



| IBM Software Group

# Implementing SSL with HTTP nodes in Websphere Message Broker V6.x

**Vivek Grover**

WebSphere Message Broker Level 2 Support, IBM

*vgrover@us.ibm.com*

A horizontal banner featuring a series of small, square images. From left to right, it includes a green square, a yellow square, a red square, a purple square, a cyan square, a grayscale image of a highway, a circular arrow icon, a grayscale image of a person's face, a grayscale image of a hand, and several grayscale squares of varying shades.

@business on demand.

WebSphere® Support Technical Exchange

© 2007 IBM Corporation

# Agenda

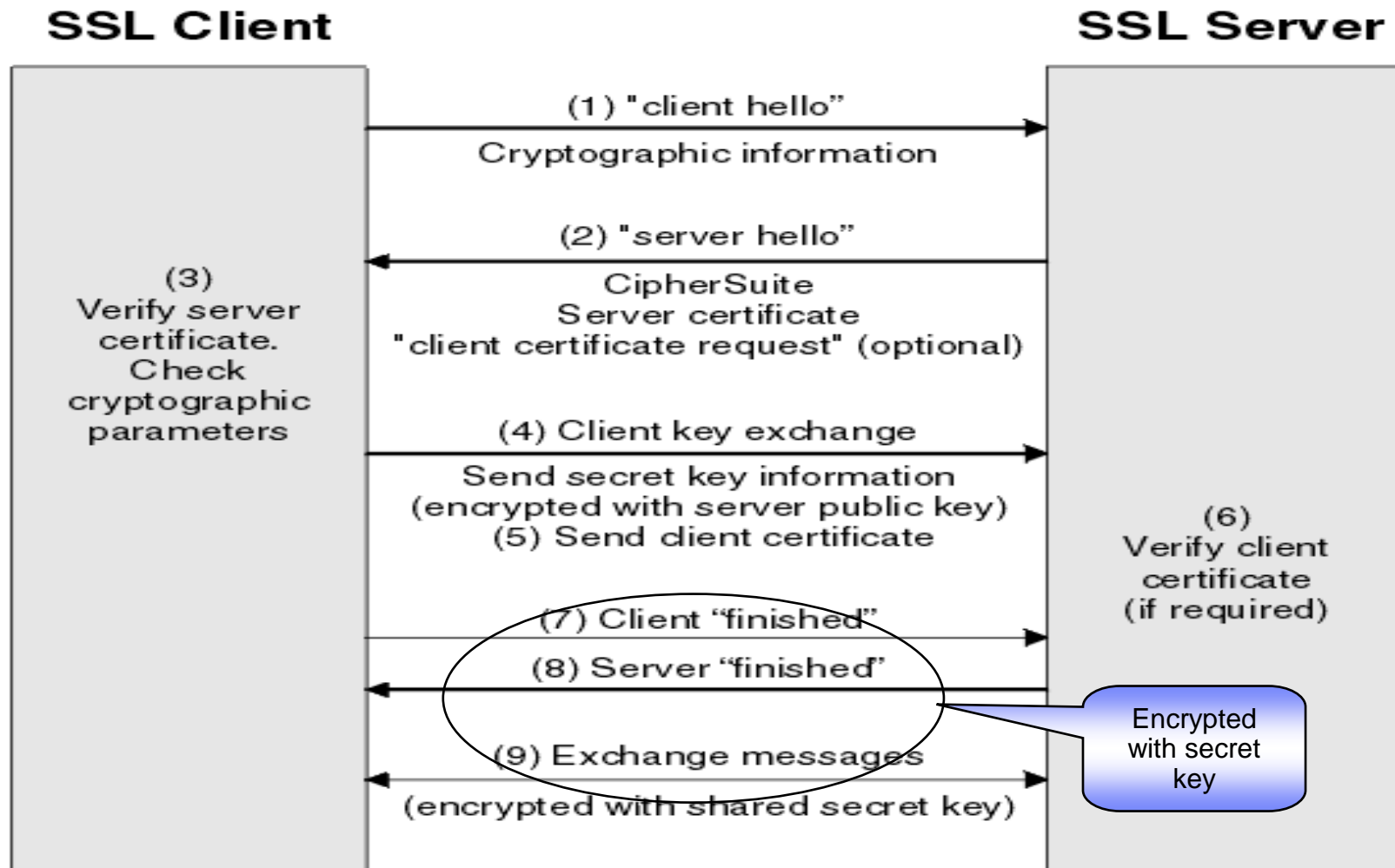
- **SSL Basics**
- **HTTP Basics**
  - ▶ **HTTPRequest node**
  - ▶ **HTTPInput node & HTTPReply node**
- **HTTPS Basics**
- **Certificate management tools**
- **Configuration of HTTPRequest nodes**
- **Configuration of HTTPInput nodes and HTTPReply nodes**
- **Troubleshooting tips**

# SSL Basics

- Protocol developed by Netscape to manage the security of message transmission on the Internet
- Creates a secure connection between a client and a server, over which any amount of data can be sent securely
- Useful terms –
  - ▶ **KeyStore** – File or Database that stores the keys and digital certificates
    - Example – cacerts file located in - `C:\ProgramFiles\IBM\MQSI\6.0\jre\lib\security`
  - ▶ **Digital Certificates** – Provides security against impersonation by binding the key to its owner
    - Contains Public key + Information about the Owner and/or CA
    - Example - verisignserverca
  - ▶ **CipherSuites** – Set of algorithms providing means of Encryption, Hash (MAC) and Key exchanges
    - Example – DES\_SHA\_EXPORT



