

# **Coexistence of Default and Custom TLS Certificates**

**until the custom certificates are added on all components  
and the default certificates deleted**

## **IBM/HCL Workload Automation (WA)**

**Sajjad M. Kabir**  
Solutions Architect  
IBM/HCL Lab Services

**Version: 2.4**  
**Date: Feb 09, 2025**

---

# 1 Document History

---

## 1.1 Document Location

This is a snapshot of an on-line document. Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

---

## 1.2 Revision History

The following is a revision history of the document.

| Revision Number | Revision Date | Summary of Changes   | Changes Marked |
|-----------------|---------------|--|----------------|
| 1.0             | May 17, 2024  | Initial version  |                |
| 1.1             | May 28, 2024  | Updated some of the steps  | No             |
| 1.2             | May 30, 2024  | Added a new step 6 in section 4.5 Populate the Certificate Depot from the new Certificate              | No             |
| 1.3             | Jun 5, 2024   | Reordered and added steps 7 & 8 in section 4.4 Populate the Certificate Depot from the new Certificate | No             |
| 1.4             | Jun 11, 2024  | Added sections 4.8 and 4.9   | No             |
| 1.5             | Jun 12, 2024  | Updated section 4.6.6  | No             |
| 1.6             | Jun 24, 2024  | Updated various sections   | No             |
| 1.7             | Jun 26, 2024  | Added section 5 Other Resources Regarding Replacing Certificates                                       | No             |
| 1.8             | Jul 1, 2024   | Included product version in section 2 Introduction   | No             |
| 1.9             | Jul 8, 2024   | Added section 4.9 Certificates in each Store Before and After Updating all Components                  | No             |
| 2.0             | Jul 9, 2024   | Added steps b and c in section 4.8.1   | No             |
| 2.1             | Aug 2, 2024   | Updated step7 in section 4.4 Populate the Certificate Depot from the new Certificate                   | No             |
| 2.2             | Aug 21, 2024  | Updated section 4.6 and added section 4.9  | No             |
| 2.3             | Aug 27, 2024  | Updated sections 4.4, 4.6, and 4.10  | No             |
| 2.4             | Feb 9, 2025   | Moved section 4.4 to 4.2 and updated step 6 in section 4.3   | No             |

---

## 1.3 Approvals

This document requires following approvals.

| Name | Title |
|------|-------|
|      |       |
|      |       |

---

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>DOCUMENT HISTORY.....</b>  | <b>2</b>  |
| 1.1      | DOCUMENT LOCATION .....   | 2         |
| 1.2      | REVISION HISTORY.....   | 2         |
| 1.3      | APPROVALS .....   | 2         |
| <b>2</b> | <b>INTRODUCTION.....</b>  | <b>4</b>  |
| <b>3</b> | <b>OVERVIEW OF TLS IN IWA .....</b>                                       | <b>5</b>  |
| <b>4</b> | <b>REPLACING THE DEFAULT CERTIFICATES WITH CUSTOM CERTIFICATES .....</b>  | <b>7</b>  |
| 4.1      | KEYSTORE AND TRUSTSTORE USED BY MDM AND DWC.....                          | 7         |
| 4.2      | POPULATE THE CERTIFICATE DEPOT FROM THE NEW CERTIFICATE .....             | 7         |
| 4.3      | IMPORTING NEW CERTIFICATES FOR MDM AND DWC .....                          | 9         |
| 4.4      | UPDATE DWB WORKSTATION PROPERTIES .....                                   | 12        |
| 4.5      | IMPORTING NEW CERTIFICATES FOR BKMs, DDMs, BDDMs .....                    | 12        |
| 4.6      | IMPORTING NEW CERTIFICATES FOR ALL DAs EXCEPT THE ONE ON THE MDM.....     | 13        |
| 4.7      | PLACE THE NEW CERTIFICATES ON DMS AND FTAs.....                           | 16        |
| 4.8      | IMPORTING NEW CERTIFICATES FOR THE DA AND FTA ON THE MDM.....             | 17        |
| 4.9      | DELETE THE DEFAULT CERTIFICATES .....                                     | 18        |
| 4.10     | CERTIFICATES IN EACH STORE BEFORE AND AFTER UPDATING ALL COMPONENTS ..... | 21        |
| 4.11     | RENEWING THE EXPIRED CERTIFICATE .....                                    | 22        |
| <b>5</b> | <b>OTHER RESOURCES REGARDING REPLACING CERTIFICATES .....</b>             | <b>23</b> |
| <b>6</b> | <b>AUTHOR'S BIO.....</b>  | <b>24</b> |

---

## 2 Introduction

Network communication between each component of Workload Automation (WA) is secured using TLS v1.2 and v1.3 protocols. This is accomplished by deploying TLS Certificates for each component. WA deploys Self-Signed certificates at installation time to make it easy to deploy and have a working scheduling environment with little effort. To enhance security, organizations decide to replace the default certificates with commercially recognized CA (Certificate Authority) signed certificate for each server in the environment. This requires all default self-signed certificates used by every component to be replaced by CA signed certificates.

In order to keep the environments intact and all components communicating while the default certificates are replaced, instead of a rip and replace approach, a cap and grow approach is adopted where the default certificates coexist with the new certificates until the new certificates have been imported into the KeyStores and TrustStores of all components. The default certificates can then be deleted from all KeyStores and TrustStores.

The following sections describe the procedure to replace the certificates. This procedure was developed based on v10.1 Fix Pack 04.

---

## 3 Overview of TLS in IWA

IWA utilizes IBM WebSphere Application Server Liberty Profile (WLP) as the frontend web user interface as well as the backend engine. As such, there are two WLP instances that provide the services for DWC and MDM. The TLS implementation in WLP for DWC and MDM requires two stores to be present. The following is a description of the terminologies and concept used.

|                               |   |
|-------------------------------|---|
| <b>KeyStore (KeyFile)</b>     | Is a file to store private keys in the form of Personal Certificates. Some KeyStores also contain trusted or public keys. |
| <b>TrustStore (TrustFile)</b> | Is a file to store public keys in the form of Signer Certificates. Some TrustStores also contain trusted or private keys. |
| <b>Personal Certificate</b>   | Represents the identity of a server and contains a private key for signing/encrypting data.                               |
| <b>Signer Certificate</b>     | Represents the identity of a server and contains a public key for decrypting data.  |
| <b>Certificate Chain</b>      | Is a concatenation of CA signed server certificate and certificates of Intermediate and Root CAs.                         |
| <b>Trusted CAs</b>            | Is a concatenation of trusted CA certificates.  |

TLS Certificates are imported into the KeyStores for each profile and then the public keys are extracted and imported into the TrustStores.

