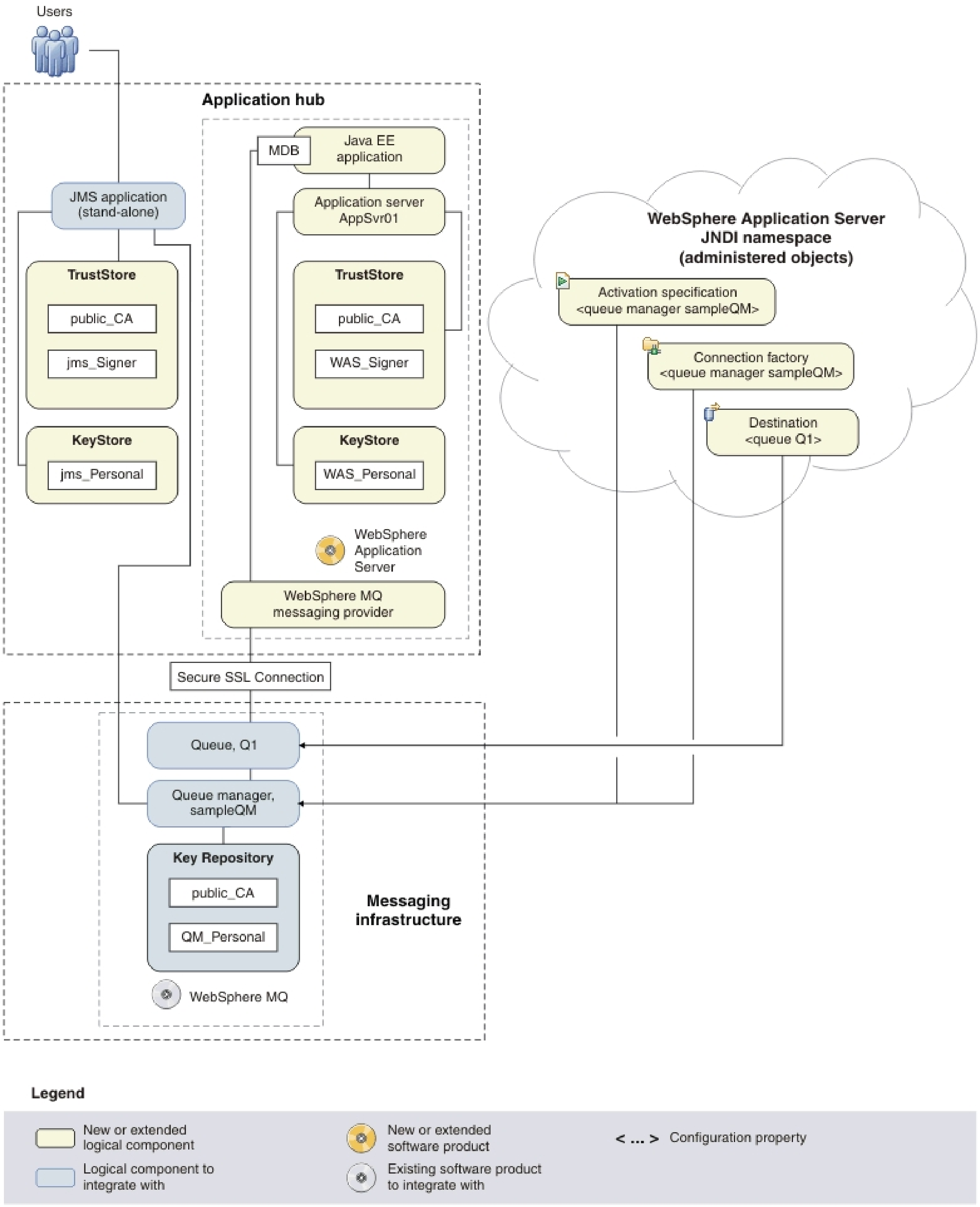
Users
Application hub
MDB
Java EE application
Application server AppSvr01
JMS application (stand-alone)
TrustStore
public_CA
jms_Signer
KeyStore
jms_Personal
TrustStore
public_CA
WAS_Signer
KeyStore
WAS_Personal
WebSphere Application Server
WebSphere MQ messaging provider
WebSphere Application Server JNDI namespace (administered objects)
Activation specification <queue manager sampleQM>
Connection factory <queue manager sampleQM>
Destination <queue Q1>
Secure SSL Connection
Queue, Q1
Queue manager, sampleQM
Key Repository
public_CA
QM_Personal
WebSphere MQ
Messaging infrastructure
Legend
New or extended logical component
Logical component to integrate with
New or extended software product
Existing software product to integrate with
< ... > Configuration property

Figure 2. Delivered logical topology. The high-level topology diagram for the secured WebSphere MQ and WebSphere Application Server configuration. This delivered logical topology shows the new elements that are integrated with the products from the initial topology.

Production physical topology and product mapping

The production physical topology specifies the recommended operational environment (machines, operating systems, and software products) for running the applications and services of the solution.

Node	Operating system and hardware	Software
Host 1 (runtime node)	• Windows Server 2003 or 2008, Standard or Enterprise Edition. You can use Windows 7, or for 32-bit environments, Windows Vista or Windows XP Professional, but these operating systems are not supported for production use of WebSphere® Application Server. • Intel EM64T, AMD Opteron, or for 32-bit environments, Intel Pentium processor at 2.33 GHz or faster • 1 GB physical memory • 3 GB free disk space	• WebSphere MQ • WebSphere Application Server • IBM Key Manager Prerequisite software: • A web browser for use by the Launchpad, Windows Explorer, and WebSphere Application Server administrative console: Microsoft Internet Explorer Version 6.0, or Mozilla-based browsers Version 1.7 or higher

Chapter 4. Implementing the solution

Implementing the solution in this scenario involves securing communications between WebSphere MQ and WebSphere Application Server or clients.

Before you begin

The starting point for this scenario is the IT configuration that is created by working through the Getting started scenario. This configuration consists of WebSphere MQ connected to WebSphere Application Server on the same Windows computer.

About this task

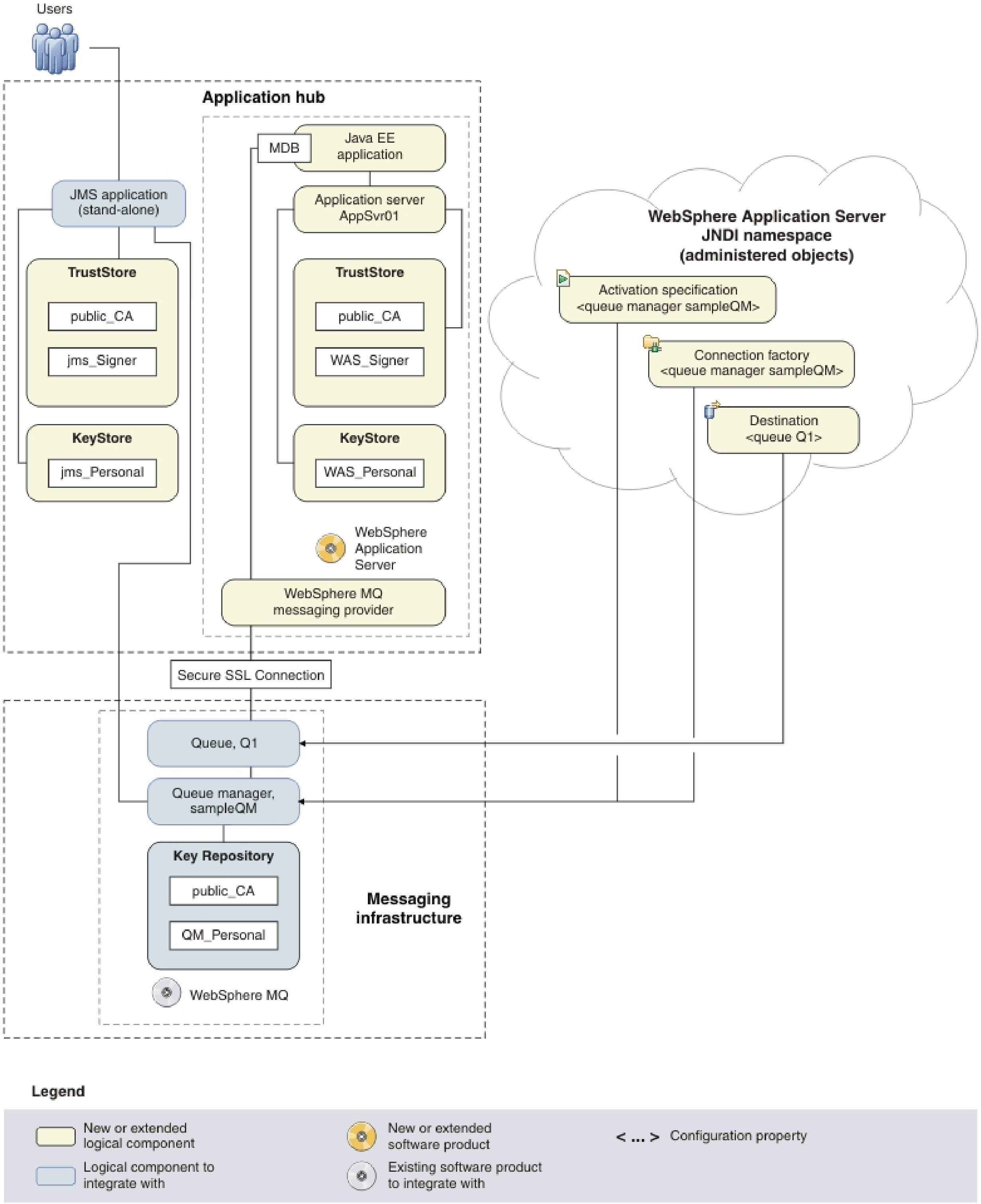
Users
Application hub
MDB
Java EE application
JMS application (stand-alone)
Application server AppSvr01
TrustStore
public_CA
jms_Signer
KeyStore
jms_Personal
TrustStore
public_CA
WAS_Signer
KeyStore
WAS_Personal
WebSphere Application Server
WebSphere MQ messaging provider
WebSphere Application Server JNDI namespace (administered objects)
Activation specification <queue manager sampleQM>
Connection factory <queue manager sampleQM>
Destination <queue Q1>
Secure SSL Connection
Queue, Q1
Queue manager, sampleQM
Key Repository
public_CA
QM_Personal
WebSphere MQ
Messaging infrastructure
Legend
New or extended logical component
Logical component to integrate with
New or extended software product
Existing software product to integrate with
< ... > Configuration property