# SSL Basics



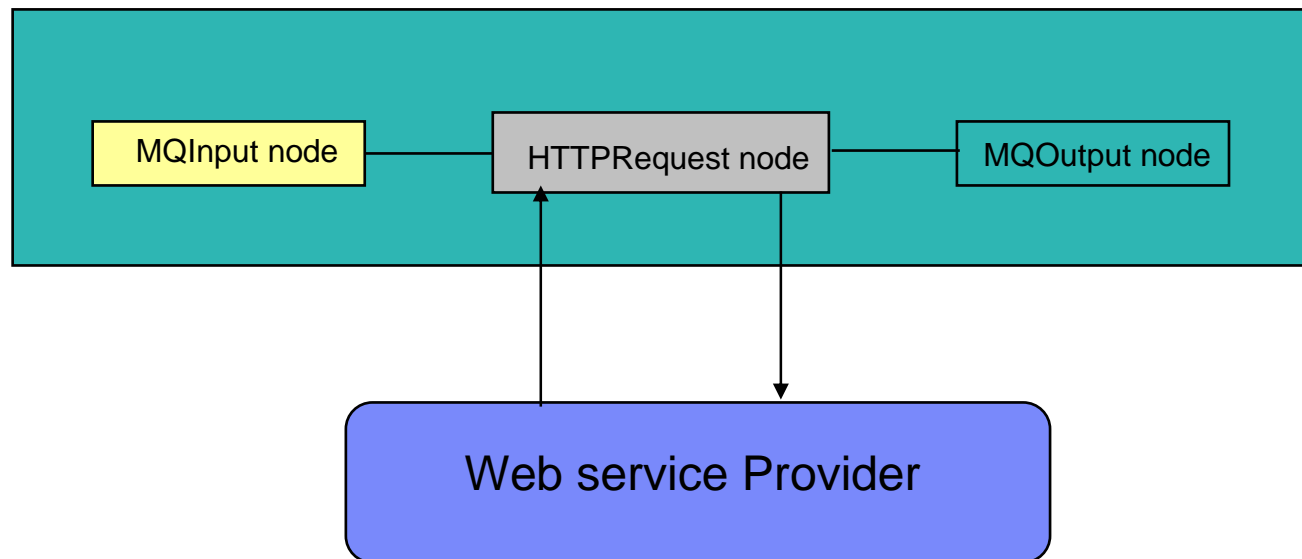
# HTTP Basics

- Communications protocol used to transfer information on the World Wide web
- Useful questions -
- What is a Web service?
  - ▶ A standard way to allow functions/methods to be invoked using HTTP
  - ▶ Used for program to program interactions
  - ▶ Uses the XML, SOAP, WSDL and UDDI open standards over an Internet backbone
- How does WebSphere Message Broker fit in?
  - ▶ Convenient central point for Web services brokering – eg. Transform WSDL definitions or act as a SOAP intermediary etc.
  - ▶ Message Flow can be a requester (Client) – calls out to a Web Service
  - ▶ Message Flow can be a Service provider – lets Web Service clients to invoke it or other flows
  - ▶ Uses HTTPInput node, HTTPReply node, HTTPRequest node

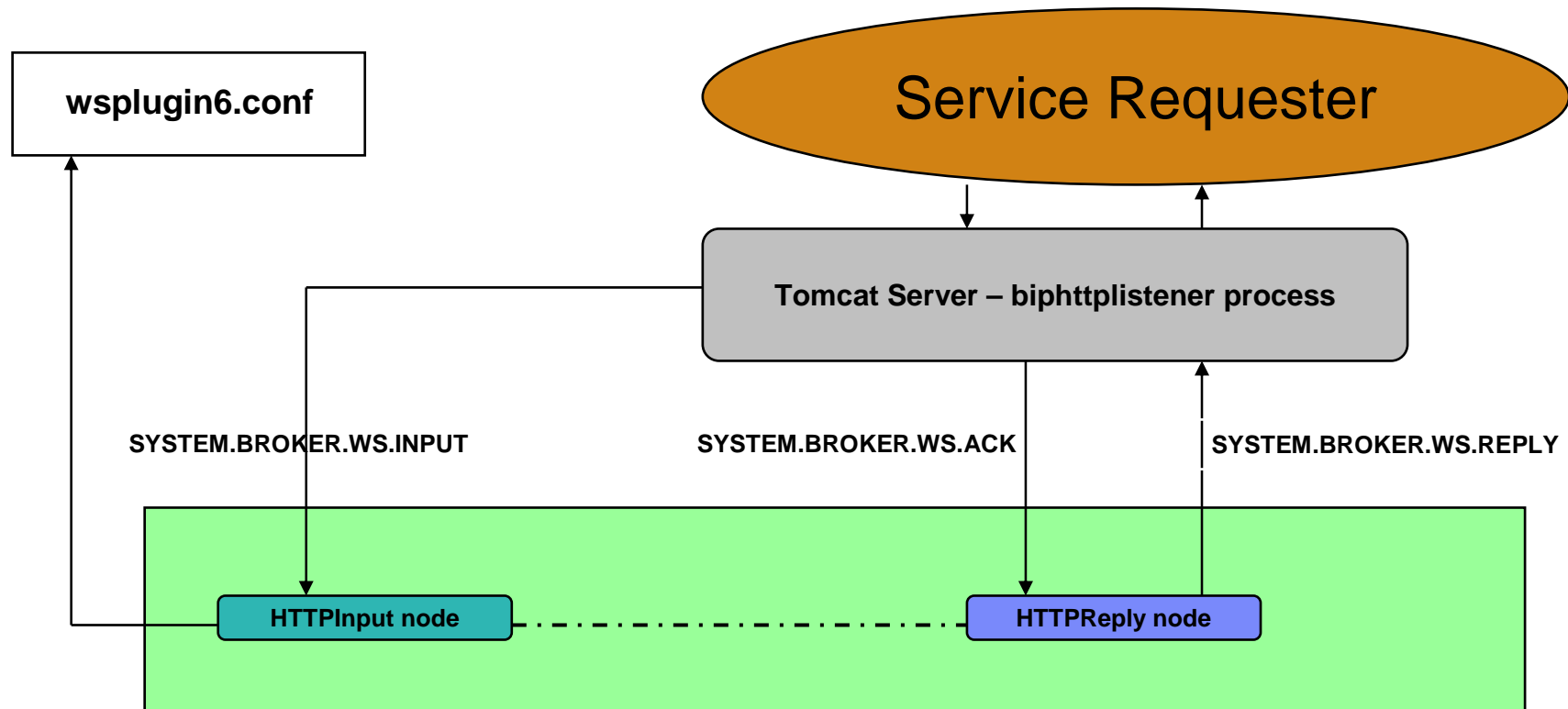


## Web Service Requester

- HTTPRequest node serves as the gateway from a message flow to the broker network invoking Web services within the network
- Sends requests and receives responses from a Web service provider



## Web Service Provider



- Uses HTTPInput node to receive Web service requests from clients at a certain port
- Uses HTTPReply node to send replies back to the clients
- Uses Internal Tomcat servlet engine running as biphttplistener process

# HTTP + SSL = HTTPS

- HTTP over SSL or SSL over HTTP
- Secure messaging over HTTP
- Handled by Tomcat Servlet (Internal or External) - when Web service provider
- Relies on Java JSSE code in JVM
- Must use HTTP/1.1 to implement SSL support
- Uses Port#7083 by default for SSL
- Broker uses CACERTS keystore located in - C:\ProgramFiles\IBM\MQSI\6.0\jre\lib\security





# Certificate Management Tools

- Enables users to administer their public/private key pairs and associated certificates used in SSL
- Different CMTs are available:
  - ▶ Keytool – Tested and supported with WMB V6.0
    - Supplied with IBM JRE
    - Default location for WMB V6 on Windows - C:\Program Files\IBM\MQSI\6.0\jre\bin
    - Default location for WMB V6 on UNIXes - /opt/IBM/mqsi/6.0/jre/bin
    - Command line tool
  - ▶ Ikeyman – Known to work and is supported with WMB V6.0
    - Supplied with IBM JRE
    - Default location for WMB V6 on Windows - C:\Program Files\IBM\MQSI\6.0\jre\bin
    - Default location for WMB V6 on UNIXes - /opt/IBM/mqsi/6.0/jre/bin
    - GUI-based tool

## Configuration of HTTPRequest nodes

- **Configure the HTTPRequest node for SSL with server authentication –**
  - ▶ The receiver of the requests will present certificates to the Broker and the Broker will validate them with the signer's certificates stored in the "cacerts" keystore
  - ▶ Add signer's (or trusted) certificates to the existing "cacerts" keystore –