The stores can have different formats for compatibility and security reasons and are used by different components as shown below.

|            |  |
|------------|--|
| <b>JKS</b> | Java Key Store<br>Used by the MDM and DWC WLP as well as a component of the Dynamic Agent<br>File extension is jks |
| <b>CMS</b> | Cryptographic Message Syntax<br>Used by the Dynamic Agents<br>File extension is kdb                                |

There are different types of file extensions used to signify the type of certificate or key it contains as described below.

|              |  |
|--------------|--|
| <b>CSR</b>   | A Personal Certificate Signing Request contains the request to be sent to the CA for signing. It is generated from a KeyStore.                               |
| <b>CRT</b>   | A Personal Certificate that is signed by the CA. It is imported into the KeyStore  |
| <b>CER</b>   | A Personal Certificate that is signed by the CA or a public key for root CA, Intermediate CA, server exported from a KeyStore and imported into a TrustStore |
| <b>KEY</b>   | A Private Key  |
| <b>Alias</b> | A label for a signed certificate after being imported into a KeyStore or TrustStore  |
| <b>ARM</b>   | A Signer Certificate chain. It is exported from a KeyStore and imported in to a TrustStore.  |
| <b>DER</b>   | Distinguished Encoding Rules to encode any data object into a binary file. A certificate is encoded using these rules and presented in a binary file.        |

---

**PKCS12** A Public-Key Cryptography Standards (PKCS#12) keystore that contains the private key(s), public certificate(s), and their CA certificate chain(s). The file extension can be either p12 or pfx (Personal Information Exchange).

**PEM** Privacy-Enhanced Electronic Mail is an encryption format used by certificates. It is a base 64 encoding of the Certificate and is enclosed between

"-----BEGIN CERTIFICATE-----"

and

"-----END CERTIFICATE-----"

---

## 4 Replacing the Default Certificates with Custom Certificates

Follow the steps below to replace the default certificates with the custom certificates.

---

### 4.1 KeyStore and TrustStore Used by MDM and DWC

The MDM and DWC prior to version 10.2.3 (denoted as <10.2.3) each has one KeyStore and one TrustStore in JKS format. In version 10.2.3 and onward ((denoted as >=10.2.3), the format of the KeyStore and TrustStore has changed to PKCS12 (p12 extension).

The sample commands in this document uses the jks extension for all KeyStore and TrustStores. Based on the version of the product, make the substitutions accordingly.

The KeyStores and TrustStores are located in the following directories:

|                    |            |   |
|--------------------|------------|---|
| <b>MDM</b>         |            | <install_dir>/usr/servers/engineServer/resources/security |
| <b>&lt;10.2.3</b>  | KeyStore   | <b>TWSServerKeyFile.jks</b>                               |
|                    | TrustStore | <b>TWSServerTrustFile.jks</b>                             |
| <b>&gt;=10.2.3</b> | KeyStore   | <b>TWSServerKeyFile.p12</b>                               |
|                    | TrustStore | <b>TWSServerTrustFile.p12</b>                             |
| <b>DWC</b>         |            | <<install_dir>/usr/servers/dwcServer/resources/security   |
| <b>&lt; 10.2.3</b> | KeyStore   | <b>TWSServerKeyFile.jks</b>                               |
|                    | TrustStore | <b>TWSServerTrustFile.jks</b>                             |
| <b>&gt;=10.2.3</b> | KeyStore   | <b>TWSServerKeyFile.p12</b>                               |
|                    | TrustStore | <b>TWSServerTrustFile.p12</b>                             |

#### **TWSServerKeyFile**

This KeyStore contains the private and public certificates for the server, the Root, and Intermediate public certificates of the CA that signed the server certificate. The public certificate of the server is exported and imported into the TWSServerTrustStore.jks and TWSClientKeyStore.

#### **TWSServerTrustFile**

This TrustStore contains the public certificates for the server, the Root and Intermediate public certificates of the CA that signed the server certificate.

---

### 4.2 Populate the Certificate Depot from the new Certificate

The utility, AgentCertificateDownloader, downloads the new certificates from the **MDM** to the **DA**, but the certificates need to be in PEM format. Follow the steps below to create the certificates in PEM format.

1. Create a request with the CA for a certificate for the DWCs and MDMs and include the DNS names for all other MDMs, BKMs, DDMs, and BDDMs in different environments in the Subject Alternate Field (SAN) field. This simplifies the process of managing one certificate for all components.

- 
- The signed certificate, private key, and the CA Chain are usually provided in a PFX (Personal Information Exchange) file, for example, **iwa.certs.pfx**. This file is essentially a KeyStore in PKCS12 format. Copy the PKCS12 KeyStore to a directory on the MDM, for example:

```
/home/iwadmin/ssl
```

- Export the **private key** of the new certificate from the PKCS12 KeyStore

```
openssl pkcs12 -in iwa.certs.pfx -nodes -nocerts -out tls.key  
Enter Import Password: <password_provided_by_CA>
```

- Export the **public certificate** (clcert – client cert) of the new certificate from the PKCS12 KeyStore

```
openssl pkcs12 -in iwa.certs.pfx -clcerts -nokeys -out tls.crt  
Enter Import Password: <password_provided_by_CA>
```

- Export the **CA** certificate chain from the PKCS12 KeyStore

```
openssl pkcs12 -in iwa.certs.pfx -nokeys -cacerts -chain -out ca.crt  
Enter Import Password: <password_provided_by_CA>
```

- For coexistence of both the default and new certificates, concatenate the contents of the new certificate chain, **ca.crt**, and the **trusted certificates** used by the **FTA on the MDM**, **TWSTrustCertificates.cer** (<10.2.3) or **TWSCientTrustStore.pem** (>=10.2.3), to a temporary file. Copy this file and overwrite the existing file. This allows the MDM to trust the new certificates to be deployed to the FTAs along with the default certificates used by the FTAs, which haven't been updated yet.