Figure 3. Delivered logical topology. The high-level topology diagram for the secured WebSphere MQ and WebSphere Application Server configuration. This delivered logical topology shows the new elements that are integrated with the products from the initial topology.

Securing the channel to WebSphere MQ

Restrict access to WebSphere MQ by creating rules to define who has privileges to make a connection with WebSphere MQ, promoting the integrity of data within the system.

About this task

Create WebSphere MQ channel authentication records so that you have more control over the connections that are made to WebSphere MQ.

Creating the channel authentication record

Channel authentication records restrict connections to a channel in WebSphere MQ. Offering another layer of protection to the data used.

About this task

Create a channel authentication record to allow WebSphere Application Server to connect to WebSphere MQ. The channel authentication record allows a connection from the corresponding IP address. Optionally create a channel authentication record to allow client applications to connect to WebSphere MQ.

Procedure

1. Create a channel authentication record.
   a. Start WebSphere MQ Explorer. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **WebSphere MQ Explorer**. If WebSphere MQ Explorer is running and displaying the Welcome page, close the Welcome page to start administering WebSphere MQ objects.
   b. Expand **Queue Managers** > **sampleQM** > **Channels**. Right-click **Channel Authentication Records** and click **New** > **Channel Authentication Record**.
   c. Select **Allow access** and click **Next**.

   > **Why am I doing this?**
   > Allow access defines rules that enable people to make a connection. Block access defines rules that restrict people to make a connection.

   > **What else might I do or be interested in?**
   > If block access is selected, then warn mode becomes available, meaning that the rule does not actually block the connections it just warns of connections that would be blocked.

   d. Select **IP address**.
   e. In the **Channel profile** field, type `WAS_SVRCONN` and click **Next**.

   > **Why am I doing this?**
   > `WAS_SVRCONN` is the name of the channel used in this scenario.

   f. In the **IP address pattern** field, type the IP address of the WebSphere Application Server and click **Next**.

g. Select **Fixed user ID** and enter a user ID which has the required privileges on the queue manager.

h. In the **Description of rule** type `Allows the client application to connect to WebSphere MQ`.

i. Click **Next** and then click **Finish**. The channel authentication is shown in the channel authentication records pane.

2. Optional: Create a channel authentication to allow client applications to connect to WebSphere MQ.

   a. Expand **Queue Managers** > **sampleQM** > **Channels**. Right-click **Channel Authentication Records** and click **New** > **Channel Authentication Record**.

   b. Select **Allow access** and click **Next**.

   c. Select **Client application user ID**.

   d. In the **Channel profile** field, type `WAS_SVRCONN` and click **Next**.

   e. In the **Remote client user ID** field, type the user ID running the client application and click **Next**.

   f. Select **Fixed user ID** and enter a user ID which has the required privileges on the queue manager.

   g. In the **Description of rule** type `Allows the client application to connect to WebSphere MQ`.

   h. Click **Next** and then click **Finish**. The channel authentication is shown in the channel authentication records pane.

Results

A channel authentication record is created that allows a connection from a client with the specified IP address.

What to do next

Complete the instructions in "Configuring the solution to use SSL."

Configuring the solution to use SSL

Complete the instructions in the subtopics to enable the solution to use SSL to connect WebSphere MQ with WebSphere Application Server.

Before you begin

The starting point for this scenario is the sample IT configuration that is described in the getting started scenario.

About this task

Secure Sockets Layer (SSL) is an example of link level security. SSL is used to promote authentication, confidentiality, and data integrity within the messaging environment. SSL initiates communication with the authentication of the users at each end of the connection. Each user has a signed certificate, hash values authenticate each user against the certificates. The hash value provides assurances that the data transmitted is identical to what was originally sent, confirming the authenticity of the certificate.

Creating a certificate authority

A certificate authority (CA) signed certificate proves the authenticity of the entity that uses the certificate. Used in combination with the public certificate of the CA, the authenticity of the certificate is tested.

Before you begin

- Create a directory in `C:\Program Files\IBM\` called `myCA`.

About this task

In this task, you create a CA to sign the certificates of the other entities in the environment. Create a self signed certificate to be the public internal certificate. To deploy the solution in a production environment use a third party, approved, CA to maximize security. In a production environment, a certificate request is created and then sent to a third-party CA to sign.

Procedure

1. Start the IBM Key Management utility. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the CA keystore.
   a. In the **Key Database File** menu click **New**.
   b. Set the **Key Database Type** to **CMS**.
   c. Enter `myCA.kdb` as the **File Name**.
   d. Set the **Location** to `C:\Program Files\IBM\myCA`.
   e. Click **OK**.
   f. Type `passw0rd` in the **Password** field, then retype the password.
   g. Check **Stash password to a file**. Click **OK**.

   > **Why am I doing this?**
   > The password must be stashed so that WebSphere MQ and WebSphere Application Server can access it.

   The keystore is created and the home screen opens.
3. Create the CA certificate.
   a. Click **Create** > **New Self-signed certificate**.
   b. In the **Key Label** field, type `myCAcertificate`.
   c. Leave the **Key Size** and **Signature Algorithm** fields with their default settings. In the **Common Name** field, type `myCAName`. All the other fields are optional, click **OK**. A confirmation message is displayed, the certificate request is created.
4. Extract the CA certificate.

   > **Why am I doing this?**
   > Extracting rather than exporting a certificate means that only the public part of the certificate is included. The public certificate is used by the other entities in the solution to verify the authenticity of the CA.

   a. Select the newly created certificate and click **Extract Certificate**.

b. Click **Browse** and locate the directory that was created earlier.

c. Set the file name to `myCertfile.cer` and click **OK**.

Results

A CA is created which is used to sign the certificates of the other entities in the scenario. The public internal certificate that was created by extracting the self-signed certificate, is used to verify the CA to other entities.

What to do next

To configure WebSphere MQ to use SSL, follow the instructions in "Configuring WebSphere MQ to use SSL."

Configuring WebSphere MQ to use SSL

Complete the instructions in the subtopics to enable WebSphere MQ to use SSL.

Before you begin

- You must have a CA to sign certificates. To create a CA to use in this scenario, follow the instruction in "Creating a certificate authority" on page 17.

About this task

Create a secure WebSphere MQ server-connection channel to accept SSL connections. Create a key repository for the WebSphere MQ queue manager, the key repository stores the signed queue manager's certificate and the public part of the CA certificate.

Creating a secure WebSphere MQ channel

Create a server-connection channel that uses SSL to receive connections.

About this task

In WebSphere MQ Explorer, create a server-connection channel that uses an SSL CipherSpec.

