CHAPTER 9

Configuring SSL Messaging

This section discusses the procedure to configure SSL messaging on a TES 6.2 system. TES uses Java Messaging Service (JMS) to implement communications among its components, including:

- Primary Master
- Remote Master
- Backup Master
- Fault Monitor
- Client Manager

This document discusses SSL configuration for each of these components.

Obtaining Server Keys and Certificates

You will need a pair of server key and certificate for each of the following components:

- Client Manager
- Primary Master

If you are setting up a Remote Master, you will need a pair of server key and certificate for it too.

If you are setting up a fault tolerant system, you will also need a pair of server key and certificate for each of the following components:

- Backup Master
- Fault Monitor

All of these servers require keys and certificates be stored in Java Keystore (JKS) files.

You may generate key and certificate by yourself or obtain them from a trusted certificate authority (CA).

1. Generating Key and Certificate

   There are various tools that allow you to generate keys and certificates, among them the Java Keytool that comes with JRE installation.

   Java Keytool Example: generating key & certificate in a keystore

   keytool -keystore my_keystore -alias my_alias -genkey -keyalg RSA

   You can use the keys and certificates you generate to get your implementation and testing going quickly. However, to set up a production grade server, it's recommended you request a well known certificate authority (CA) to sign the keys and certificates.

2. Obtaining Key and Certificate from a Trusted CA

   There are many trusted CA's, such as AddTrust, Entrust, GeoTrust, RSA Data Security, Thawte, VISA, ValiCert, Verisign, beTRUSTed. Each CA has its own instructions which should be followed (look for JSSE section), but all will involve a step to generate a certificate signing request (CSR).

   Java Keytool Example: generating CSR

   ```
   keytool -certreq -alias my_alias -keystore my_keystore -file my_csr.csr
   ```

3. Exporting and Importing Certificate

   When SSL messaging is enabled, each of TES servers will only send messages to and accept messages from the servers it trusts. To authorize messaging between two servers, you must make sure the certificate of one server is registered in the other's trust store, and vice versa. Java Keytool provides certificate import and export options to help you accomplish this goal.

   Java Keytool Example: exporting certificate from a key store to a file

   ```
   keytool -export -alias my_alias -file my_cer.cer -keystore my_keystore -storepass
   my_keystore_password
   ```

   Java Keytool Example: importing certificate from a file to a trust store

   ```
   keytool -import -v -trustcacerts -alias my_alias -file my_cer.cer -keystore my_truststore
   -storepass my_truststore_password
   ```

   Each of the following sections describes configuration for each TES server. It will indicate what other TES server's certificates must be imported into TES server's trust store.

Configuring SSL on the Primary Master

In this section, you will enable SSL on the Primary Master with the key stores you obtained from earlier section.

To enable:

Step 1 Shut down the Primary Master.

Step 2 Copy the key store for the Primary Master to the **config** directory in the Master's installation directory.

Step 3 Create a trust store by importing Client Manager's certificate. Follow the instructions in "Obtaining Server Keys and Certificates".

If you are setting up Remote Master, import the certificate of the Remote Master into this trust store too.

If you are setting up a fault tolerant system, import the certificate of the Fault Monitor into this trust store too.

When done, copy the trust store to the **config** directory in the Master's installation directory.

Step 4 Use a text editor to open the property file *master.props* located in the Master's installation directory.

Note It may be a good idea to back up this file before editing it to ensure there is a good copy to fall back to.

Step 5 In the editor, locate the segment of SSL properties that looks like the following.

#MessageBroker.SSL.enabled=Y

#MessageBroker.SSL.keyStore=

#MessageBroker.SSL.keyStorePassword=

#MessageBroker.SSL.keyPassword=

#MessageBroker.SSL.trustStore=

#MessageBroker.SSL.trustStorePassword=

If such segment can't be found, manually insert these lines.

Uncomment each property starts with "#MessageBroker.SSL." by removing the leading pound sign '#' character.

The property MessageBroker.SSL.enabled determines whether to activate other SSL properties and enable SSL messaging. Value 'Y' means yes, and 'N' no. You can use this property switch between SSL and non SSL messaging modes.

Step 6 For each of the above SSL properties, assign value applicable to your certificate.

MessageBroker.SSL.keyStore: Path to the key store

MessageBroker.SSL.keyStorePassword: Password needed to open the key store

MessageBroker.SSL.keyPassword: Password needed to read the key, if it's different from the password of the key store

MessageBroker.SSL.trustStore: Path to the trust store

MessageBroker.SSL.trustStorePassword: Password needed to open the trust store

Note You must obfuscate the passwords before storing them in the property files. Refer to Securing Key Store Passwords for instructions.

Step 7 Save the property file.

If you are setting up Remote Master, continue on to Configuring SSL on Remote Master.