Notice the order of appearance of the files in the command below. This results in the new certificates at the top of the file, which makes it easier to extract them when deleting the default certificates in a later section.

### <10.2.3

```
cat ca.crt <Install_Dir>/TWS/ssl/OpenSSL/TWSTrustCertificates.cer > /tmp/ca.new  
mv /tmp/ca.new <Install_DIR>/TWS/ssl/OpenSSL/TWSTrustCertificates.cer
```

### >=10.2.3

```
cat ca.crt <Install_Dir>/TWS/ssl/certs/TWSCientTrustStore.pem > /tmp/ca.new  
mv /tmp/ca.new <Install_DIR>/TWS/ssl/certs/TWSCientTrustStore.pem
```

- Open the resulting file, TWSTrustCertificates.cer or TWSCientTrustStore.pem, and make sure that each certificate begins with, **BEGIN CERTIFICATE**, and ends with, **END CERTIFICATE**, on a line by itself. Any text in between two certificates is ignored.

- Create the stash file with the password

```
echo -n <password> | base64 > tls.sth
```

- Copy the following certificates to the Certificate Depot. Notice that **ca.crt** is not copied, which is copied in a following step.

```
cp tls.key tls.crt tls.sth <Install_Dir>/TWSDATA/ssl/depot
```

- Since the server certificate is signed by a chain of certificates - an intermediate certificate and a Root CA certificate, each certificate must be extracted into its own file. The file name is used to create the alias name when AgentCertificateDownloader imports them, in alphabetical order, into the keystores. The Root CA certificate must be extracted into **ca.crt** and the intermediate certificate must be extracted into the folder **additionalCAs**. For coexistence of both the default and new certificates, the default certificates must also be extracted into their own files in the same directory, additionalCAs. The certificates found in



---

this directory and **ca.crt** in the depot directory are imported into the two KeyStores, JKS and CMS, used by the DAs and concatenated into ca.crt, used by the FTA.

Create the folder, **additionalCAs**, in the depot directory.

```
cd <install_dir>/TWSDATA/ssl/depot
mkdir additionalCAs
```

11. Copy each certificate from <Install\_Dir>/TWS/ssl/OpenSSL/**TWSTrustCertificates.cer** or <Install\_DIR>/TWS/ssl/certs/**TWSClientTrustStore.pem**, starting at the top, and paste it into a separate file in additionalCAs dir with the following names. The order of appearance of the certificates matches the names of the certificates listed below.

```
cd additionalCAs
ls -l
Intermediate.crt
ca.crt
Server.crt
Client.crt
ClientNew.crt
ServerNew.crt
```

14. Move ca.crt one level above, to the depot directory

```
mv ca.crt ../
```

15. The following is the final contents of the depot directory:

```
<Install_Dir>/TWSDATA/ssl/depot> ls -lR
.:
total 36
drwxrwxr-x 2 iwadmin iwadmin 152 Aug 20 21:33 additionalCAs
-rw----- 1 iwadmin iwadmin 2428 Aug 20 18:19 ca.crt
-rw----- 1 iwadmin iwadmin 2832 Aug 20 17:09 tls.crt
-rw----- 1 iwadmin iwadmin 1857 Aug 20 17:09 tls.key
-rw-rw-r-- 1 iwadmin iwadmin 13 Aug 20 17:09 tls.sth

./additionalCAs:
total 28
-rw----- 1 iwadmin iwadmin 2666 Aug 20 17:09 Intermediate.crt
-rw----- 1 iwadmin iwadmin 725 Aug 20 17:09 Client.crt
-rw----- 1 iwadmin iwadmin 733 Aug 20 17:09 ClientNew.crt
-rw----- 1 iwadmin iwadmin 725 Aug 20 17:09 Server.crt
-rw----- 1 iwadmin iwadmin 733 Aug 20 17:09 ServerNew.crt
```

---

### 4.3 Importing new Certificates for MDM and DWC

The following is a procedure to import the new signed certificates on the MDM. The steps are the same for the DWC.

1. Backup the KeyStore and TrustStore

**MDM:** cd <install\_dir>/usr/servers/engineServer/resources/security

**DWC:** cd <install\_dir>/usr/servers/dwcServer/resources/security

**<10.2.3** cp TWSServerKeyFile.jks TWSServerKeyFile.jks.orig  
cp TWSServerTrustFile.jks TWSServerTrustFile.jks.orig

---

```
>=10.2.3 cp TWSServerKeyFile.p12 TWSServerKeyFile.p12.orig
cp TWSServerTrustFile.p12 TWSServerTrustFile.p12.orig
```

2. Import the contents of the PFX file provided by the CA into the MDM KeyStore as shown below:

```
cp /home/iwadmin/ssl/ <Install_dir>/usr/servers/<component>/resources/security
export PATH=<Install_dir>/TWS/JavaExt/jre/jre/bin:$PATH
```

**<10.2.3**

```
keytool -importkeystore -srckeystore iwa.certs.pfx -srcstoretype PKCS12 -srcstorepass
<password_provided_by_CA> -destkeypass <password> -destkeystore
TWSServerKeyFile.jks -deststorepass <password>
```

**>=10.2.3**

```
keytool -importkeystore -srckeystore iwa.certs.pfx -srcstoretype PKCS12 -srcstorepass
<password_provided_by_CA> -destkeypass <password> -destkeystore
TWSServerKeyFile.p12 -deststorepass <password>
```

3. View the contents of the KeyStore

**<10.2.3**

```
keytool -list -keystore TWSServerKeyFile.jks -storepass <password>
```

```
Keystore type: jks
Keystore provider: SUN
```

```
Your keystore contains 2 entries
```

```
server (default certificate)
glowfish (new certificate)
```

**>=10.2.3**

```
keytool -list -keystore TWSServerKeyFile.p12 -storepass <password>
```

```
Keystore type: PKCS12
Keystore provider: SUN
```

```
Your keystore contains 2 entries
```

```
server (default certificate)
glowfish (new certificate)
```

4. Export the **public** certificate of the **new certificate** into a file from the **KeyStore**

**<10.2.3**

```
keytool -exportcert -alias glowfish -file glowfish.pub.crt -keystore
TWSServerKeyFile.jks -storetype jks -storepass <password>
```

**>=10.2.3**

```
keytool -exportcert -alias glowfish -file glowfish.pub.crt -keystore
TWSServerKeyFile.p12 -storetype pkcs12 -storepass <password>
```