Procedure

1. Start WebSphere MQ Explorer. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **WebSphere MQ Explorer**. If WebSphere MQ Explorer is running and displaying the Welcome page, close the Welcome page to start administering WebSphere MQ objects.
2. Create a channel for the queue manager.
   a. Expand **Queue Managers**, then expand **sampleQM**. Right-click **Channels** > **New** > **Sever-connection Channel**. The Create a Server-connection Channel window opens.
   b. In the **Name** field, enter `WAS_SVRCONN` and click **Next**.
   c. On the SSL pane, from the **SSL Cipher Spec** menu select **TRIPLE_DES_SHA_US**. Click **Finish**.

Results

The server connection channel can be used for connection over SSL.

What to do next

To create the WebSphere MQ certificate, follow the instructions in "Creating the certificate request."

Creating the certificate request

The WebSphere MQ queue manager's key repository is used to store the queue manager's personal certificate and the public CA certificate. The personal certificate request from the WebSphere MQ queue manager must be signed by a CA, the public certificate is used by the other entities to authenticate the WebSphere MQ queue manager.

About this task

Create the WebSphere MQ queue manager's key repository, then create the queue manager's personal certificate request.

Procedure

1. Create the queue manager key repository.
   a. Open the IBM Key Management utility, click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
   b. In the **Key Database File** menu, click **New**.
   c. Set the **Key Database Type** to **CMS**.
   d. Leave the default **File Name**.
   e. Set the **Location** to `C:\Program Files\IBM\WebSphere MQ\Qmgrs\sampleQM\ssl`.
   f. Click **OK**.
   g. In the Password Prompt window, type `passw0rd`, then retype the password.
   h. Select **Stash password to a file**. Click **OK**.

   > **Why am I doing this?**
   > The password must be stashed so that WebSphere MQ and WebSphere Application Server can access it.

   The key repository is created and the home screen opened.
2. Create the certificate request.
   a. Click **Create** > **New Certificate Request**.
   b. In the **Key Label** field, type `ibmwebspheremqsampleqm`.

   > **Why am I doing this?**
   > The key label is *ibmwebspheremq* and the name of the queue manager, in this case `sampleQM`. The key label must be lowercase.

   c. Leave the **Key Size** and **Signiture Algorithm** fields with their default settings. In the **Common Name** field type `sampleQM`. In the **Enter the name of the file** field, edit the file name to `myqmgr.req` and click **OK**. All the other fields are optional. A confirmation message is displayed and the certificate request created.

Results

The WebSphere MQ queue manager's key repository is created. A certificate request is ready to be signed by the CA.

What to do next

In a production environment sign the certificate with a third party, approved, CA. To sign the certificate for the test solution, follow the instructions in "Sign the WebSphere MQ queue manager's certificate."

Sign the WebSphere MQ queue manager's certificate

Sign the WebSphere MQ queue manager's certificate with the CA. Add the signed certificate and the public internal certificate to the queue manager's key repository.

About this task

The WebSphere MQ queue manager certificate must be signed by the CA. Enabling the other entities in the environment to verify the authenticity of the certificate. The WebSphere MQ queue manager requires the public internal certificate to verify the certificates that are received from other entities in the scenario.

Procedure

1. Sign the queue manager's certificate.
   a. In `C:\Program Files\IBM\WebSphere MQ\Qmgrs\sampleQM\ssl\`, locate `myqmgr.req`.
   b. Use FTP to transfer `myqmgr.req` to the `myCA` directory.
   c. In a command prompt, navigate to the `myCA` directory and enter `runmqckm -cert -sign -db myCA.kdb -label "myCAcertificate" -expire 365 -format ascii -file myqmgr.req -target myqmgr.cer`. In the password prompt window, enter `passw0rd`.
2. Use FTP to transfer `myqmgr.cer` and `myCertfile.cer` into `C:\Program Files\IBM\WebSphere MQ\Qmgrs\sampleQM\ssl\`.
   a. In the IBM Key Management utility, select **Personal Certificates** from the menu.
   b. Click **Receive**, then click **Browse** and locate `myqmgr.cer`.
   c. Select **Signer Certificates** from the menu.
   d. Click **Add** then click **Browse** and locate `myCertfile.cer`.
   e. In the Enter a Label window, type `public_CA` and click **OK**.

Results

The WebSphere MQ queue manager recognizes the CA as approved. The queue manager's CA signed certificate proves its authenticity to other entities in the scenario.

What to do next

To configure WebSphere Application Server to use SSL with WebSphere MQ, follow the instructions in "Configuring WebSphere Application Server to use SSL" on page 21.

Configuring WebSphere Application Server to use SSL

WebSphere Application Server requires keystore configuration and an SSL configuration. All the administered objects require a reconfiguration to use the new SSL configuration. Enabling communication between WebSphere MQ and WebSphere Application Server to use SSL.

Before you begin

- You must install and configure WebSphere Application Server as described in the Getting Started scenario. To install WebSphere Application Server, follow the instructions in Installing WebSphere Application Server from the Getting started scenario. To configure WebSphere Application Server, follow the instructions in Configuring WebSphere Application Server from the Getting started scenario.

About this task

Complete the instructions in the subtopics to set up of WebSphere Application Server to accept SSL connections. At the end of this task your WebSphere Application Server resources are only accessible from authorized connections.

The securing process consists of defining a new SSL configuration with a keystore and truststore, and how to configure the resources to use this new configuration.

Creating the WebSphere Application Server keystore and truststore

Create the keystore and truststore that contain the necessary certificates for WebSphere Application Server. The keystore contains the WebSphere Application Server personal certificate. The truststore contains the public WebSphere Application Server certificate and the public CA certificate.

About this task

Use the IBM Key Manager to create the keystore and truststore for WebSphere Application Server. Create the certificate request from WebSphere Application Server in the keystore.

Procedure

1. Start the IBM Key Manager. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the WebSphere Application Server keystore.
   a. In the **Key Database File** menu, click **New**.
   b. Set the **Key Database Type** to **JKS**.
   c. In the **File Name** field, enter `WASKeyStore`.
   d. Set the **Location** to `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\MACHINE NAME\nodes\MACHINE NAME`.
   e. Click **OK**.
   f. In the Password Prompt, type `passw0rd`, then retype the password. The keystore is created and the home screen opened.
3. Create the certificate request for WebSphere Application Server.
   a. Click **Create** > **New Certificate Request**.
   b. In the **Key Label** field type `WAS_Personal_Cert`.
   c. Leave the **Key Size** and **Signature Algorithm** fields with their default settings. In the **Common Name** field, type `WASKeyStore`. In the **Enter the**

name of the file field, edit the file name to mywaskeystore.req and click **OK**. All the other fields are optional. A confirmation message is displayed and the certificate request created.

4. Create the WebSphere Application truststore.
   a. In the **Key Database File** menu, click **New**.
   b. Set the **Key Database Type** to **JKS**.
   c. In the **File Name** field, enter WASTrustStore.
   d. Set the **Location** to C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\MACHINE NAME\nodes\MACHINE NAME.
   e. Click **OK**.
   f. In the Password Prompt, type passw0rd, then retype the password. The keystore is created and the home screen opened.

What to do next

To sign and add the required certificates to the truststore, follow the instructions in "Adding certificates to the WebSphere Application Server truststore and keystore."

Adding certificates to the WebSphere Application Server truststore and keystore

Add the signed personal certificate and the public internal certificate from the CA to the WebSphere Application Server truststore. Add the signed personal certificate to the WebSphere Application Server keystore.

About this task

The WebSphere Application Server truststore contains the public certificates from the CA, and WebSphere Application Server. The public CA certificate is used to verify the certificates from other entities in the solution.