Using "keytool" -

```
keytool -import -alias mykey -file <name of certificate file> -keystore  
"C:\Program Files\IBM\MQSI\6.0\jre15\lib\security\cacerts" -storepass  
changeit
```

file – Name of the certificate being imported

alias – Name it will show in the keystore

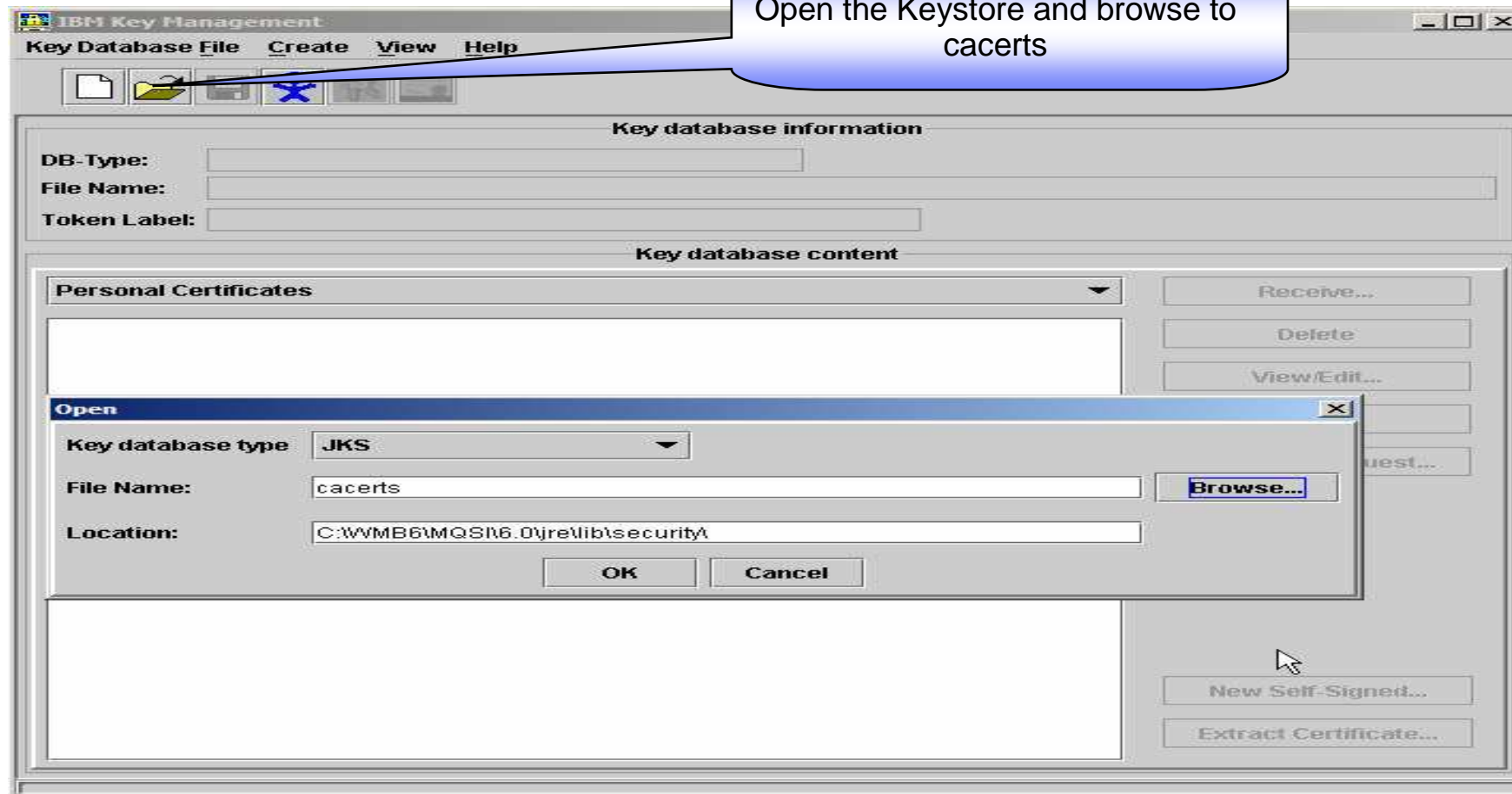
keystore – Keystore DB being used

storepass – password for the above keystore



## Configuration of HTTPRequest nodes

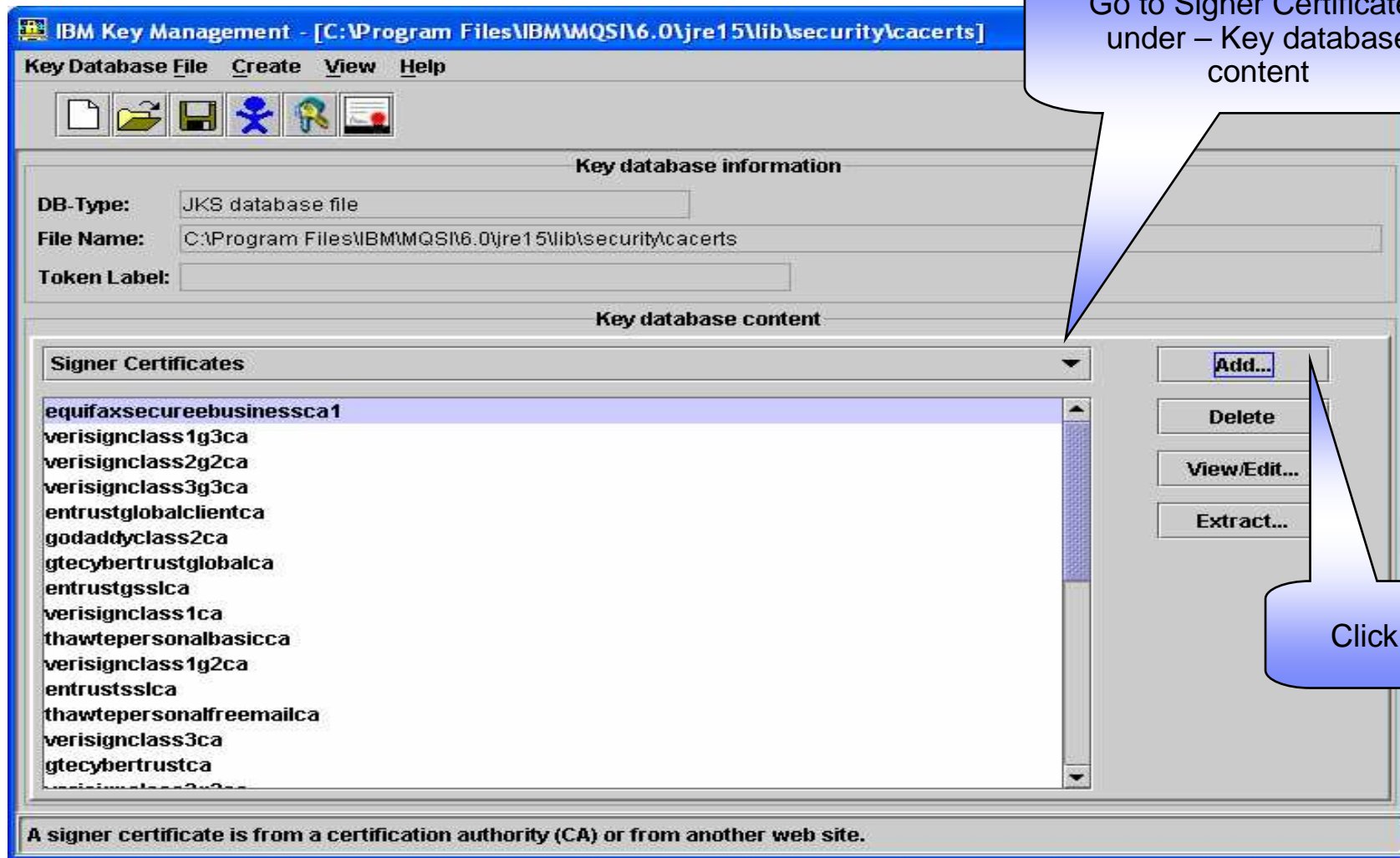
- Using “ikeyman”