5. Import the public certificates of the new certificate exported in the above steps into the TrustStore

**<10.2.3**

```
keytool -importcert -alias glowfish -file glowfish.pub.crt -keystore
TWSServerTrustFile.jks -storepass <password> -trustcacerts -noprompt
```

**>=10.2.3**

---

```
keytool -importcert -alias glowfish -file glowfish.pub.crt -keystore
TWSServerTrustFile.p12 -storepass <password> -trustcacerts -noprompt
```

6. Import the **CA Chain** from the individual files extracted earlier in the depot directory

**<10.2.3**

```
keytool -importcert -alias Intermediate -file
<Install_Dir>/TWSDATA/ssl/depot/additionalCAs/Intermediate.crt -keystore
TWSServerTrustFile.jks -storepass <password> -trustcacerts -noprompt
```

```
keytool -importcert -alias RootCA -file <Install_Dir>/TWSDATA/ssl/depot/ca.crt -
keystore TWSServerTrustFile.jks -storepass <password> -trustcacerts -noprompt
```

**>=10.2.3**

```
keytool -importcert -alias Intermediate -file
<Install_Dir>/TWSDATA/ssl/depot/additionalCAs/Intermediate.crt -keystore
TWSServerTrustFile.p12 -storepass <password> -trustcacerts -noprompt
```

```
keytool -importcert -alias RootCA -file <Install_Dir>/TWSDATA/ssl/depot/ca.crt -
keystore TWSServerTrustFile.p12 -storepass <password> -trustcacerts -noprompt
```

7. Export the certs included in the TrustStore to a file

**<10.2.3**

```
keytool -list -keystore TWSServerTrustFile.jks -storepass <password> -v > cert.list
```

**>=10.2.3**

```
keytool -list -keystore TWSServerTrustFile.p12 -storepass <password> -v > cert.list
```

8. Find the following certificates in the file, **cert.list**

vi **cert.list**

Search for the following text:

|                                 |                                   |
|---------------------------------|-----------------------------------|
| Alias name: <b>glowfish</b>     | Match: CN= <b>glowfish</b>        |
| Alias name: <b>intermediate</b> | Match: CN= <b>Intermediate...</b> |
| Alias name: <b>rootca</b>       | Match: CN= <b>Root CA...</b>      |
| Alias name: client              | Match: CN=ClientNew               |
| Alias name: clientold           | Match: CN=Client                  |
| Alias name: twstrustkey         | Match: CN=ServerNew               |
| Alias name: twstrustkeyold      | Match: CN=Server                  |

9. A trusting relationship already exists between the DWC and MDM since their certificates are signed by the same CA. Therefore, it is not necessary to import the CA Certificate chain of one to the other component. If that is not the case, import the CA certificate chain from one to the other.

10. WebSphere Liberty Profile uses the KeyStore, TWSServerKeyFile.jks, and now that it contains two certificates, which certificate it should use must be indicated by adding the following attribute in **ssl\_config.xml** file in the **defaults** directory as shown below on the MDM:

```
cd <TWSDATA>/usr/servers/engineServer/configDropins/defaults
cp ssl_config.xml ssl_config.xml.orig
vi ssl_config.xml
```

Add the following line in bold:

```
<ssl id="twaSSLSettings" keyStoreRef="twaKeyStore" trustStoreRef="twaTrustStore"
sslProtocol="TLSv1.2" clientAuthenticationSupported="true"
serverKeyAlias="server"/>
```

- 
11. The **DWC** doesn't need to keep both certificates in the KeyStore, hence, delete the old certificate

```
cd <install_dir>/usr/servers/dwcServer/resources/security
```

**<10.2.3**

```
keytool -delete -alias server -keystore TWSServerKeyFile.jks
```

**>=10.2.3**

```
keytool -delete -alias server -keystore TWSServerKeyFile.p12
```

12. Restart the **DWC** WLP

```
cd <Install_Dir>/appservertools
```

```
./stopAppServer.sh
```

```
./startAppServer.sh
```

13. Go to the DWC URL and verify that it is using the new certificate by clicking on the pad lock on the URL address bar and viewing the certificate.

---

## 4.4 Update DWB Workstation Properties

To work with the Dynamic Agents (DA) through the Dynamic Workload Broker (DWB), the CN of the CA signed certificate needs to be added to the Broker.**AuthorizedCNs** property in **BrokerWorkstation.properties** file.

Follow the steps below to update this file.

1. Go to the directory where the **BrokerWorkstation.properties** file is located

```
cd <Install_Dir>/TWSDATA/broker/config
```

2. Backup and edit the **BrokerWorkstation.properties** file and append the values shown in bold below

```
cp BrokerWorkstation.properties BrokerWorkstation.properties.orig
```

```
vi BrokerWorkstation.properties
```

```
Broker.AuthorizedCNs=Server;ServerNew;glowfish
```

### Note:

The CN=Server has the Alias twstrustkeyold in TWSServerTrustFile.jks

The CN=ServerNew has the Alias twstrustkey in TWSServerTrustFile.jks

The CN=glowfish has the alias glowfish in TWSServerTrustFile.jks

3. Restart the **MDM** WLP

```
cd <Install_Dir>/appservertools
```

```
./stopAppServer.sh
```

```
./startAppServer.sh
```

---

## 4.5 Importing new Certificates for BKMs, DDMs, BDDMs

These components have the same structure as the MDM. As such, copy the **KeyStore**, **TrustStore** from the **MDM** to these components, update the **BrokerWorkstation.properties** and **ssl\_config.xml** files, and restart WLP.

---

## 4.6 Importing new Certificates for All DAs except the one on the MDM

The certificates for the **default DA and FTA on the MDM** must be imported **after** certificates on all components are imported. If the certificates for the default DA and FTA on the MDM are updated before all other components are updated, FTAs don't link to the MDM anymore. This is because once the new certificate is imported for the FTA on the MDM, other FTAs don't trust that certificate as they don't have the CA chain of the certificate that signed that certificate in their trusted list of certificates, **TWSTrustCertificates.cer**, yet, which is updated when the procedure is executed on those FTAs.