Procedure

1. Sign the WebSphere Application Server certificate.
   a. In C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\MACHINE NAME\nodes\MACHINE NAME, locate mywaskeystore.req.
   b. Use FTP to transfer mywaskeystore.req to the myCA directory.
   c. In a command prompt, navigate to the myCA directory and enter `runmqckm -cert -sign -db myCA.kdb -label "myCAcertificate" -expire 365 -format ascii -file mywaskeystore.req -target mywaskeystore.cer`. In the password prompt, enter the password for the CA keystore.
2. Open the WebSphere Application Server truststore.
   a. Open the IBM Key Management utility, click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
   b. In the **Key Database File** menu, click **Open.**
   c. Ensure the **Key Database Type** is **JKS**.
   d. Click **Browse** and locate the truststore.
   e. Click **OK**.
   f. In the Password Prompt window, type passw0rd then retype the password. The home screen opens.
3. Add the signed WebSphere Application Server certificate to the truststore.

   a. Use FTP to transfer `mywaskeystore.cer` into the WebSphere Application Server truststore directory.
   b. In the IBM Key Manager, select **Signer Certificates** from the menu.
   c. Click **Add**, then click **Browse** and locate `mywaskeystore.cer`.
   d. In the Enter a Label window, type `WAS_Personal` and click **OK**.
4. Add the CA public certificate to the truststore.
   a. Use FTP to transfer `myCertfile.cer` into the WebSphere Application Server truststore directory.
   b. In the IBM Key Manager, select **Signer Certificates** from the menu.
   c. Click **Receive**, then click **Browse** and locate `myCertfile.cer`.
   d. In the Enter a Label window, type `public_CA` and click **OK**.
5. Open the WebSphere Application Server keystore.
   a. Open the IBM Key Management utility, click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
   b. In the **Key Database File** menu, click **Open.**
   c. Ensure the **Key Database Type** is **JKS**.
   d. Click **Browse** and locate the keystore.
   e. Click **OK**.
   f. In the Password Prompt window, type `passw0rd` then retype the password. The home screen opens.
6. Add the signed WebSphere Application Server certificate to the keystore.
   a. Use FTP to transfer `mywaskeystore.cer` into the WebSphere Application Server keystore directory.
   b. In the IBM Key Manager, select **Personal Certificates** from the menu.
   c. Click **Receive**, then click **Browse** and locate `mywaskeystore.cer`.

What to do next

To link WebSphere Application Server to the keystore and truststore, follow the instructions in “Creating WebSphere Application Server SSL key repositories.”

Creating WebSphere Application Server SSL key repositories

In the WebSphere Application Server administrative console, create the key repository links to the keystore and truststore.

About this task

The WebSphere Application Server SSL keystore configuration provides a link to the key repositories created. Ensuring that WebSphere Application Server has access to the correct certificates.

Procedure

1. If WebSphere Application Server is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server**. A command window is displayed until the server is started; this process might take a few minutes.
2. In the administrative console click **Security** > **SSL certificate and key management**. Under **Related Items**, click **Keystores and certificates**. The “SSL certificate and key management” window opens.
3. Ensure that **SSL Keystores** is selected from the **Keystore usages** menu. Click **New**.

4. Configure the keystore.
   a. Enter the following information and click **OK**.

      Name `NewKeyStore`

      The name that is given to the keystore.

      Management scope
      (cell):*Machine Name***Node01Cell:(node):***Machine Name***Node01**

      Path Enter the location of the keystore.

      Password
      `passw0rd`

      Repeat for the **Confirm password** field.

      Type **JKS**

      Must be the same type as the keystore.
5. Configure the truststore.
   a. Enter the following information and click **OK**.

      Name `NewTrustStore`

      The name that is given to the keystore.

      Management scope
      (cell):*Machine Name***Node01Cell:(node):***Machine Name***Node01**

      Path Enter the location of the truststore.

      Password
      `passw0rd`

      Repeat for the **Confirm password** field.

      Type **JKS**

      Must be the same type as the truststore.

What to do next

To create the SSL configuration that is used by WebSphere Application Server, follow the instructions in "Creating the WebSphere Application Server SSL configuration."

Creating the WebSphere Application Server SSL configuration

In the WebSphere Application Server administrative console, create the SSL configuration that is used by the JMS resources.

About this task

The SSL configuration defines what is needed by each endpoint in a communicating system. Enabling elements in the application-serving environment to communicate with each other securely.

Procedure

1. Create an SSL configuration.
   a. In the administrative console click **Security** > **SSL certificate and key management**. Under **Related Items**, click **SSL configurations**.
   b. Click **New** and complete the following information.

Name `NewNodeSSLConfig`

The name that is given to the keystore.

Truststore name
NewTrustStore

Select from the menu.

Keystore name
NewKeyStore

Select from the menu.

Management scope
(cell):*Machine Name***Node01Cell:(node):***Machine Name***Node01**

c. Click **OK** to save the configuration.

2. Configure the SSL configuration.
   a. On the list of available SSL configurations, click **NewNodeSSLConfig**.
   b. Click **Get certificate aliases** to populate the default certificate aliases.
   c. Under **Additional Properties** click **Quality of protection (QoP) settings**
   d. In the **Cipher suite groups** menu, select **Strong** and click **Update selected ciphers**.
   e. Highlight all the ciphers and click **<< Remove**. Then, select **SSL_RSA_WITH_3DES_EDE_CBC_SHA** and click **Add >>**.

> **What else might I do or be interested in?**
> Remove all the Cipher Suites from the Selected ciphers, except the suite that matches what is configured on the Server connection channel of the WebSphere MQ Queue Manager.

   f. Click **OK** to save the configuration.

What to do next

To configure the WebSphere Application Server objects for SSL use, follow the instructions in "Configuring WebSphere Application Server resources."

Configuring WebSphere Application Server resources

Configure the WebSphere Application Server resources to use the SSL configuration and the secure server-connection channel.

About this task

The WebSphere Application Server resources are linked to an SSL configuration. Ensuring that when they make a connection the correct SSL options are used.

Procedure

1. Configure the connection factory.
   a. In the administrative console, click **Resources** > **JMS** > **Connection factories**.
   b. Ensure that the scope is set to **Cell=***cell_name*, then click the connection factory that is created from the Getting Started scenario.
   c. Under the **Connection** pane, in the **Server connection channel** field type `WAS_SVRCONN`.

d. Check **Use SSL to secure communication with WebSphere MQ** and then **Specific configuration**. From the menu select **NewNodeSSLConfig**. Click **OK**.

2. Configure the listener port for the connection factory.
   a. In the administrative console, click **Servers** > **Server Types** > **WebSphere application servers**.
   b. In the list of application servers, click **server1**.
   c. Expand **Messaging** under the **Communications** pane to the right of the window. Click **Message listener service**.
   d. Under **Additional Properties** click **Listener Ports**.
   e. Click **New** and enter the following values for the required fields, then click **OK**.

      Name `SampleQueueLP`

      This name is used to display the listener port in the administrative console.

      Initial State
      Started

      Specifies the execution state that is requested when the server is first started.

      Connection factory JNDI name
      `jms/replyCF`

      Specifies the JNDI name for the JMS connection factory to be used by the listener port.

      Destination JNDI name
      `jms/wasQueue`

      Specifies the JNDI name for the destination to be used by the listener port.
3. Configure the activation specification for the queue.
   a. In the administrative console, click **Resources** > **JMS** > **Activation specifications**.
   b. On the list of available activation specifications, click the activation specification from the Getting Started scenario. Under the **Connection** pane, in the **Server connection channel** field type `WAS_SVRCONN`. Also, check **Use SSL to secure communication with WebSphere MQ** and then **Specific configuration**. From the menu select **NewNodeSSLConfig**. Leave the other settings and click **OK**.
4. Click the **Save** link in the Messages section at the top of the pane to save your changes to the master configuration.

What to do next

To test the solution with the sample JMS application, follow the instructions in “Creating the JMS application key repositories.”

Creating the JMS application key repositories

Create the keystore and truststore for the stand-alone JMS application.

About this task

The stand-alone JMS client requires a keystore and truststore so that the application can communicate securely with WebSphere MQ and WebSphere Application Server.

Procedure

1. Start the IBM Key Manager. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the JMS application keystore.
   a. In the **Key Database File** menu, click **New**.
   b. Set the **Key Database Type** to **JKS**.
   c. In the **File Name** field, enter `jmsKeyStore`. The name of the keystore is used in by the `requestor.cmd` file. To change the name of the keystore, you must edit the `requestor.cmd` file to match the different name.
   d. Set the **Location** to the directory the sample application is stored.
   e. Click **OK**.
   f. In the Password Prompt window, type `passw0rd`, then retype the password.
3. Create the JMS application truststore.
   a. In the **Key Database File** menu, click **New**.
   b. Set the **Key Database Type** to **JKS**.
   c. In the **File Name** field, enter `jmsTrustStore`.
   d. Set the **Location** to the directory the sample application is stored.
   e. Click **OK**.
   f. In the Password Prompt window, type `passw0rd`, then retype the password.

What to do next

Follow the instructions in "Populating the key repositories" to create and sign the JMS application certificates.

Populating the key repositories

Create and sign the JMS application certificate request. Add the signed personal certificate and the public internal certificate to the JMS application truststore. Add the signed personal certificate to the JMS application keystore.

Before you begin

- You must have the JMS application key repositories that are set up in "Creating the JMS application key repositories" on page 26.

About this task

The JMS application truststore contains the public certificates from the CA, and JMS application. The public CA certificate is used to verify the certificates from other entities in the solution.

