Installing Custom CA Certificates and Importing SUDI Certificate

By default, the IoT FND OVA comes bundled with keys and certificates which is stored in a keystore. The default values are:

- On IoT FND OVA Linux Host:

  Keystore Location: /opt/fnd/data/

  Keystore Name: cgms_keystore.selfsigned

- On IoT FND container:

  Keystore Location: /opt/cgms/server/cgms/conf/

  Keystore Name: cgms_keystore

- **Default Password: Public123!**

Important

- This is the default password for both the files mentioned above.
- Both these files have the same content.
- When IoT FND container is restarted, the values of /opt/cgms/server/cgms/conf/cgms_keystore file in IoT FND container is overwritten by /opt/fnd/data/cgms_keystore file. If /opt/fnd/data/cgms_keystore file is not present in host, then /opt/fnd/data/cgms_keystore.selfsigned file is used.

When IoT FND OVA is a new installation, each certificate/key entry is referenced by an alias name in the keystore. The default alias are:

- cisco_sudi (cisco root CA certificate with 2029 expiry)
- jmarconi (cisco certificate)
- cgms (self signed certificate that is used by IoT FND when communicating with devices it has to manage)

Note This keystore is specific for certificates used for IoT FND communication with its managed devices. There is a different keystore for web certificate.

Custom cgms_keystore

The cgms certificate in /opt/cgms/server/cgms/conf/cgms_keystore file in IoT FND container and /opt/fnd/data/cgms_keystore.selfsigned file of the linux host has by default self signed certificate of IoT FND. There are two options to build a custom cgms_keystore in /opt/fnd/data location on linux host, where the IoT FND certificate of the customer organisation can be imported and stored.

We can either copy the existing /opt/fnd/data/cgms_keystore.selfsigned file on the Linux host or build it from scratch. After the cgms_keystore file is present on the linux host, if both /opt/fnd/data/cgms_keystore.selfsigned and /opt/fnd/data/cgms_keystore files are present, then /opt/fnd/data/cgms_keystore takes precedence.

Note NTP is a mandatory requirement for Public Key Infrastructure. Hence NTP should be in sync between the issuing Certificate Authority (CA) server, IoT FND, TPS, and FAR/HER. If hostname or IP address has to be changed for the IoT FND host, it has to be done before certificate for IoT FND is issued and hence it should be done before starting to build cgms_keystore.

The SAN field in IoT FND certificate is a mandatory requirement and contains the hostname of the IoT FND server. Any change in hostname or IP address is listed in SAN field (if IP address is also present in the SAN field), then the certificate should be reissued. Depending on the PnP type used, the SAN field contains the hostname of the IoT FND or the IP address or both.

The cgms_keystore should contain the below mandatory certificates/keys:

- Issuing CA certificate of the organisation – This is the certificate of the issuing CA server of the organisation. The issuing CA server can be a root CA server or intermediate CA server. If it is an intermediate CA, it is recommended to import root CA and also intermediate CA certificates into the keystore.
- IoT FND device certificate is issued for IoT FND by issuing CA server.
- Cisco SUDI with 2029 expiry date – This is the cisco manufacturer certificate for IoT FND issued by Cisco with expiry date 2029.
- Cisco SUDI with 2099 expiry date – This is the cisco manufacturer certificate for IoT FND issued by Cisco with expiry date 2099.

The below option shows how to build cgms_keystore file from scratch that contains the required certificates and keys.

Step 1 Change directory to /opt/fnd/data on linux host.

```
# cd /opt/fnd/data
```

Importing Root/Issuing CA Certificate

Step 2 Importing any certificate using keytool command creates the keystore file, if keystore file does not exist. Note that the name of the file has to be cgms_keystore as IoT FND refers the file with this name. Copy the issuing CA certificate of your organisation to any location (using scp or any other file transfer method). In this illustration, it is copied to

/root/rootca.pem. The certificate can be of the format .cer or .crt or .pem. In this illustration, the issuing CA is the root CA and hence the alias name root is used.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore -alias
 root -file /root/rootca.pem
```

Convert the keystore from jks to pkcs12.

```
# keytool -importkeystore -srckeystore /opt/fnd/data/cgms_keystore
-destkeystore /opt/fnd/data/cgms_keystore -deststoretype pkcs12
```

Verify that the file has been created by listing the contents of the keystore.

```
# keytool -list -keystore /opt/fnd/data/cgms_keystore
```

Importing IoT FND Certificate

Step 3 Import the IoT FND certificate.

Note IoT FND certificate has to be imported ONLY with alias name of cgms.

The below steps tells how to generate a key pair .csr file that can be presented to the issuing CA server for a certificate to be granted for IoT FND server. The .csr file is required by few issuing CA servers of few PKI vendors. If the .csr certificate is given to the issuing CA server, then a certificate is generated based on the contents of the .csr file. If a certificate of IoT FND has already been issued, use the following steps, if the IoT FND certificate issued has .pem or .cer or .crt extension. If the IoT FND certificate has .pfx extension, follow step 4.

a) Generate a key pair and .csr file .