The DA prior to version 10.2.3 (denoted as <10.2.3) has two KeyStores one in JKS and the other in CMS format (kdb extension). Both KeyStores contain the same certificates, which include the private certificate for the server where the agent is installed, the public certificate of the MDM server and the CA Chain, and all third party trusted CA certificates. In addition, the individual certs in pem format used by the FTAs are located in OpenSSL folder as shown below.

In version 10.2.3 and onward (denoted as >=10.2.3), the two KeyStores have been consolidated into one KeyStore and one TrustStore both in PKCS12 format (p12 extension). A copy of the TrustStore is also provided in PEM format (pem extension). FTAs no longer use the PEM certificates in <Install\_Dir>/TWSADATA/ssl/OpenSSL dir. Instead, they use the same KeyStore as the DA and the TrustStore in PEM format.

The sample commands in this document use the jks and kdb extensions for all KeyStores. Based on the version of the product, make the substitutions accordingly.

The KeyStores and TrustStores are located in the following directory:

### <10.2.3

JKS KeyStore: <Install\_Dir>/TWSADATA/ssl/**TWSCientKeyStoreJKS.jks**  
CMS KeyStore: <Install\_Dir>/TWSADATA/ssl/GSKit/**TWSCientKeyStore.kdb**  
Certs in PEM format: <Install\_Dir>/TWSADATA/ssl/**OpenSSL**

### >=10.2.3

KeyStore: <Install\_Dir>/TWSADATA/ssl/certs/**TWSCientKeyStoreJKS.p12**  
TrustStore: <Install\_Dir>/TWSADATA/ssl/certs/**TWSCientTrustStore.p12**  
<Install\_Dir>/TWSADATA/ssl/certs/**TWSCientTrustStore.pem**

To make things easier and economical, the new certificates used on the MDM can be used for the DAs without compromising security. A new utility, AgentCertificateDownloader, can download the certificates from the Certificate Depot on the MDM and build the KeyStores. Since it only downloads the new certificates and leaves out the default certificates, it breaks the trusting relationship with the MDM and causes the DAs to not link. As such, a few additional steps need to be executed.

Follow the steps below to download the new certificates and import the default certificates.

1. Backup the following KeyStores and files

- a. cd <Install\_Dir>/TWSADATA  
cp **localopts** localopts.orig

### <10.2.3

- b. cd <Install\_Dir>/TWSADATA/ssl/  
cp **TWSCientKeyStoreJKS.jks** TWSCientKeyStoreJKS.jks.orig
- c. cd <Install\_Dir>/TWSADATA/ssl/GSKit  
cp **TWSCientKeyStore.kdb** TWSCientKeyStore.kdb.orig

---

**>=10.2.3**

- b. cd <Install\_Dir>/TWSDATA/ssl/certs  
cp **TWClientKeyStoreJKS.p12** TWClientKeyStoreJKS.p12.orig
- c. cp **TWClientTrustStore.p12** TWClientTrustStore.p12.orig
- d. cp **TWClientTrustStore.pem** TWClientTrustStore.pem.orig

2. The following certs are included in the CMS KeyStores before running AgentCertificateDownloader

**<10.2.3**

```
export PATH=/usr/Tivoli/TWS/GSKit64/8/bin/:$PATH
cd <Install_Dir>/TWSDATA/ssl/GSKit
gsk8capicmd_64 -cert -list -db TWClientKeyStore.kdb -v -stashed
```

Certificates found

\* default, - personal, ! trusted, # secret key

! **server** CN=ServerNew,OU=TWS,O=IBM,C=US CN=ServerNew,OU=TWS,O=IBM,C=US

! **serverold** CN=Server,OU=TWS,O=IBM,C=US CN=Server,OU=TWS,O=IBM,C=US

- **client** CN=ClientNew,OU=TWS,O=IBM,C=US CN=ClientNew,OU=TWS,O=IBM,C=US

3. The following certs are included in the JKS and PKCS12 KeyStores and TrustStore before running AgentCertificateDownloader

```
cd <Install_Dir>/TWSDATA/ssl
export PATH=<Install_Dir>/TWS/JavaExt/jre/jre/bin:$PATH
```

**<10.2.3**

```
keytool -list -keystore TWClientKeyStoreJKS.jks -storepass <password> -v > cert.list
```

Find the following 2 certificates in the file, cert.list

vi **cert.list**

Search for the following text:

Alias Name: **server** Match: CN=ServerNew

Alias Name: **client** Match: CN=ClientNew

**>=10.2.3**

```
cd <Install_Dir>/TWSDATA/ssl/certs
keytool -list -keystore TWClientKeyStoreJKS.p12 -storepass <password> -v > kcert.list
keytool -list -keystore TWClientTrustStoreJKS.p12 -storepass <password> -v > tcert.list
```

4. In versions prior to 10.2.3, the utility, AgentCertificateDownloader, needs to be launched in a shell where the IWS environmental variables haven't been set. This is due to a conflict with the curl libraries sourced with the OS provided curl binary. Follow the steps below to setup such a shell env and run the utility.

In version 10.2.3 and onward, the utility, AgentCertificateDownloader, can be launched in the current shell.

- a. vi .profile and comment the line that sources the env setting script, tws\_env.sh
- b. Logout and Log back in
- c. cd <Install\_Dir>/TWS
- d. **./AgentCertificateDownloader.sh --apikey <Personal API key from DWC> --tdwbhostname <MDM> --tdwbport 31116 --work\_dir /tmp/acd**

5. Uncomment the line in .profile to source the env setting script, tws\_env.sh

6. Source the environment  
. ./profile

- 
7. The following certs are now included in the CMS KeyStores after running AgentCertificateDownloader and importing the default certificate

**<10.2.3**

```
gsk8capicmd_64 -cert -list -db TWSCientKeyStore.kdb -v -stashed  
Certificates found
```

```
! ca CN=Root CA...  
! Intermediate CN=Intermediate...  
! Client CN=Client,OU=TWS,O=IBM,C=US CN=Client,OU=TWS,O=IBM,C=US  
! ClientNew CN=ClientNew,OU=TWS,O=IBM,C=US CN=ClientNew,OU=TWS,O=IBM,C=US  
! Server CN=Server,OU=TWS,O=IBM,C=US CN=Server,OU=TWS,O=IBM,C=US  
! ServerNew CN=ServerNew,OU=TWS,O=IBM,C=US  
- client CN=glowfish...
```

8. The following certs are now included in the JKS and PKCS12 KeyStores and TrustStore after running AgentCertificateDownloader and importing the default certificate