Procedure

1. Start the IBM Key Manager. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the JMS application personal certificate.
   a. Open the JMS keystore.

b. Click **Create** > **New Certificate Request**.

c. In the **Key Label** field, type `jmsPersonalCert`.

d. Leave the **Key Size** and **Signature Algorithm** fields with their default settings. In the **Common Name** field, type `jmsApplication`. In the **Enter the name of the file** field, edit the file name to `myJmsApplication.req` and click **OK**. All the other fields are optional. A confirmation message is displayed and the certificate request created.

3. Sign the JMS application personal certificate.

   a. Use the FTP to transfer `myJmsApplication.req` to the `myCA` directory.

   b. In a command prompt, navigate to the `myCA` directory and enter `runmqckm -cert -sign -db myCA.kdb -label "myCAcertificate" -expire 365 -format ascii -file myJmsApplication.req -target myJmsApplication.cer`. In the password prompt, enter the password for the CA keystore.

4. Add the signed JMS application certificate to the JMS keystore.

   a. Use FTP to transfer `myJmsApplication.cer` into the JMS keystore directory.

   b. In the IBM Key Management utility, open the JMS keystore.

   c. Select **Personal Certificates** from the menu.

   d. Click **Receive**, then click **Browse** and locate `myJmsApplication.cer`.

   e. In the Enter a Label window, type `jms_Personal` and click **OK**.

5. Add the CA public certificate to the JMS truststore.

   a. Use FTP to transfer `myCertfile.cer` into the JMS truststore directory.

   b. In the IBM Key Management utility, open the JMS truststore.

   c. Select **Signer Certificates** from the menu.

   d. Click **Add** and then click **Browse**, locate `myCertfile.cer`.

   e. In the Enter a Label window, type `public_CA` and click **OK**.

6. Add the JMS application certificate to the JMS truststore.

   a. In the IBM Key Management utility, open the JMS truststore.

   b. Select **Signer Certificates** from the menu.

   c. Click **Add** and then click **Browse**, locate `myJmsApplication.cer`.

   d. In the Enter a Label window, type `jms_Signer` and click **OK**.

What to do next

To test the solution, follow the instructions in "Testing SSL use between WebSphere MQ and WebSphere Application Server."

Testing SSL use between WebSphere MQ and WebSphere Application Server

Run the sample JMS and message drive bean (MDB) applications to verify that the scenario is implemented correctly.

Before you begin

- WebSphere Application Server and WebSphere MQ must be configured as described in "Configuring WebSphere Application Server to use SSL" on page 21 and "Configuring WebSphere MQ to use SSL" on page 18.
- The sample application must be deployed to WebSphere Application Server as described in Deploying the sample application to WebSphere Application Server.
- Download the sample application SampleJMSAppSecure.zip.

About this task

In this task, you run the sample application to verify that the connection is properly secured. The sample JMS application uses the objects in both WebSphere MQ and WebSphere Application Server.

Procedure

1. If WebSphere Application Server is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server**. A command window is displayed until the server is started; this process might take a few minutes.
2. Click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Administrative console** then log in to the administrative console.
3. Click **Applications** > **Application Types** > **WebSphere enterprise applications** to display a list of installed applications.
4. Select the sampleMDB application and click **Start**. The application status changes to a green arrow. The MDB application is now waiting to receive a message.

   If you see an error message similar to the following message, there might be a mismatch between your WebSphere Application Server configuration and the configuration of WebSphere MQ. Check the WebSphere Application Server error logs in `*install_directory*\profiles\AppSrv01\logs\server1` for more information.

```
ErrorsampleMDB failed to start. Check the logs for server server1 on node mymachineNode01 for details.
ErrorAn error occurred while starting sampleMDB. Check the logs for server server1 on node mymachineNode01 for more infor
```

5. Double-click the `runrequester.cmd` batch file in the sample package. This command file opens a Command Prompt window then runs the sample JMS requester client, which sends a message to WebSphere MQ. The `runrequester.cmd` file uses environment variables to set the class path for running the JMS application. If you see a Java `java.lang.NoClassDefFoundError`, you might need to adjust the class path line in the command file.

Results

The sample requester application sends the same request message as before, but the response now comes from the MDB application. The following output, showing sent and received messages, is displayed in the requester application Command Prompt window:

```
> Connection factory located in JNDI.
> Destination located in JNDI.
> Creating connection to QueueManager.
> Connection created.

> Sending stock request for 'BakedBeans'
> Sent Message ID=ID:414d5120514d5f574153494e5354414c423fe14c20010602

> Received Message ID=ID:414d5120514d5f574153494e5354414c423fe14c2000f205 for 'B
akedBeans - 15 tins in stock. MDB has connected to business session bean to find
 expected delivery time: 1 day'

> Closing connection to QueueManager.
> Closed Connection.
```

If you see errors similar to the following error in the WebSphere Application Server logs, check that the name of the queue in WebSphere Application Server matches the administrative name (not the JNDI name) of the queue in WebSphere MQ:

```
WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2085' ('MQRC_UNKNOWN_OBJECT_NAME')
com.ibm.msg.client.jms.DetailedInvalidDestinationException: JMSWMQ2008: Failed to open MQ queue 'myQueue'. JMS attempted to
```

Note: If you update the WebSphere Application Server configuration, you must restart the application server for the changes to take effect.

You have finished implementing the solution, and verified that WebSphere Application Server can connect to WebSphere MQ for the successful transmission of secure messages.

What to do next

To connect other client applications to WebSphere MQ with SSL, complete the instructions in “Connecting client applications to WebSphere MQ with SSL.”

To add WebSphere Advanced Message Security to the solution, complete the instructions in “Adding WebSphere Advanced Message Security to the solution” on page 34.

Connecting client applications to WebSphere MQ with SSL

A stand-alone client application offers an efficient way of implementing WebSphere MQ messaging.

About this task

Connect different types of client application to WebSphere MQ. Use SSL to ensure that the clients are authenticated, and that confidentiality and integrity of data is confirmed.

Connecting a C client

Connect a stand-alone C client application to WebSphere MQ over an SSL connection.

Before you begin

- Download the *MO04* SSL pack from http://www-01.ibm.com/support/docview.wss?uid=swg24010367 to obtain the sample C application.
- Complete the instructions in “Creating the channel authentication record” on page 15 to create a channel authentication record to allow client applications to connect to WebSphere MQ.

About this task

Complete the instructions in the subtopics to create a keystore for the C application, generate the certificates, and test the SSL connection between the sample application and WebSphere MQ.

Configuring the client keystore and certificates

The keystore holds the certificate that is used to verify the client to the WebSphere MQ server.

About this task

Complete the steps in this task on the client host.

Procedure

1. Start the IBM Key Management utility. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the C client repository.
   a. In the **Key Database File** menu, click **New**.
   b. Set the**Key Database Type** to **CMS**.
   c. In the **File Name** field, enter `Cclient`.
   d. Set the **Location** to the `C:\Program Files\IBM\WebSphere MQ\ssl\`. Create the directory if it does not exist.
   e. Click **OK**.
   f. In the Password Prompt window, type `passw0rd`, then retype the password.
   g. Select **Stash password to a file**. Click **OK**.

   > **Why am I doing this?**
   > The password must be stashed so that WebSphere MQ and WebSphere Application Server can access it.

   The keystore is created and the home screen opened.
3. Create the certificate request for the client.
   a. Click **Create** > **New Certificate Request**.
   b. In the **Key Label** field, type `ibmwebspheremqmyuserid`. `myuserid` must all be lowercase.
   c. Leave the **Key Size** and **Signature Algorithm** fields with their default settings. In the **Common Name** field type `myuserid`. In the **Enter the name of the file** field, edit the file name to `Cclient.req` and click **OK**. All the other fields are optional. A confirmation message is displayed and the certificate request is created.
4. Sign the client certificate request.
   a. Use FTP to transfer `Cclient.req` to the `myCA` directory on the host of the certificate authority.
   b. In a command prompt, navigate to the `myCA` directory and enter `runmqckm -cert -sign -db myCA.kdb -label "myCAcertificate" -expire 365 -format ascii -file Cclient.req -target Cclient.cer`. In the password prompt window, enter `passw0rd`.
5. Use FTP to transfer `Cclient.cer` and `myCertfile.cer` to the client.
   a. In the IBM Key Management utility, select **Personal Certificates** from the menu.
   b. Click **Receive**, then click **Browse** and locate `Cclient.cer`.
   c. Select **Signer Certificates** from the menu.
   d. Click **Add**, then click **Browse** and locate `myCertfile.cer`

Results

The key repository is created and the client certificate request signed. The client has a valid certificate to enable secure communication with the WebSphere MQ server.

What to do next

To test the solution, follow the instructions in “Testing the SSL connection.”

Testing the SSL connection

Test the secure connection between the C client and WebSphere MQ.