Otherwise, if you setting up a fault tolerant system, continue on to Configuring SSL on the Backup Master.

Otherwise, continue on to “Configuring SSL on the Client Manager”.

Configuring SSL on Remote Master

In this section, you will enable SSL on Remote Master with the key stores you obtained from earlier section.

To enable:

Step 1 Shut down the Remote Master.

Step 2 Copy the key store for the Remote Master to the *config* directory in the Master's installation directory.

Step 3 Create a trust store by importing the certificates of Primary Master. Follow the instructions in "Obtaining Server Keys and Certificates".

If you are setting up a fault tolerant system, import the certificates of the Backup Master and Fault Monitor into this trust store too.

When done, copy the trust store to the **config** directory in the Master's installation directory.

Step 4 Use a text editor to open the property file *config/master.props* located in the Master's installation directory.

Note It may be a good idea to back up this file before editing it to ensure there is a good copy to fall back to.

Step 5 In the editor, locate the segment of SSL properties that looks like the following.

#MessageBroker.SSL.enabled=Y

#MessageBroker.SSL.keyStore=

#MessageBroker.SSL.keyStorePassword=

#MessageBroker.SSL.keyPassword=

#MessageBroker.SSL.trustStore=

#MessageBroker.SSL.trustStorePassword=

If such segment can't be found, manually insert these lines.

Uncomment each property starts with "#MessageBroker.SSL." by removing the leading pound sign '#' character.

The property MessageBroker.SSL.enabled determines whether to activate other SSL properties and enable SSL messaging. Value 'Y' means yes, and 'N' no. You can use this property switch between SSL and non SSL messaging modes.

Step 6 For each of the above SSL properties, assign value applicable to your certificate.

MessageBroker.SSL.keyStore: Path to the key store

MessageBroker.SSL.keyStorePassword: Password needed to open the key store

MessageBroker.SSL.keyPassword: Password needed to read the key, if it's different from the password of the key store

MessageBroker.SSL.trustStore: Path to the trust store

MessageBroker.SSL.trustStorePassword: Password needed to open the trust store

Note You must obfuscate the passwords before storing them in the property files. Refer to Securing Key Store Passwords for instructions.

Step 7 Save the property file.

If you setting up a fault tolerant system, continue on to Configuring SSL on the Backup Master. Otherwise, continue on to Configuring SSL on the Client Manager.

Configuring SSL on the Backup Master

In this section, you will enable SSL on the Backup Master with the key stores you obtained from earlier section.

To ena ble:

Step 1 Shut down the Backup Master.

Step 2 Copy the key store for the Backup Master to the **config** directory in the Master's installation directory.

Step 3 Create a trust store by importing Client Manager's certificate. Follow the instructions in "Obtaining Server Keys and Certificates"e. Import the certificate of the Fault Monitor into this trust store too.

If you are setting up Remote Master, import the certificate of the Remote Master into this trust store too.

When done, copy the trust store to the *config* directory in the Master's installation directory.

Step 4 Use a text editor to open the property file *config/master.props* located in the Master's installation directory.

Note It may be a good idea to back up this file before editing it to ensure there is a good copy to fall back to.

Step 5 In the editor, locate the segment of SSL properties that looks like the following.

#MessageBroker.SSL.enabled=Y

#MessageBroker.SSL.keyStore=

#MessageBroker.SSL.keyStorePassword=

#MessageBroker.SSL.keyPassword=

#MessageBroker.SSL.trustStore=

#MessageBroker.SSL.trustStorePassword=

If such segment can't be found, manually insert these lines.

Uncomment each property starts with "#MessageBroker.SSL." by removing the leading pound sign '#' character.

The property MessageBroker.SSL.enabled determines whether to activate other SSL properties and enable SSL messaging. Value 'Y' means yes, and 'N' no. You can use this property switch between SSL and non SSL messaging modes.

Step 6 For each of the above SSL properties, assign value applicable to your certificate.

MessageBroker.SSL.keyStore: Path to the key store

MessageBroker.SSL.keyStorePassword: Password needed to open the key store

MessageBroker.SSL.keyPassword: Password needed to read the key, if it's different from the password of the key store

MessageBroker.SSL.trustStore: Path to the trust store

MessageBroker.SSL.trustStorePassword: Password needed to open the trust store

Note You must obfuscate the passwords before storing them in the property files. Refer to Securing Key Store Passwords for instructions.

Step 7 Save the property file.

Step 8 Continue on to Configuring SSL on the Fault Monitor.

Configuring SSL on the Fault Monitor

In this section, you will enable SSL on the Fault Monitor with the key stores you obtained from earlier section.

To enable:

Step 1 Shut down the Fault Monitor.

Step 2 Copy the key store for the Fault Monitor to the **config** directory in its installation directory.

Step 3 Create a trust store by importing Client Manager's certificate. Follow the instructions in Exporting and Importing Certificate. Import the certificates of the Primary Master and Backup Master into this trust store too.