**<10.2.3**

```
cd <Install_Dir>/TWSDATA/ssl  
keytool -list -keystore TWSCientKeyStoreJKS.jks -storepass <password> -v >  
cert.list.new
```

Find the following 2 certificates in the file, cert.list.new

```
vi cert.list.new
```

search for the following text:

```
Alias name: ca Match: CN=Root CA...  
Alias name: client Match: CN=Client,OU=TWS,O=IBM,C=US  
Alias name: clientnew Match: CN=ClientNew,OU=TWS,O=IBM,C=US  
Alias name: Intermediate Match: CN=Intermediate...  
Alias name: server Match: CN=glowfish  
Alias name: servernew Match: CN=ServerNew
```

**>=10.2.3**

```
cd <Install_Dir>/TWSDATA/ssl/certs  
keytool -list -keystore TWSCientKeyStoreJKS.p12 -storepass <password> -v > kcert.list  
keytool -list -keystore TWSCientTrustStoreJKS.p12 -storepass <password> -v > tcert.list
```

9. Go to the **OpenSSL** dir and verify that the following files are downloaded.

```
cd <Install_Dir>/TWSDATA/ssl/OpenSSL
```

```
ls -l
```

**additionalCAs**

**ca.crt** (should contain 6 certs)

**tls.crt**

**tls.key**

**tls.rnd**

**tls.sth**

```
cd additionalCAs
```

```
ls -l
```

**Intermediate.crt**

**Client.crt**

**ClientNew.crt**

**Server.crt**

**ServerNew.crt**

10. Restart the DA

---

```
<Install_Dir>/TWS/ShutDownLwa
<Install_Dir>/TWS/StartUpLwa
```

11. Restart the FTA, if one exists

```
conman "stop;wait"
conman "shut;wait"
<Install_Dir>/TWS/StartUp
conman start
```

12. Verify that the DA and FTA are linked

13. Verify that composer and optman commands work (on BKMs, DDMs, and BDDMs)

---

## 4.7 Place the New Certificates on DMs and FTAs

An FTA when enabled to use SSL uses the following PEM files referenced in **localopts** file.

```
<10.2.3
<Install_Dir>/TWSDATA/ssl/OpenSSL
tls.key
tls.crt
tls.sth
ca.crt
```

```
>=10.2.3
<Install_Dir>/TWSDATA/ssl/certs
tls.sth
tls.rnd
TWSClietKeyStore.p12
TWSClietTrustStore.pem
```

Follow the steps below to replace the certificates on the DM and FTA.

1. Copy the above certificates from the directory on the MDM as shown below to the FTA. If the DA on the FTA was already updated with running the utility, AgentCertificateDownloader, these certificates would have been already copied.

```
<10.2.3
<Install_Dir>/TWSDATA/ssl/depot (including additionalCAs)
```

```
>=10.2.3
<Install_Dir>/TWSDATA/ssl/certs
```

2. Place the above certificates in the dir shown below on the FTA

```
<10.2.3
<FTA_Install_Dir>/TWSDATA/ssl/OpenSSL
```

```
>=10.2.3
<FTA_Install_Dir>/TWSDATA/ssl/certs
```

3. Go to the following directory and edit localopts

```
cd <Install_Dir>/TWSDATA
cp localopts localopts.orig
vi localopts
```

4. Make the following changes in **bold**

```
<10.2.3
nm SSL full port      =31113
```



---

```
SSL key = "<Install_Dir>/TWSDATA/ssl/OpenSSL/tls.key"
SSL certificate = "<Install_Dir>/TWSDATA/ssl/OpenSSL/tls.crt"
SSL key pwd = "<Install_Dir>/TWSDATA/ssl/OpenSSL/tls.sth"
SSL CA certificate = "<Install_Dir>/TWSDATA/ssl/OpenSSL/ca.crt"
SSL random seed = "<Install_Dir>/TWSDATA/ssl/OpenSSL/tls.rnd"
```

5. Update the Workstation definition of the FTA with the following settings in **bold**. For example, the definition of the FTA, GLOWFISH, is shownn below.

### **<10.2.3**

```
composer cr glowfish.txt from ws=@!glowfish
```

```
CPUNAME glowfish
OS UNIX
NODE glowfish.raleigh.ibm.com TCPADDR 31111
SECUREADDR 31113
TIMEZONE America/New_York
DOMAIN MASTERDM
FOR MAESTRO
TYPE FTA
AUTOLINK ON
BEHINDFIREWALL OFF
SECURITYLEVEL FORCE_ENABLED
FULLSTATUS OFF
END
```

6. Import the modified FTA definition

```
composer replace glowfish.txt
```

7. Update the plan so that the updated FTA definition is reflected in the plan

```
optman chg cf = all
JnextPlan -for 0000
optman chg cf = yes
```

8. Stop the FTA

```
conman "stop;wait"
conman "shut;wait"
```

9. Start the FTA

```
cd <Install_Dir>/TWS
./StartUp
conman start
```

10. Verify that the FTA linked with the server successfully and it shows all the flags, **LTI JW**

---

## **4.8 Importing new Certificates for the DA and FTA on the MDM**

Verify that certificates on all DAs and FTAs are imported. Repeat the same steps as in section [4.6 Importing new Certificates for All DAs except the one on the MDM](#) for the **DA and FTA on the MDM**.

---

## 4.9 Delete the Default Certificates

Once the certificates on all components have been updated, the default certificates can be deleted from various KeyStore and TrustStores. Follow the steps below to delete the default certificates.