About this task

Run the sample program on the client to make a secure connection to WebSphere MQ.

Procedure

1. Locate the `SSLSample.exe` in `mo04\client_samples\bin\`.
2. In the command prompt, navigate to the directory that contains the`SSLSample.exe`. Enter the command `SSLSample.exe Server Address(port) WAS_SVRCONN sampleQM TRIPLE_DES_SHA_US "C:\Program Files\IBM\WebSphere MQ\ssl\Cclient"`.

> **Why am I doing this?**
> The arguments in the **`MQCONNX`** call are: **`program, server-address(port), channel, queue manager, CipherSpec, key repository location`**.

Results

If secure communication is achieved `Connection Successful!` is displayed.

What to do next

- To add a Java client to the solution, follow the instructions in “Connecting a Java client.”
- To configure WebSphere Application Server to use SSL, follow the instructions in “Configuring WebSphere Application Server to use SSL” on page 21.

Connecting a Java client

Connect a stand-alone Java client application to WebSphere MQ over an SSL connection.

Before you begin

- Download the MO04 SSL pack from http://www-01.ibm.com/support/docview.wss?uid=swg24010367 to obtain the sample Java application.
- Complete the instructions in “Creating the channel authentication record” on page 15 to create a channel authentication record to allow client applications to connect to WebSphere MQ.

About this task

Complete the instructions in the subtopics to create a keystore for the Java application, generate the certificates, and test the SSL connection between the sample application and WebSphere MQ.

Configuring the Java keystore and certificates

Configure the keystore to hold the certificate that is used to verify the client to the server.

About this task

Complete the steps in this task on the client host.

Procedure

1. Start the IBM Key Management utility. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
2. Create the Java client repository.
   a. In the **Key Database File** menu, click **New**.
   b. Set the **Key Database Type** to **JKS**.
   c. In the **File Name** field, enter `javaClient`.
   d. Set the **Location** to `C:\Program Files\IBM\WebSphere MQ\javassl\`.
   e. Click **OK**.
   f. In the Password Prompt window, type `passw0rd`, then retype the password.
   g. Select **Stash password to a file**. Click **OK**.

   > **Why am I doing this?**
   > The password must be stashed so that WebSphere MQ and WebSphere Application Server can access it.

   The keystore is created and the home screen opened.
3. Create the certificate request for the Java client.
   a. Click **Create** > **New Certificate Request**.
   b. In the **Key Label** field, type `ibmwebspheremqmyuserid`. `myuserid` must all be lowercase.
   c. Leave the **Key Size** and **Signature Algorithm** fields with their default settings. In the **Common Name** field type `myuserid`. In the **Enter the name of the file** field, edit the file name to `javaClient.req` and click **OK**. All the other fields are optional. A confirmation message is displayed and the certificate request is created.
4. Sign the client certificate request.
   a. Use FTP to transfer `javaClient.req` to the `myCA` directory on the host of the certificate authority.
   b. In a command prompt, navigate to the `myCA` directory and enter `runmqckm -cert -sign -db myCA.kdb -label "myCAcertificate" -expire 365 -format ascii -file javaClient.req -target javaClient.cer`. In the password prompt window, enter `passw0rd`.
5. Use FTP to transfer `javaClient.cer` and `myCertfile.cer` to the client.
   a. In the IBM Key Management utility, select **Personal Certificates** from the menu.

b. Click **Receive**, then click **Browse** and locate `javaClient.cer`.

c. Select **Signer Certificates** from the menu.

d. Click **Add**, then click **Browse** and locate `myCAcertfile.cer`

Results

The required certificates are prepared for secure communication.

What to do next

To test the solution, follow the instructions in "Testing the SSL connection."

Testing the SSL connection

Test the secure connection between the Java client and WebSphere MQ.

About this task

Run the sample program on the client to make a secure connection to WebSphere MQ.

Procedure

1. Locate `SSLSample.class` in `mo04\client_samples\bin\`.
2. In the command prompt, navigate to the directory that contains the `SSLSample.class` file. Enter the command `java SSLSample Server Address Port WAS_SVRCONN sampleQM SSL_RSA_WITH_3DES_EDE_CBC_SHA "C\Program Files\IBM\WebSphere MQ\javassl\javaClient.jks" passw0rd`.

> **Why am I doing this?**
>
> The arguments for the **`MQCONNX`** call are: **`Conname`**, **`Port`**, **`Channel`**, **`Qmgr`**, **`CipherSpec`**, **`SSL keystoreSSL keystore password`**.

Results

If secure communication is achieved, `Connection Successful!` is displayed.

What to do next

To configure WebSphere Application Server to use SSL, follow the instructions in "Configuring WebSphere Application Server to use SSL" on page 21.

Adding WebSphere Advanced Message Security to the solution

WebSphere Advanced Message Security (AMS) expands WebSphere MQ security services to provide protection at the message level. These services ensure that message data is not modified between being placed on the queue and when it is retrieved.

AMS is an example of application level security, acting at the interface between the application and the queue manager. Application level security promotes the confidentiality and integrity of data.

Before you begin

- For this scenario you must have two user accounts on the host with the names *alice* and *bob*. The *HOMEDRIVE* and *HOMEPATH* variables must be set up.
- If WebSphere Advanced Message Security was not installed originally, you must have access to the WebSphere MQ installation image
- You must have the Server and Development Toolkit installed.

Note: For more information about WebSphere Advanced Message Security, see WebSphere MQ Advanced Message Security in the WebSphere MQ product documentation.

Installing WebSphere Advanced Message Security

After you purchase the IBM WebSphere MQ Advanced Message Security license, you can install the component on Windows platforms.

Before you begin

- If WebSphere Advanced Message Security was not installed originally, you must have access to the WebSphere MQ installation image.

Procedure

1. Access the WebSphere MQ installation image. The location might be the mount point of the DVD, a network location, or a local file system directory.
2. Locate the `setup.exe` in the base directory of the WebSphere MQ installation image.
   - From a DVD, this location might be:

```
E:\setup.exe
```

   - From a network location, this location might be:

```
m:\instmqs\setup.exe
```

   - From a local file system directory, this location might be:

```
C:\instmqs\setup.exe
```

3. Double-click the **Setup** icon.
4. Start the IBM WebSphere MQ Installer.
5. Install the WebSphere Advanced Message Security component.
   a. Click **Maintain or upgrade an existing instance**. Click **Next**.
   b. In the Program Maintenance window, click **Modify** then click **Next**.
   c. In the Features window, expand **Advanced Message Security** and click **Install this feature**. Click **Next**.

> **Why am I doing this?**
> The red cross signifies "do not install this feature (and remove if already installed)". Ensure that nothing necessary is removed during this process.

   d. Click **Modify**. Click **Finish** to complete the installation.

Results

WebSphere Advanced Message Security is now installed.

What to do next

To grant users the required authorities to use WebSphere Advanced Message Security, follow the instructions in "Authorizing users."

Authorizing users

Authorize the users to connect, browse, and put messages on the queue. Authorize the users to work with the WebSphere MQ queue manager.

Before you begin

- You must have a command prompt open as a member of the *mqm* user group.

About this task

WebSphere MQ Advanced Message Security requires authorized user accounts to encrypt and decrypt messages. Run the commands from the `\WebSphere MQ\bin` directory.

Procedure

1. Authorize the users to connect to the queue manager.
   a. Enter `setmqaut -m sampleQM -t qmgr -p alice -p bob +connect +inq`
2. Authorize the users to work with the queue.
   a. Enter `setmqaut -m sampleQM -n Q1 -t queue -p alice +put`
   b. Enter `setmqaut -m sampleQM -n Q1 -t queue -p bob +get`
3. Enable the users to browse the system policy queue.
   a. Enter `setmqaut -m sampleQM -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse`
4. Enable the users to put messages on the error queue.
   a. Enter `setmqaut -m sampleQM -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put`

Results

The users now have the required authorities to use the WebSphere MQ objects.

What to do next

To create the key database and certificates, follow the instructions in "Creating key database and certificates."

Creating key database and certificates

Create the key database files and certificates for both users.

Before you begin

You must have a directory in each of the user accounts called */AMS*.

About this task

WebSphere Advanced Message Security requires key databases and certificates to verify users.

Procedure

1. Create alice's key database.
   a. Start the IBM Key Manager. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **IBM Key Management**.
   b. In the **Key Database File** menu click **New**.
   c. Set the **Key Database Type** to **CMS**.
   d. Enter alicekey.kdb as the file name.
   e. Set the **Location** to C:/Documents and Settings/alice/AMS. Click **OK**.
   f. Type passw0rd in the **Password** field, then retype the password.
   g. Select **Stash password to a file**. Click **OK**.
2. Create alice's certificate.
   a. Change the key database view to **Personal Certificates**.
   b. Click **Create** > **New Self-signed certificate**.
   c. In the **Key Label** field type Alice_Cert.
   d. In the **Common Name** field type alice.
   e. In the **Organisation** field type IBM.
   f. In the **Country** field type GB.
3. Repeat these steps for the user bob.