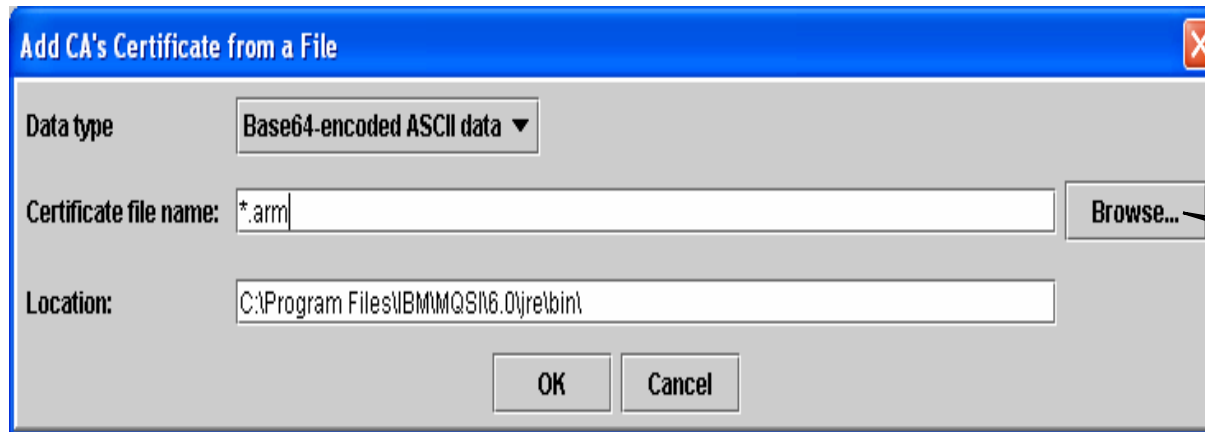
- The password for this file is – “changeit”



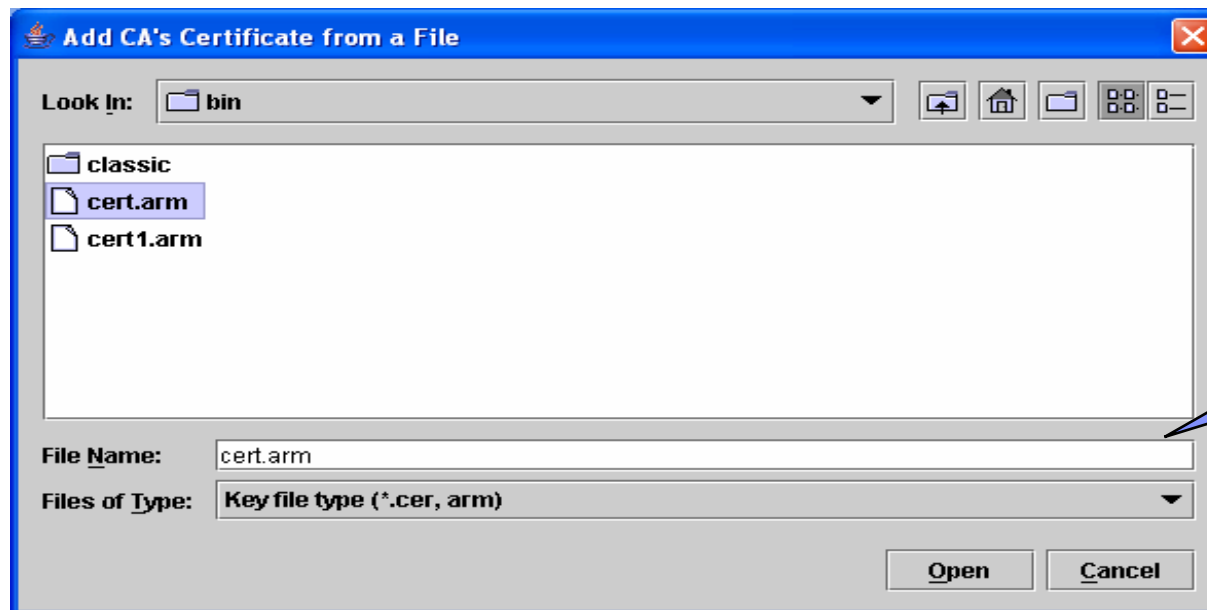
# Configuration of HTTPRequest nodes



# Configuration of HTTPRequest nodes



Browse to the certificate file



Select the Certificate to be added and click open



## Configuration of HTTPRequest nodes

- ***Configure the HTTPRequest node for SSL with mutual authentication***
  - ▶ The receiver and the sender of the requests need to present their certificates to each other and each end will validate using the local copies of the signer's certificates
    - Create a keystore file
    - Create a self-signed certificate (or use a CA certificate)

Using keytool -

**keytool -genkey -storepass <password> -keystore <keystore file> -alias <self-signed certificate>**

Answer the options it asks you while creating the certificate

- Ensure the server keystore contains the above created certificate
- Ensure that the servers signers certificate is imported into cacerts on broker side (extracted as discussed later)



## Configuration of HTTPRequest nodes

- For V6.0 - Update the mqsiprofile.cmd to add the following environment variables with the location of the keystore and the password –

**IBM\_JAVA\_OPTIONS=**

**-Djavax.net.ssl.keyStore=<keystore\_path>/<keystore\_filename>**

**-Djavax.net.ssl.keyStorePassword=<keystore\_password>**

- Stop and start the broker

## Configuration of HTTPRequest nodes

- For V6.1 – Run the mqsichangeproperties command to point the broker to the keystore file –

**mqsichangeproperties <Broker name> -o BrokerRegistry –  
n brokerKeystoreFile -v <Fully qualified name of the new Keystore>**