1. Update the MDMs and DDMs to start using the new certificate by switching the alias in the following files from **server** to the alias of the new certificate, **glowfish**
  - a. `cd <install_dir>/usr/servers/engineServer/configDropins/defaults`  
`cp ssl_config.xml ssl_config.xml.orig2`  
`vi ssl_config.xml`  
Change the value of following attribute in bold:  
  
From: `<ssl id="twaSSLSettings" keyStoreRef="twaKeyStore" trustStoreRef="twaTrustStore" sslProtocol="TLSv1.2" clientAuthenticationSupported="true" serverKeyAlias="server"/>`  
  
To: `<ssl id="twaSSLSettings" keyStoreRef="twaKeyStore" trustStoreRef="twaTrustStore" sslProtocol="TLSv1.2" clientAuthenticationSupported="true" serverKeyAlias="glowfish"/>`
  - b. `cd <install_dir>/usr/servers/engineServer/configDropins/overrides`  
`cp jwt_variables.xml jwt_variables.xml.orig`  
`vi jwt_variables.xml`  
Change the value of following variable in bold:  
  
From: `<variable name="mp.jwt.trust.key" value="twstrustkey" />`  
To: `<variable name="mp.jwt.trust.key" value="glowfish" />`
  - c. `cd <install_dir>/usr/servers/engineServer`  
`cp server.xml server.xml.orig`  
`vi server.xml`  
Find three occurrences of the following attribute and change the value as shown in bold:  
  
From: `keyAlias="server"`  
To: `keyAlias="${mp.jwt.trust.key}"`

2. Delete the default certificates from the KeyStores and TrustStores of the MDM, BKM, DDM, BDDM

- a. Delete the old CNs from BrokerWorkstation.properties file  
`Broker.AuthorizedCNs=Server;ServerNew;glowfish`

- b. `cd <install_dir>/usr/servers/engineServer/resources/security`

### <10.2.3

- c. `keytool -delete -alias server -keystore TWSServerKeyFile.jks -storepass <password>`
- d. `keytool -delete -alias client -keystore TWSServerTrustFile.jks -storepass <password>`
- e. `keytool -delete -alias clientold -keystore TWSServerTrustFile.jks -storepass <password>`
- f. `keytool -delete -alias twstrustkey -keystore TWSServerTrustFile.jks -storepass <password>`
- g. `keytool -delete -alias twstrustkeyold -keystore TWSServerTrustFile.jks -storepass <password>`

3. Delete the default certificates from the two KeyStores of the DA.

### <10.2.3

- a. `cd <Install_Dir>/TWSDATA/ssl`
- b. `cp TWSSClientKeyStore.jks TWSServerKeyFile.jks.both`

- 
- c. `keytool -delete -alias client -keystore TWClientKeyStoreJKS.jks -storepass <password>`
  - d. `keytool -delete -alias clientnew -keystore TWClientKeyStoreJKS.jks -storepass <password>`
  - e. `keytool -delete -alias servernew -keystore TWClientKeyStoreJKS.jks -storepass <password>`
  - f. `cd <Install_Dir>/TWSDATA/ssl/GSKit`
  - g. `cp TWClientKeyStore.kdb TWClientKeyStore.kdb.both`
  - h. Notice that the alias is in upper case **Client**. An alias with lower case client must not be deleted.  
`gsk8capicmd_64 -cert -delete -label Client -db TWClientKeyStore.kdb -stashed`
  - i. `gsk8capicmd_64 -cert -delete -label ClientNew -db TWClientKeyStore.kdb -stashed`
  - j. `gsk8capicmd_64 -cert -delete -label Server -db TWClientKeyStore.kdb -stashed`
  - k. `gsk8capicmd_64 -cert -delete -label ServerNew -db TWClientKeyStore.kdb -stashed`
4. Delete the default certificates from the trusted certificates PEM file, ca.crt, on the **FTAs** and **DMs**

### <10.2.3

- a. `cd <Install_Dir>/TWSDATA/ssl/OpenSSL`
  - b. `cp ca.crt ca.crt.both`
  - c. The new certificates should be at the top and the default certificates should be at the bottom of the file. Run the following commands to extract the top two certificates:
    - Linux:**  
`(openssl x509; openssl x509) < ca.crt > ca.crt.new`  
`mv ca.crt.new ca.crt`
    - Windows:**  
`(openssl x509 & openssl x509) < ca.crt > ca.crt.new`  
`ren ca.crt.new ca.crt`
  - d. Restart the FTAs and DMs
  - e. Verify that they link with the MDM
5. When AgentCertificateDownloader was run for the **DA on the MDM**, it updated the certificates used by the FTA also. Run the following commands to verify that the FTA on the MDM is using the new certificates.

- a. The new certificates should be present in the following dir:

### <10.2.3

```
cd <Install_DIR>/TWSDATA/ssl/OpenSSL
```

```
ls -l
```

```
tls.key
```

```
tls.crt
```

```
tls.sth
```

```
ca.crt
```

- b. Verify that the following parameters in **localopts** are pointing to the correct path:

```
SSL key = "<Install_DIR>/TWSDATA/ssl/OpenSSL/tls.key"
SSL certificate = "<Install_DIR>/TWSDATA/ssl/OpenSSL/tls.crt"
SSL key pwd = "<Install_DIR>/TWSDATA/ssl/OpenSSL/tls.sth"
SSL CA certificate = "<Install_DIR>/TWSDATA/ssl/OpenSSL/ca.crt"
SSL random seed = "<Install_DIR>/TWSDATA/ssl/OpenSSL/tls.rnd"
```

- 
6. Delete the default certificates from the trusted certificates PEM file, **ca.crt**, from the **FTA** on the **MDM**. This step must be performed at the end after all components have been updated with new certificates.

**<10.2.3**

- a. `cd <Install_Dir>/TWSDATA/ssl/OpenSSL`
- b. `cp ca.crt ca.crt.both`
- c. The new certificates should be at the top and the default certificates should be at the bottom of the file. Run the following commands to extract the top three certificates:  

```
(openssl x509; openssl x509; openssl x509) < ca.crt > ca.crt.new  
mv ca.crt.new ca.crt
```
- d. Verify that the FTAs are still linked with the MDM

## 4.10 Certificates in each Store Before and After Updating all Components

The following tables depicts the status of certificates in each store before and after updating all components.