```
# keytool -genkeypair -keyalg RSA -keysize 2048 -alias cgms
-ext "SAN=dns.labfnd.cisco.com, ip:1.0.0.1" -keystore /opt/fnd/data/cgms_keystore
-dname CN=labfnd, OU=iotescblr, O=cisco, L=Bengaluru, ST=Karnataka, C=IN"
```

Note The key size in this example is 2048, but 4096 can also be used.

```
# keytool -certreq -file labfnd.csr -keystore
/opt/fnd/data/cgms_keystore -alias cgms -ext "SAN=dns:labfnd.cisco.com,ip:1.0.0.1"
```

Note This .csr file is then presented to the issuing CA server and a certificate is obtained for IoT FND server.

b) Copy the issued certificate to FND server in any location. In this sample, it is copied to /opt/fnd/data as labfnd.pem file. Import the certificate using below command.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore
-alias cgms -file /root/labfnd.pem
```

Step 4 If the FND issued certificate issued has .pfx format, then we will have a .pfx file instead of the .pem file. If it is a .pfx file, check the alias name of the .pfx file and then import it using the alias cgms in the cgms_keystore.

- Find the alias name of the pfx file. In this case, the nms.pfx is copied to the current location.

```
# keytool -list -v -keystore /opt/fnd/data/nms.pfx -srcstoretype
 pkcs12 | grep Alias
```

- Import the pfx into the cgms_keystore with alias cgms. In this sample, "le-IoT FND-8f0908aa-dc8d-4101-a526-93b4eaad9481" is the alias present in the .pfx file.

```
# keytool -importkeystore -v -srckeystore /opt/fnd/data/nms.pfx
-destkeystore /opt/fnd/data/cgms_keystore -srcalias
 le-IoT FND-8f0908aa-dc8d-4101-a526-93b4eaad9481 -destalias cgms
```

Importing SUDI with 2029 Expiry

Step 5 The SUDI certificate with 2029 expiry is present in /opt/fnd/data directory as cisco-sudi-ca.pem. Import this file to cgms_keystore.

```
# keytool -import -trustcacerts -alias cisco_sudi -file
/opt/fnd/data/cisco-sudi-ca.pem -keystore /opt/fnd/data/cgms_keystore
```

Importing SUDI with 2099 Expiry

Step 6 The updated SUDI certificate with 2099 expiry is present in IoT FND container as cisco-ca.pem file, copy this to /opt/fnd/data in linux host.

```
docker cp fnd-container:/opt/cgms/server/cgms/conf/ciscosudi/cisco-ca.pem
/opt/fnd/data/
```

Import the SUDI with 2099 expiry, that is, cisco-ca.pem to cgms_keystore.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore
-alias sudi1 -file /opt/fnd/data/cisco-ca.pem
```

Step 7 Restart the container.

```
docker stop fnd-container
docker start fnd-container
```

Important It is not advised to use the restart command. It is best practice to stop the container and then start the container so the services can stop gracefully. Sometimes restart will not be graceful and can lead to operational issues.

Step 8 After restart, verify that the contents of the cgms_keystore in the IoT FND container has same contents as that of the cgms_keystore in /opt/fnd/data of linux host using the below command.

```
# keytool -list -v -keystore -/opt/fnd/data/cgms_keystore
```

```
# docker exec -it fnd-container keytool -list -v -keystore /opt/cgms/server/cgms/conf/cgms_keystore
```

Step 9 To configure or change the cgms_keystore password, see Changing Password for more information.

• Changing Password, on page 4

Changing Password

The cgms.properties file should contain the password for cgms_keystore, so that IoT FND application can access the cgms_keystore. Hence the first time cgms_keystore is created, encrypt the password of cgms_keystore and provide this encrypted password in cgms.properties file.

If at any time the password for cgms_keystore is changed, then the changed password has to be encrypted again and updated in the cgms.properties file.

Step 1 Run the following command to encrypt the password for the new **cgms_keystore**. The sample is provided below.

```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
encrypt <keystore password>
```

```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
encrypt cisco123
#2bVvZsq+vsq94YxuAKdaag--
```

Step 2 Modify the **cgms.properties** file in the `/opt/fnd/data` folder, and edit the following line to set the new encrypted **cgms_keystore** password:

```
cgms-keystore-password-hidden=<encrypted new cgms_keystore password>
```

Note: With OVA 4.3.x and above, you can leave the `cgms_keystore.selfsigned` default bundled keystore untouched.

If both the files (**cgms_keystore** and **cgms_keystore.selfsigned**) are present, the **cgms_keystore** will be used by the container.