- The password can be changed using mqsisetdbparms command –

**mqsisetdbparms <Broker name> -n brokerkeystore::password -u temp -p  
<password>**

The user ID (-u) can be any value

- Stop and Start the broker

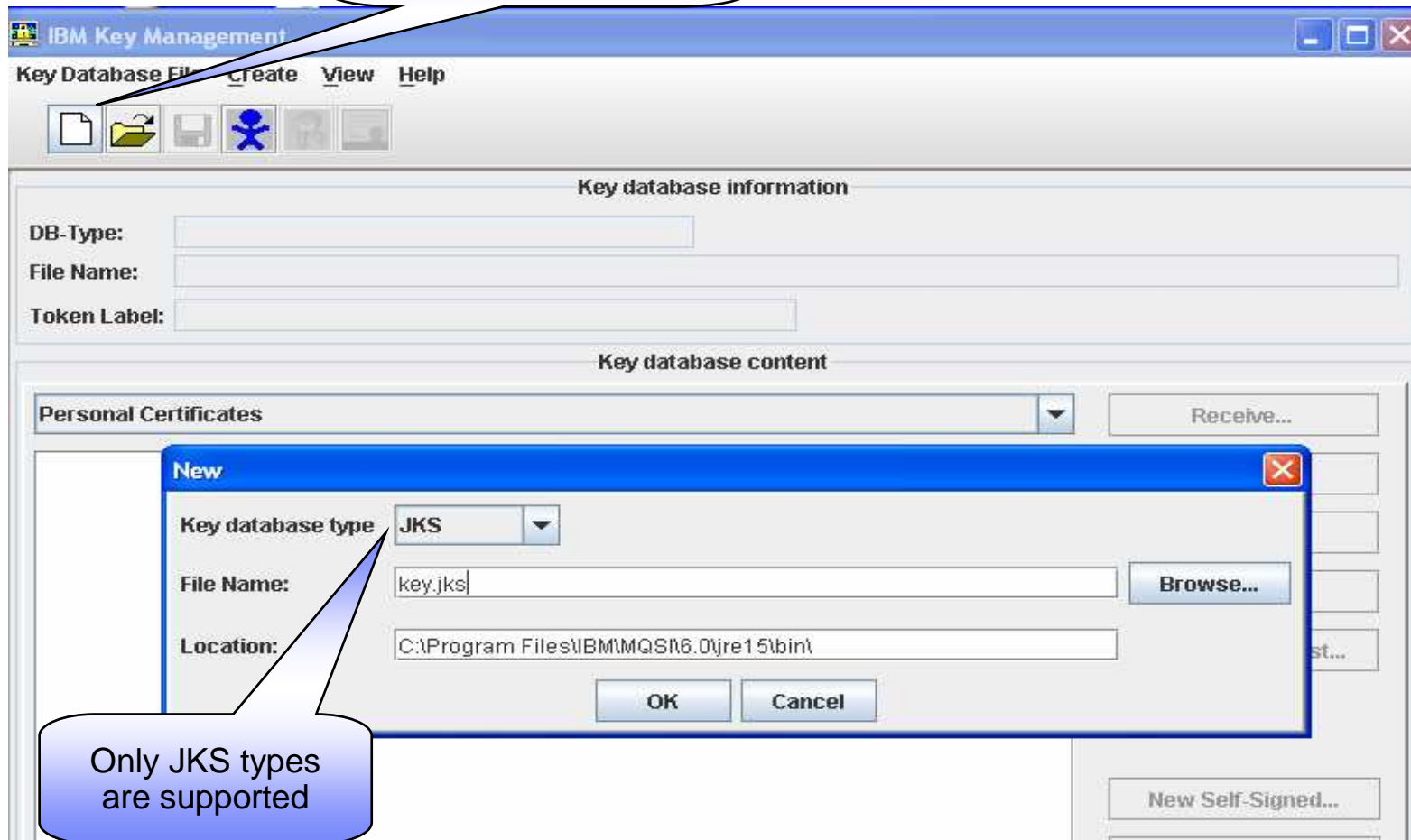




# Configuration of HTTPRequest nodes

- Using ikeyman-

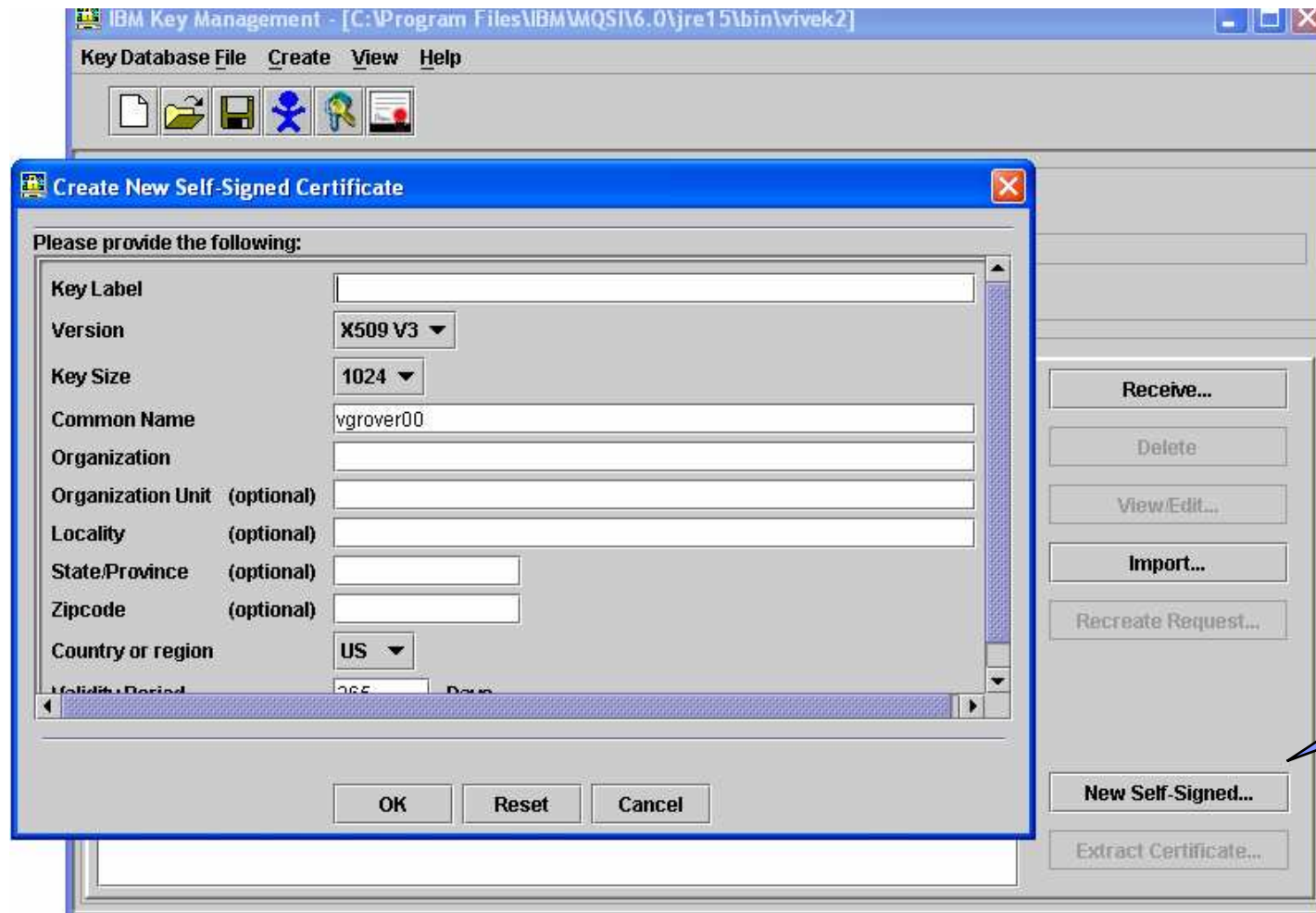
Create a new keystore file



Only JKS types are supported



# Configuration of HTTPRequest nodes



## Configuration of HTTPRequest nodes

- **Configure the node in the Toolkit**
  - ▶ Ensure that the destination URL starts with **https**
  - ▶ In the SSL tab (following options) –
    - **Protocol** - SSL – Try SSLv3 first, allows fallback on SSLv2
      - SSLv3 – Try SSLv3 only
      - TLS (Transport Layer Security) – try TLS only
    - **Allowed SSL ciphers**
      - default of empty means all broker JVM supported ciphers
      - Can specify 1 or more ciphers
- **Configure the bar file in the Toolkit**
  - Same above mentioned properties can be configured via bar file
    - Protocol, URL, Ciphers