| Component           | Store & Files   | Default Certs  | Custom Certs/After AgentCertificateDownloader  | Delete Certs   |
|---------------------|---|--|--|--|
| MDM, BKM, DDM, BDDM | TWSServerKeyFile.jks  | Alias name: <b>server</b> CN=ServerNew   | Alias name: <b>server</b> CN=ServerNew   | Alias name: <b>server</b> CN=ServerNew                 |
|                     |   |  | Alias name: <b>glowfish</b> CN=glowfish...<br>Alias name: <b>intermediate</b> CN=...<br>Alias name: <b>root</b> CN=... |  |
|                     | TWSServerTrustFile.jks  | Alias name: <b>twstrustkey</b> CN=ServerNew  | Alias name: <b>twstrustkey</b> CN=ServerNew  | Alias name: <b>twstrustkey</b> CN=ServerNew            |
|                     |   | Alias name: <b>twstrustkeyold</b> CN=Server  | Alias name: <b>twstrustkeyold</b> CN=Server  | Alias name: <b>twstrustkeyold</b> CN=Server            |
|                     |   | Alias name: <b>client</b> CN=ClientNew   | Alias name: <b>client</b> CN=ClientNew   | Alias name: <b>client</b> CN=ClientNew                 |
|                     |   | Alias name: <b>clientold</b> CN=Client   | Alias name: <b>clientold</b> CN=Client   | Alias name: <b>clientold</b> CN=Client                 |
|                     |   | Alias name: <b>glowfish</b> CN=glowfish...<br>Alias name: <b>intermediate</b> CN=...<br>Alias name: <b>root</b> CN=... |  |  |
|                     | BrokerWorkstation.properties  | Broker.AuthorizedCNs= <b>Server;ServerNew</b>  | Broker.AuthorizedCNs=Server;ServerNew;<br><b>glowfish</b>  | Broker.AuthorizedCNs= <b>Server;ServerNew</b>          |
| ssl_config.xml      | serverKeyAlias= <b>'server'</b>                                     | serverKeyAlias= <b>'server'</b>  | serverKeyAlias= <b>'glowfish'</b> [delete server]  |  |
| jwt_variables.xml   | <variable name="mp.jwt.trust.key"<br>value= <b>"twstrustkey"</b> /> | <variable name="mp.jwt.trust.key"<br>value= <b>"twstrustkey"</b> />  | <variable name="mp.jwt.trust.key"<br>value= <b>"glowfish"</b> /> [delete twstrustkey]                                  |  |
| server.xml          | keyAlias= <b>"server"</b>   | keyAlias= <b>"server"</b>  | keyAlias= <b>"\${mp.jwt.trust.key}"</b> [delete server]  |  |
| DA                  | TWSSClientKeyStoreJKS.jks   | Alias name: <b>server</b> CN=ServerNew   | Alias name: <b>server</b> CN=glowfish...   |  |
|                     |   | Alias name: <b>client</b> CN=ClientNew   | Alias name: <b>servernew</b> CN=ServerNew  | Alias name: <b>servernew</b> CN=ServerNew              |
|                     |   |  | Alias name: <b>client</b> and <b>clientnew</b>   | Alias name: <b>client</b> and <b>clientnew</b>         |
|                     | TWSSClientKeyStore.kdb  | ! <b>server</b> CN=ServerNew   | ! <b>ServerNew</b> CN=ServerNew  | ! <b>ServerNew</b> CN=ServerNew                        |
|                     |   | ! <b>serverold</b> CN=Server   | ! <b>Server</b> CN=Server  | ! <b>Server</b> CN=Server                              |
|                     |   | - <b>client</b> CN=ClientNew   | ! <b>ClientNew</b> CN=ClientNew  | ! <b>ClientNew</b> CN=ClientNew                        |
|                     |   | ! <b>Client</b> CN=Client [and New Certs]  | ! <b>Client</b> CN=Client [uppercase Client]   |  |
| FTA, DM, BDM        | TWSDATA/localopts   | <b>TWSClient.cer</b>   | <b>tls.crt</b>   |  |
|                     |   | <b>TWSClient.key</b>   | <b>tls.key</b>   |  |
|                     |   | <b>password.sth</b>  | <b>tls.sth</b>   |  |
|                     |   | <b>TWSClientCertificates.cer</b>   | <b>ca.crt</b> (2 root chain certs + 4 certs from TWSClientCertificates.cer + ca.crt)                                   | <b>ca.crt</b> (4 certs from TWSClientCertificates.cer) |
|                     |   | <b>TWS.rnd</b>   |  |  |

---

## 4.11 Renewing the Expired Certificate

Each certificate has an expiration date and it must be renewed ahead of that date in order to keep the various components communicating with each other and not interrupt the execution of the workload automation.

The AgentCertificateDownloader script is also capable of downloading certificates automatically 15 days before the expiration. Place new certificates in PEM format on the MDM's depot directory for the AgentCertificateDownloader script to automatically download and import them.

Follow the steps below to renew the Certificates for the MDM and DWC.

1. Follow the procedure in the organization to request for certificates. If the original CSR created earlier is still available, send it to the CA and have it signed. If it is not available, generate a new CSR according to the process established by the security team.
2. Once the new certificate is received, delete the expired certificate from the KeyStore and TrustStore.
3. Import the new certificate into the KeyStores and TrustStores following the steps described in the previous sections.
4. The CA Certificate chain usually has a long expiration time and may not need to be renewed. If it is approaching expiration or the CA that signs the certificate has changed, import the public certificate chain into the TrustStore.

---

## 5 Other Resources Regarding Replacing Certificates

1. Refer to the following blog for a procedure on how to replace the default certificates with **CA signed** certificates. Note that this procedure adopts the **rip and replace** method where certificates on all components must be replaced at the same time to avoid any down time. This may be suitable for smaller and non-complex environments.

[Replacing Default SSL Certificates with CA signed Custom Certificates](#)

2. Refer to the following Technote for a procedure on how to replace the older weak default certificates with **self-signed** certificates using the **same CN** (Common Name) as the default certificates. Note that this procedure adopts the **rip and replace** method where certificates on all components must be replaced at the same time to avoid any down time. This may be suitable for smaller and simpler environments.

[How to Substitute older weak default certificates \(1024 key size\) with new stronger default certificates \(2048 key size\)](#)

---

## 6 Author's Bio



Sajjad Kabir is a Solutions Architect and a certified Workload Automation consultant with over 30 years of experience. He has a B.Sc. in Computer Systems Engineering from Western Michigan University, Kalamazoo, MI. Sajjad started working with IBM Workload Automation in 1998 while on assignment in IBM Singapore. He has been assisting clients deploy Workload Automation solutions worldwide. He has worked with clients of all sizes and complexities. He has published an IBM Redbook and numerous articles and blogs. He loves to work with people, especially who have just started their endeavors with workload automation. With the transition from IBM to HCL, he has mentored HCL consultants and continued to assist clients with implementing automation solutions on-prem and in the cloud. Sajjad loves to travel, enjoys ethnic cuisines, and SCUBA diving in exotic places around the world.