Results

The two users have a self-signed certificate and a key database.

What to do next

To create the configuration file that is used for identifying the location of user's information, follow the instructions in "Creating the keystore.conf file."

Creating the keystore.conf file

Create a directory called .mqs, in the directory create a file called keystore.conf.

About this task

The keystore.conf file stores the location of the key databases and certificates. The WebSphere MQ Advanced Message Security interceptors require this information.

Procedure

1. Create the directory for the keystore.conf file.
   a. Open a command prompt window.
   b. Change directory to *%HOMEDRIVE%\%HOMEPATH%*.
   c. Create a directory called.mqs.
2. Create the keystore.conf file.
   a. Open a text editor and create a file called keystore.conf.
   b. Enter the lines: cms.keystore = %HOMEDRIVE%\%HOMEPATH%\alice\AMS\alicekey and cms.certificate = Alice_Cert. The path to the keystore file must be provided with no file extension.
3. Repeat these steps for bob.

Results

The users have a valid `keystore.conf` file.

What to do next

To authenticate users against each other, follow the instructions in “Sharing the certificates.”

Sharing the certificates

Add each of the user's certificate to the other user's key database.

About this task

Each user's certificate must be added to other user's key database to proving their authenticity.

Procedure

1. Extract `alice's` personal certificate.
   a. In the IBM Key Managementutility, open `alice's` key database.
   b. In the Personal Certificates view, click **Extract Certificate**.
   c. In the **Certificate file name** field, enter `alice_public.arm`. Click **OK**.
2. Add the certificate to `bob's` key database.
   a. In the IBM Key Managementutility, open `bob's` key database.
   b. In the Signer Certificates view, click **Add**.
   c. Locate `alice_public.arm`, click **OK**.
3. Repeat these steps for `bob`.

Results

In the Personal Certificates view, both certificates are visible. Each user can prove the other is authentic.

What to do next

To define the queue policy that is used, see “Defining the security policy.”

Defining the security policy

A security policy defines the format of messages that can be transferred. The encryption and decryption algorithms are defined along with the distinguished names of the sender and receiver.

About this task

Use WebSphere MQ Explorer to define a security policy to allow `alice` and `bob` to send encrypted messages.

Procedure

1. Start WebSphere MQ Explorer. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **WebSphere MQ Explorer**. If WebSphere MQ Explorer is running and displaying the Welcome page, close the Welcome page to start administering WebSphere MQ objects.

2. In the WebSphere MQ Explorer - Navigator pane, expand **Queue Managers**, then expand **sampleQM**.
3. Right-click **Security Policies** and select **New** > **Security Policy**. The New Security Policy window opens.
4. In the **Queue** field, type `Q1`. In the Policy pane, click **Sign and encrypt**. In the Toleration plane, click **Apply this policy to all messages**. Click **Next**.
5. From the **Messaging signing algorithm** list, select **SHA1**. Click **Only accept signed messages from the message orginators listed below**.
6. Click **Add**. The "Add distinguished name" window opens, type `CN=alice, O=IBM, C=GB`. Click **OK**, then **Next**.
7. In the Encryption pane, from the **Message encryption algorithm** list, select **AES256**.
8. Click **Add**. The "Add distinguished name" window opens, type `CN=bob, O=IBM, C=GB`. Click **OK**, then **Next**.

Results

Only `alice` and `bob` can send messages on the queue.

What to do next

To verify that the users can communicate, follow the instructions in "Verifying the solution."

Verifying the solution

Verify that the users can send and receive messages.

About this task

Ensure that communication is possible between both of the users.

Procedure

1. Send a message as the user `alice`.
   a. Right-click `cmd.exe` and click **Run as...**. When prompted, log in as the user `alice`.
   b. As the user `alice` put a message on the queue. In the command prompt type `amqsput Q1 sampleQM`.
   c. Enter `Testing advanced message security communication` as the message. Press Enter twice.
2. Receive the message as the user `bob`.
   a. Right-click `cmd.exe` and click **Run as...**. When prompted, log in as the user `bob`.
   b. As the user `bob` receive the message from the queue. Type `amqsget Q1 sampleQM`.
   c. Receive the message text, `Testing advanced message security communication`.

Results

`Alice` and `bob` can communicate successfully.

What to do next

To prove that the message data is encrypted, follow the instructions in "Testing the encryption."

Testing the encryption

Test the encryption of the message transmitted between the sender and receiver.

About this task

Browse the queue with the receiving user to view the encrypted data.

Procedure

1. Create an alias queue.
   a. Start WebSphere MQ Explorer. Click **Start** > **All Programs** > **IBM WebSphere MQ** > **WebSphere MQ Explorer**. If WebSphere MQ Explorer is running and displaying the Welcome page, close the Welcome page to start administering WebSphere MQ objects.
   b. In the WebSphere MQ Explorer - Navigator pane, expand **Queue Managers**, then expand **sampleQM**.
   c. Right-click **Queues**, then select **New** > **Alias Queue**. The New Alias Queue window opens.
   d. In the name field, type `TESTALIAS`. Click **Finish**. The alias queue is created.
2. Grant `bob` access to browse from the alias queue. In the command prompt, enter the command: `setmqaut -m sampleQM -n TESTALIAS -t queue -p bob +browse`.
3. Send a message as the user `alice`.
   a. Right-click `cmd.exe` and click **Run as...**. When prompted, log in as the user `alice`.
   b. As the user `alice` put a message on the queue. In the command prompt type `amqsput TESTQ sampleQM`.
   c. Enter `Testing advanced message security encryption` as the message. Press Enter twice.
4. Browse the message as the user `bob`.
   a. Right-click `cmd.exe` and click **Run as...**. When prompted, log in as the user `bob`.
   b. As the user `bob` receive the message from the queue. Type `amqsbcg TESTALIAS sampleQM`.

Results

The output from the **`amqsbcg`** application shows the encrypted message data, proving that the solution works.