# Configuration of HTTPInput & HTTPReply nodes

- ***Configure the HTTPInput & HTTPReply nodes for SSL with server authentication***

- ▶ biphttplistener is used to receive HTTP requests on behalf of any message flow that is using HTTPInput nodes
- ▶ The responses generated by HTTPReply nodes are also handled by the biphttplistener
  - Create a keystore file for the broker
  - Create a self-signed certificate (for testing SSL)

Using keytool -

**keytool -genkey -storepass <password> -keystore <keystore file name> -alias <self-signed certificate>**

- Change the broker properties to set the following:



## Configuration of HTTPInput & HTTPReply nodes

- Enable the HTTPSConnector:

```
mqsichangeproperties <broker_name> -b httplistener -o HTTPListener -n  
enableSSLConnector -v true
```

- Point the broker to above created keystore

```
mqsichangeproperties <broker_name> -b httplistener -o HTTPSConnector -n  
keystoreFile -v <keystore file name>
```

- Set the keystore password

```
mqsichangeproperties <broker_name> -b httplistener -o HTTPSConnector -n  
keystorePass -v <MyKeystorePass>
```

- Set the Port # (if 7083 is busy)

```
mqsichangeproperties broker name -b httplistener -o HTTPSConnector -n  
port -v <Port to listen on for https>
```

- Use the following commands to verify and display the HTTP Listener properties:

```
mqsireportproperties <broker_name> -b httplistener -o HTTPListener -a
```

```
mqsireportproperties <broker_name> -b httplistener -o HTTPSConnector -a
```



## Configuration of HTTPInput & HTTPReply nodes

- Extract the certificate to be imported onto client's machine

**keytool -export -alias tomcat -file <name of certificate file> -keystore  
<keystore file> -storepass <password>**

- alias – whatever alias is specified when creating the certificate in the keystore

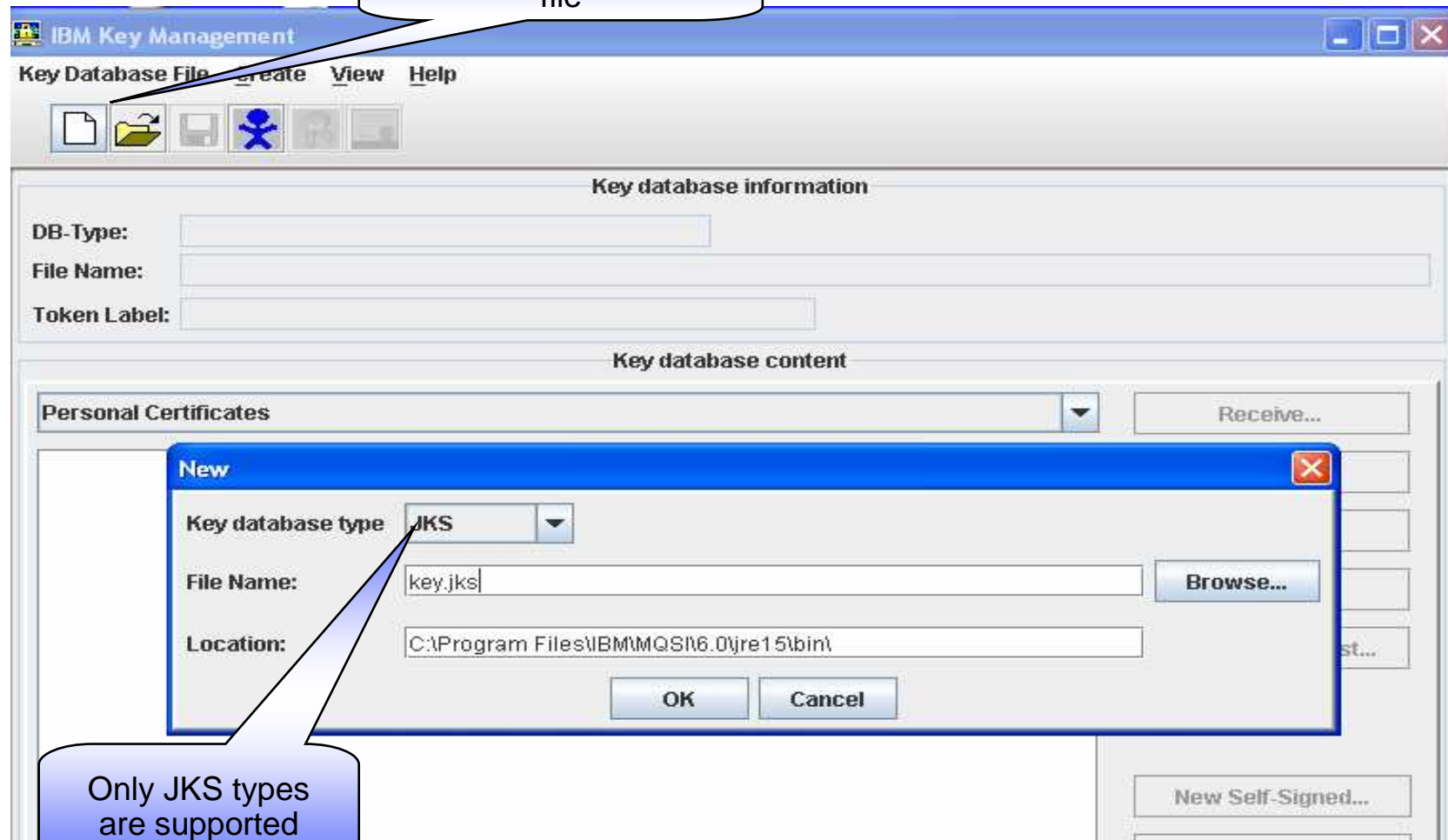
- Send the certificates file to the client machine to be imported into its keystore