When done, copy the trust store to the *config* directory in the installation directory.

Step 4 Use a text editor to open the property file *config/master.props* located in the installation directory.

Note It may be a good idea to back up this file before editing it to ensure there is a good copy to fall back to.

Step 5 In the editor, locate the segment of SSL properties that looks like the following.

#MessageBroker.SSL.enabled=Y

#MessageBroker.SSL.keyStore=

#MessageBroker.SSL.keyStorePassword=

#MessageBroker.SSL.keyPassword=

#MessageBroker.SSL.trustStore=

#MessageBroker.SSL.trustStorePassword=

If such segment cannot be found, manually insert these lines.

Uncomment each property starts with "**#MessageBroker.SSL.**" by removing the leading pound sign '#' character.

The property MessageBroker.SSL.enabled determines whether to activate other SSL properties and enable SSL messaging. Value 'Y' means yes, and 'N' no. You can use this property switch between SSL and non SSL messaging modes.

Step 6 For each of the above SSL properties, assign value applicable to your certificate.

MessageBroker.SSL.keyStore: Path to the key store

MessageBroker.SSL.keyStorePassword: Password needed to open the key store

MessageBroker.SSL.keyPassword: Password needed to read the key, if it's different from the password of the key store

MessageBroker.SSL.trustStore: Path to the trust store

MessageBroker.SSL.trustStorePassword: Password needed to open the trust store.

Note You must obfuscate the passwords before storing them in the property files. Refer to Securing Key Store Passwords for instructions.

Step 7 Save the property file.

Step 8 Continue on to Configuring SSL on the Client Manager.

Configuring SSL on the Client Manager

In this section, you will enable SSL on the Client Manager with the keystores you obtained from earlier section.

Step 1 Shut down the Client Manager.

a. Copy the key store for the Client Manager to the *config* directory in the Client Manager's installation directory.

b. Create a trust store by importing Primary Master's certificate. Follow the instructions in "Obtaining Server Keys and Certificates".

If you are setting up a fault tolerant system, import the certificates of the Backup Master and Fault Monitor into this trust store also.

When done, copy the trust store to the *config* directory in the Client Manager's installation directory.

Step 2 Use a text editor to open the property file *config/clientmgr.props* located in the Client Manager's installation directory.

Note It may be a good idea to back up this file before editing it to ensure there is a good copy to fall back to.

Step 3 In the editor, locate the segment of SSL properties that looks like the following.

#MessageBroker.SSL.enabled=Y

#MessageBroker.SSL.keyStore=

#MessageBroker.SSL.keyStorePassword=

#MessageBroker.SSL.keyPassword=

#MessageBroker.SSL.trustStore=

#MessageBroker.SSL.trustStorePassword=

If such segment can't be found, manually insert these lines.

Uncomment each property starts with "`#MessageBroker.SSL.`" by removing the leading pound sign '`#`' character.

The property MessageBroker.SSL.enabled determines whether to activate other SSL properties and enable SSL messaging. Value '`Y`' means yes, and '`N`' no. You can use this property switch between SSL and non SSL messaging modes.

Step 4 For each of the above SSL properties, assign value applicable to your certificate.

MessageBroker.SSL.keyStore: Path to the key store

MessageBroker.SSL.keyStorePassword: Password needed to open the key store

MessageBroker.SSL.keyPassword: Password needed to read the key, if it's different from the password of the key store

MessageBroker.SSL.trustStore: Path to the trust store

MessageBroker.SSL.trustStorePassword: Password needed to open the trust store

Note You must obfuscate the passwords before storing them in the property files. Refer to Securing Key Store Passwords for instructions.

Step 5 Save the property file.

Securing Key Store Passwords

For Client Manager

Perform the following steps if you are configuring SSL on Client Manager.

To configure:

Step 1 Open a command shell window and change directory to the *lib* directory under Client Manager's installation directory.

Step 2 Issue the following commands:

java -cp ClientManager.jar com.tidalsoft.framework.util.Pwd <your_password>

where **<your_password>** is the password to be obfuscated.

Step 3 Copy the entire line of command output and paste it into the value field of that password in property file.

Step 4 Repeat step 1 to 3 for each of the other passwords.

For Fault Monitor Or Any Master

Perform the following steps if you are configuring SSL on Fault Monitor or any Master.

To configure:

Step 1 Open a command shell window and change directory to the **lib** directory under Enterprise Scheduler's installation directory.

Step 2 Issue the following commands:

```
java -cp Scheduler.jar com.tidalsoft.framework.util.Pwd <your_password>
```

where **`<your_password>`** is the password to be obfuscated.

Step 3 Copy the entire line of command output and paste it into the value field of that password in property file.

Step 4 Repeat step 1 to 3 for each of the other passwords.