Example

```
****   Message      ****

 length - 1234 of 1234 bytes

00000000:  5044 4D51 0200 0200 6800 0000 6800 0000 'PDMQ....h...h...'
00000010:  0800 0000 5203 0000 1500 0000 0000 0000 '....R...........'
00000020:  4D51 5354 5220 2020 0000 0000 0000 0000 'MQSTR   ........'
00000030:  0000 0000 0000 0000 5445 5354 5120 2020 '.......TESTALIAS'
```

```
00000040:  2020 2020 2020 2020 2020 2020 2020 2020 '                '
00000050:  2020 2020 2020 2020 2020 2020 2020 2020 '                '
00000060:  2020 2020 2020 2020 3082 0466 0609 2A86 '        0é.f. *å'
00000070:  4886 F70D 0107 03A0 8204 5730 8204 5302 'Hå¸....áé.W0é.S.'
00000080:  0100 3181 CB30 81C8 0201 0030 3130 2931 '..1ü.0ü....010)1'
00000090:  0B30 0906 0355 0406 1302 4742 310C 300A '.0 ..U....GB1.0.'
000000A0:  0603 5504 0A13 0349 424D 310C 300A 0603 '..U....IBM1.0...'
000000B0:  5504 0313 0362 6F62 0204 5044 9CB1 300D 'U....bob..PD£.0.'
000000C0:  0609 2A86 4886 F70D 0101 0105 0004 8180 '. *åHå¸.......üÇ'
000000D0:  1614 18A8 9AB4 9899 9F7C 8F4D 0CB7 5FD3 '...¿Ü.ÿÖƒ|ÅM.À_Ë'
000000E0:  1F33 369F F461 E02B 3BF7 28FE 673C F250 '.36ƒ¶aÓ+;¸(.g<‗P'
000000F0:  E416 303F 4C71 F92F 375A A5B2 A469 B72D 'õ.0?Lq¨/7ZÑ.ñiÃ-'
00000100:  63F5 3ECB 6868 C677 311C 23BB 6E94 B44D 'c§>.hhãw1.#.nö.M'
00000110:  5C51 0A87 9E2C 82B7 04E4 4F6F F010 F086 '\Q.ç×,éÀ.õOo.å'
00000120:  7BFA E383 0ACE 9B2C A131 C529 6264 F54E '{•Òâ..ø,í1.)bd§N'
00000130:  569F 5A5D 064D B64F 6A32 5D35 BA73 7194 'VƒZ].MÂOj2]5.sqö'
00000140:  3F39 314D 28BF C3DD 4444 943D D3D3 40C8 '?91M(..¦DDö=ËË@.'
00000150:  3082 037E 0609 2A86 4886 F70D 0107 0230 '0é.~. *åHå¸....0'
00000160:  1D06 0960 8648 0165 0304 012A 0410 3674 '.. `åH.e...*..6t'
00000170:  0E3E D6ED E588 6A8A 2263 6C39 F415 8082 '.>ÍÝÕêjè"cl9¶.Çé'
00000180:  0350 E267 8463 C813 B01C 67F2 4790 A940 '.PÔgäc....g‗GÉ®@'
00000190:  DE9A B72A 1B47 6B50 DE31 036B 6CCD 4880 'ÌÜÀ*.GkPÌ1.kl.HÇ'
000001A0:  423F B114 6C3E C1FC 8DCB 3252 5C81 D0EF 'B?..l>.³ì.2R\üð '
000001B0:  C342 BBD7 0871 ACAF 05D7 FD4A 4C34 631E '.B.Î.q¼».Î²JL4c.'
000001C0:  2D04 22A3 1117 AA0A A945 C296 75E9 343A '-."ú..¬.®E.ûuÚ4:'
000001D0:  F834 79CD A82E CB71 6ACF 2B7D C757 BC6A '°4y.¿..qj¤+}ÃW.j'
000001E0:  2A14 511F F884 6176 4AD3 6954 29DF 529D '*.Q.°äavJËiT).RØ'

000001F0:  2631 4361 B51B 043F BF03 A540 CD9D BA2B '&1CaÁ..?..Ñ@.Ø.+'
00000200:  2CB2 F7D5 2702 6ACE BE08 FA3E B18A FAF9 ',.¸ı'.j.¥.•>.è•¨'
00000210:  D08A 8584 7ECF 478B 0454 B0EA 4201 3F52 'ðèàä~¤Gï.T.ÛB.?R'
00000220:  7399 4A6A D6EC DB2E F7BC 6F06 297C B112 'sÖJjÍý..¸.o.)|..'
00000230:  09FE 2BAC 603D B5D4 A7AB 1207 DC42 A663 ' .+¼`=ÁÈº½...Bªc'
00000240:  BCCB 621D 757F EBA5 363E 7A53 B01E E6AF '..b.u.ÙÑ6>zS..µ»'
00000250:  C29F 53B0 30A7 BAF3 817B 9439 23D4 752E '.ƒS.0º.¾ü{ö9#Èu.'
00000260:  1D56 A44F C0A8 274E EA56 4AAA 7F7A 497E '.VñO.¿'NÛVJ¬.zI~'
00000270:  DCAC E79F 5471 53C5 00C4 F88D F81C F2D2 '.¼þƒTqS...°ì°.‗Ê'
00000280:  B0F0 85EE 9745 0B35 F62C E1C6 D161 3E5A '.à¯ùE.5÷,ßãÐa>Z'
00000290:  7878 DE3E 75FF B3B2 9D5D 29FA C76E 33E1 'xxÌ>u  ..Ø])•Ãn3ß'
000002A0:  DB91 748B 3308 9DC0 BF22 10FD AC10 433A '.ætï3.Ø..".²¼.C:'
000002B0:  3D18 4CEE E6BB A594 A106 0BFE 2B70 4EF7 '=.L¯µ.Ñöí...+pN¸'
000002C0:  9BE2 46E4 A2DC 56E6 A44E 46C2 4844 231F 'øÔFõó.VµñNF.HD#.'
000002D0:  ACD3 F36E C71F 2558 240F 41B1 1231 FF02 '¼Ë¾nÃ.%X$.A..1  .'
000002E0:  66E6 FB5C C47E 81F9 8678 4F93 0EE3 8D6E 'fµ¹\.~ü¨åxOô.Òìn'
000002F0:  5796 400D AE95 2C59 171A 7DDB E1FC 475F 'Wû@.«ò,Y..}.ß³G_'
00000300:  2E25 B8D8 88BC 4EBD 7422 ABD7 2F80 8232 '.%©Ïê.N¢t"½Î/Çé2'
00000310:  AC31 4503 E4AE C56B 48F0 04AC 86E0 8DD2 '¼1E.õ«.kH.¼åÓìÊ'
00000320:  DB97 D6BF 6397 4630 93F2 26CB 3EB3 4F6E '.ùÍ.cùF0ô‗&.>.On'
00000330:  4344 348D DD77 01DF 6FF1 A987 6D04 DD68 'CD4ì¦w..o±®çm.¦h'
00000340:  FA45 EDC5 75EC AB3B 511D 22DD E937 AA0A '•EÝ.uý½;Q."¦Ú7¬.'
00000350:  9BA9 BB65 1A67 D379 B69A 7959 D18F 2C07 'ø®.e.gËyÂÜyYÐÅ,.'
00000360:  BCF1 9681 F984 20EB 561D 67BB 49EA 238A '.±ûü¨ä ÙV.g.IÛ#è'
00000370:  4DD0 3286 8BCB E312 E6C8 FBF8 2376 CDB5 'Mð2åï.Ò.µ.¹°#v.Á'
00000380:  BB40 D4BB 9538 6D78 436D D49F 49DD 1296 '.@È.ò8mxCmÈƒI¦.û'
00000390:  06B2 1886 624E B29C 0161 F5B2 60F3 3B38 '...åbN.£.a§.`¾;8'
000003A0:  1D75 ADCC E318 FFE2 8168 5622 85A2 92A7 '.ui.Ò.  ÔühV"àóÆº'
000003B0:  75B2 A83B 4397 5D80 4732 73B8 14AE 69FC 'u.¿;Cù]ÇG2s©.«i³'
000003C0:  B568 BF3A C243 5EE1 A9AE 29CE 2AD1 7255 'Áh.:.C^ß®«).*ÐrU'
000003D0:  FFFA D7D2 36D3 AE29 8603 6CE0 20D9 0ACE '  •ÎÊ6Ë«)å.lÓ ...'
000003E0:  B965 6F9A 0E1E EAE5 6A02 1A50 87B2 A3C9 '.eoÜ..ÛÕj..Pç.ú.'
000003F0:  E49E AA02 B3B0 947B FDA9 CB10 D992 8EB1 'õ×¬...ö{²®...ÆÄ.'
00000400:  BFF4 7ABD 5293 8447 B8C5 DC92 4ACC 9F2B '.¶z¢RôäG©..ÆJ.ƒ+'
00000410:  CC46 9C5E A5A2 9B40 0715 8983 975D ED89 '.F£^Ñóø@..ëâù]Ýë'
00000420:  086D 1416 ED17 638A 2AF5 9237 A8C1 32A7 '.m..Ý.cè*§Æ7¿.2º'
00000430:  1314 9E74 4B8A 283B 4130 5A68 01DC 8947 '..×tKè(;A0Zh..ëG'
00000440:  19DA 20B1 BC6D 797B D358 BF2A E7E3 BAED '.. ..my{ËX.*þÒ.Ý'
00000450:  398D 9B82 A59B 664E 0070 685B C486 E1CC '9ìøéÑøfN.ph[.åß.'
```

```
00000460:  612B 2864 98DB A6A0 5FC5 5666 3F34 277D 'a+(dÿ.ªá_.Vf?4'}'
00000470:  E997 90EE 2D24 E72C FFAA 39A7 2EFA 325D 'ÚùÉ¯-$þ,¯¬9º.•2]'
00000480:  3DDE A581 C896 3AA3 EFE5 AB99 CB46 A8A5 '=ÌÑü.û:ú´Õ½Ö.F¿Ñ'
00000490:  567F 47BB 646A 5106 93A6 D53F 1A7F 9067 'V.G.djQ.ôªı?..Ég'
000004A0:  B98E EC96 9547 3F72 A4B9 17C2 6FBA C810 '.ÄýûòG?rñ...o...'
000004B0:  BE8E 5FE7 F28F AD2F 5472 2573 E80D 9002 '¥Ä_þ‗Å¡/Tr%sÞ.É.'
000004C0:  6BBB EC69 A663 F96B 03E3 D6B0 3FF4 97C4 'k.ýiªc¨k.ÒÍ.?¶ù.'
000004D0:  E5D6                                    'ÕÍ              '


 No more messages
 MQCLOSE
 MQDISC
```

Note: The encryption might contain different characters.

Notices

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

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).

Sending your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

- Send an email to ibmkc@us.ibm.com
- Use the form on the web here: www.ibm.com/software/data/rcf/

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number
- The topic and page number related to your comment
- The text of your comment

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

Thank you for your participation.

Printed in USA