# Configuration of HTTPInput & HTTPReply nodes

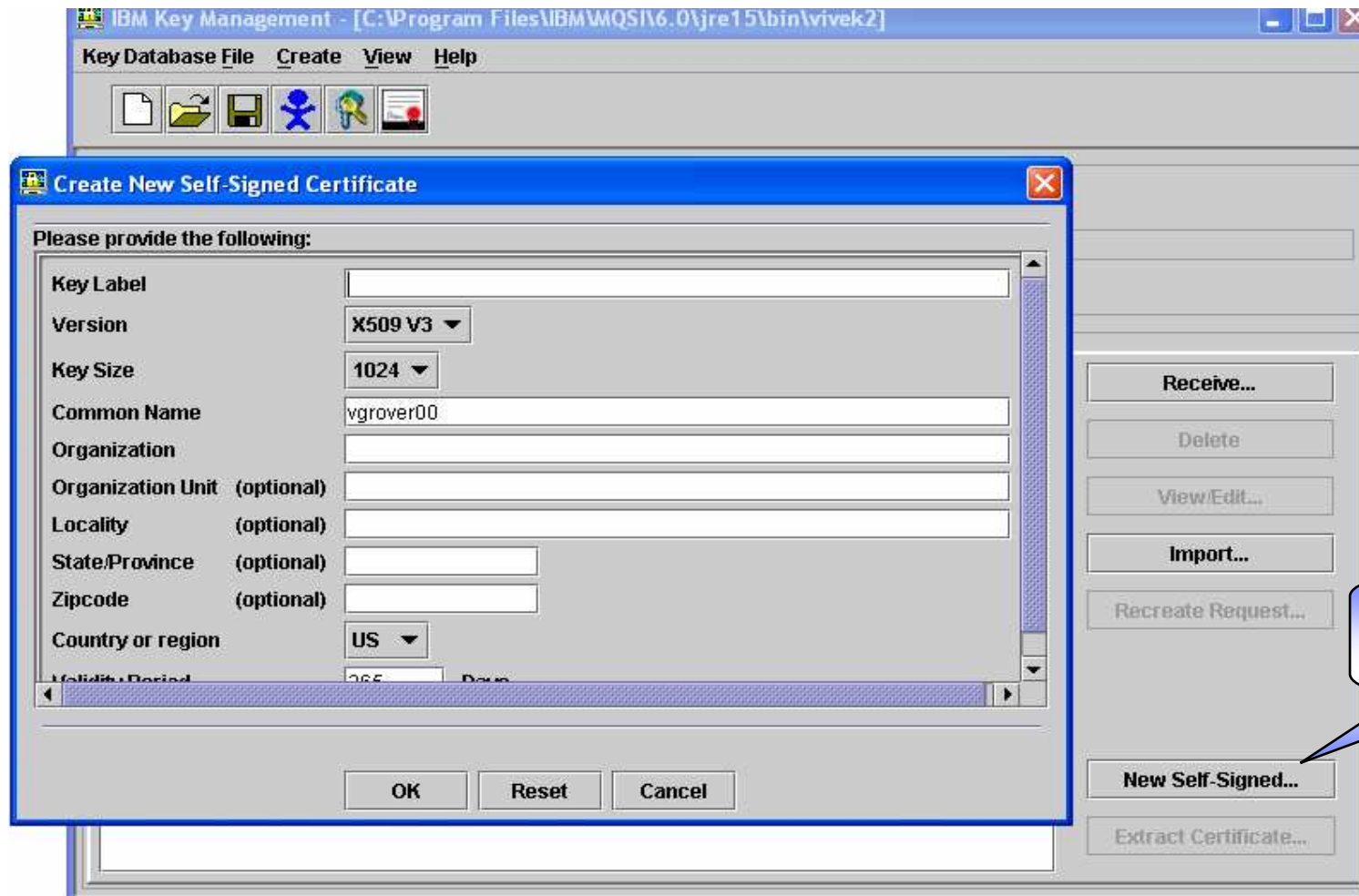
Using ikeyman

Create a new keystore  
file



Only JKS types  
are supported

# Configuration of HTTPInput & HTTPReply nodes

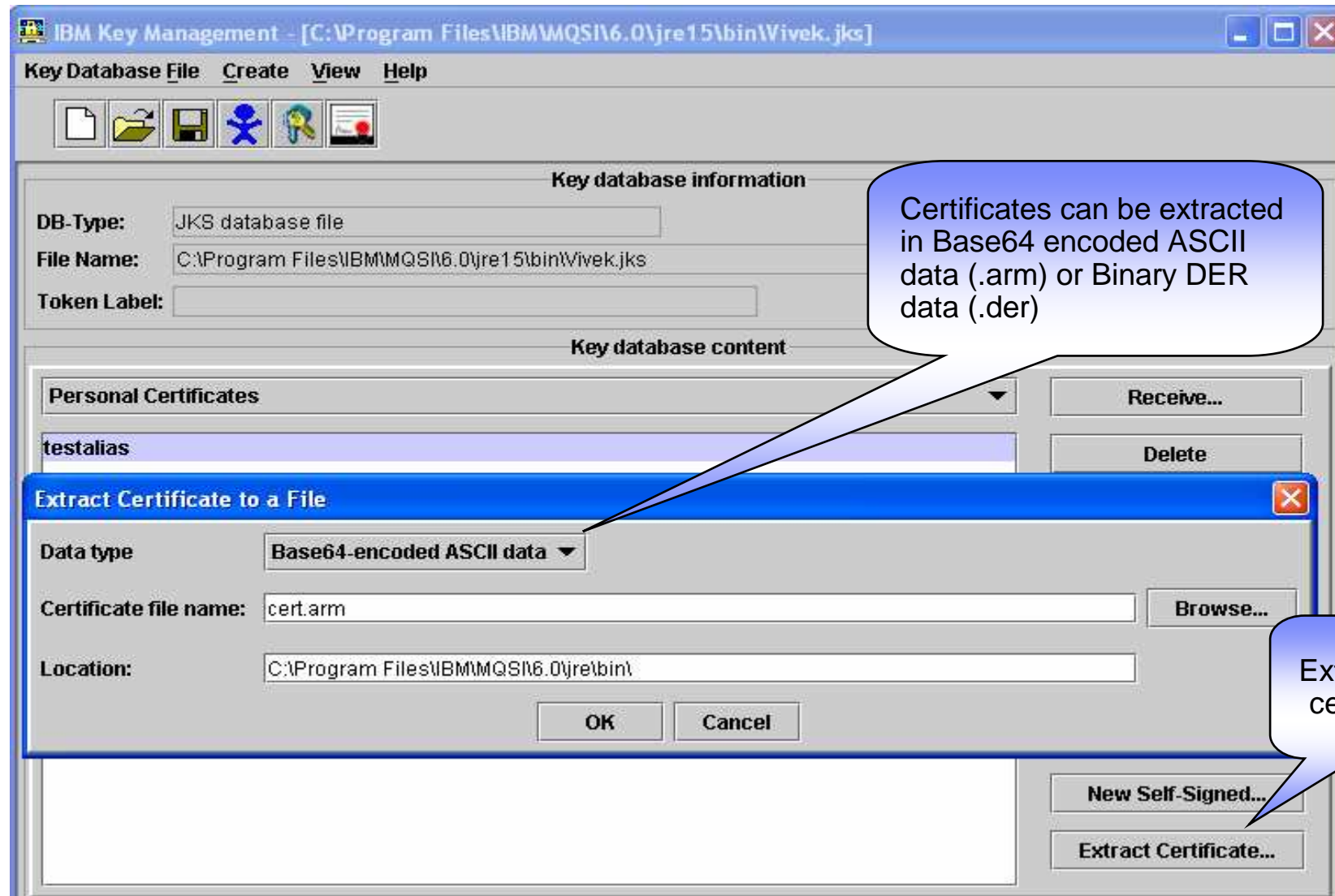


Create new Self-Signed certificate





# Configuration of HTTPInput & HTTPReply nodes



## Configuration of HTTPInput & HTTPReply nodes

- ***Configure the HTTPInput & HTTPReply nodes for SSL with mutual authentication***
  - ▶ Follow the above slides # 20 -25
  - ▶ Enable Client Authentication for broker listener
    - **mqsichangeproperties <broker\_name> -b httplistener -o HTTPSCConnector -n clientAuth -v true**
  - ▶ The Trusted (Signer or CA) Certificates from the client must be added to the broker's default keystore – cacerts

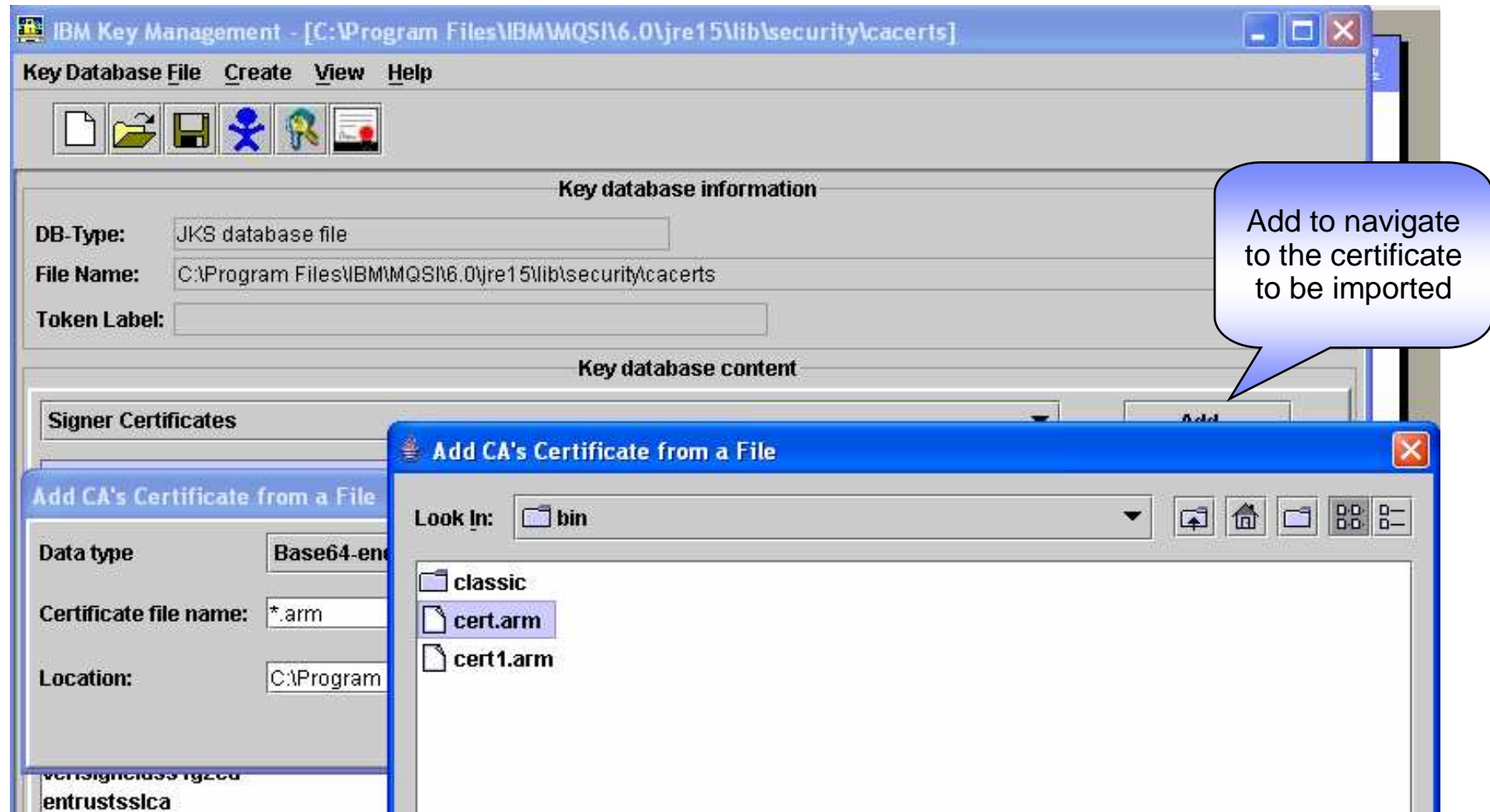
Using keytool -

**keytool -import -alias mykey -file <name of certificate file> -keystore cacerts -keypass changeit**



# Configuration of HTTPInput & HTTPReply nodes

Using ikeyman

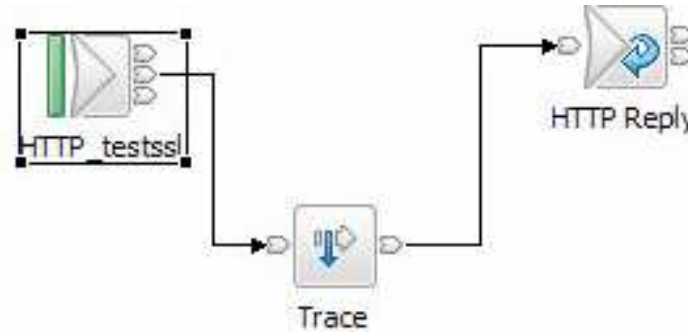


## Configuration of HTTPInput & HTTPReply nodes

- Configure the nodes in the Toolkit
  - ▶ HTTPInput node Properties
    - Select “Use HTTPS”
    - Path suffix for URL – the path part of the URL from which this node receives Web service requests (Not the full URL).
      - For example, specify /path/to/service, where the full URL is http://server/path/to/service or If the URL is http://server/testHTTPS then “testHTTPS”
  - ▶ HTTPReply node Properties
    - There are no parameters to be configured
- Configure the bar file in the Toolkit
  - ▶ Ensure that “Use HTTPS” box is checked for the HTTPInput node
- Deploy and Confirm with – BIP3132I message in the logs indicating https listener has been enabled



# Configuration of HTTPInput & HTTPReply nodes



- Test the configuration –

- Start a web browser and type the URL: <https://localhost:7083/<Path suffix>>
- Accept the certificate when the pop up Window appears and it shows -  
“XML document must have a top level element. Error processing resource <https://localhost:7083/<Path suffix>>”



## Troubleshooting configuration issues

- Ensure the message flow works fine with HTTP configured only
- Ensure the certificates have been imported in the correct keystore files located in the correct directories
- Ensure the certificates are X.509
- Ensure the keystore are in .jks format
- Capture the traces (if needed)

# Tracing

- When Broker is the Client (with HTTPRequest node) –
  - ▶ Collect EG service trace
  - ▶ Collect JSSE trace
  
- When Broker is the server (with HTTPInput node) –
  - ▶ Collect the EG service trace
  - ▶ Collect the biphttplistener trace

## Additional Resources

- [http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/topic/com.ibm.etools.mft.doc/ap12234\\_.htm](http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/topic/com.ibm.etools.mft.doc/ap12234_.htm)
- <http://dev2dev.bea.com/pub/a/2006/08/pfx-pem-certificate-formats.html>
- [http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.csqzas.doc/sy11560\\_.htm](http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.csqzas.doc/sy11560_.htm)
- <http://www-306.ibm.com/software/integration/wbimessagebroker/support/>
- Security Guide - <Install dir>\jre\docs\sdkGuides\securityguide.win32.htm
- How to setup SSL for the HTTP nodes WebSphere Message Broker V6 – Vicente Suarez
- SSL Basics WSTE – Russ Stancliffe
- SSL – Everett Turner





# Questions and Answers