Cisco IoT Field Network Director Installation Guide-Oracle Deployment, Releases 4.3.x and Later

First Published: 2021-12-17

Last Modified: 2023-12-20

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 1

Before You Install Field Network Director

When installing an Oracle database, review this chapter as preparation for your IoT Field Network Director installation.

This chapter provides details on the hardware and software you must have within your network to support the Cisco IoT Field Network Director (FND) 4.3 application and greater that employs an Oracle deployment:

Note Oracle 18c is supported from Cisco IoT FND Releases 4.4.4, 4.5.x and later. Oracle 19c is supported on Cisco IoT FND Releases 4.6.1 and later.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Minimum System Requirements for Oracle Installation

Note Before downloading and installing the Oracle Database, ensure that the /tmp folder can handle, at a minimum, a 5GB file. After you complete the full installation and are working with the FND user interface, you may need to zip up log files sets and the larger /tmp folder will be of value.

Table 1: Minimum Hardware and Software Requirements in IoT FND and Supporting Systems

Component	Minimum Hardware Requirement	Minimum Software Requirements
Cisco IoT FND application server (or comparable system that meets the hardware and software requirements)	• Processor: • Intel Xeon x5680 2.27 GHz (64-bit) • 4 CPUs • RAM: 16 GB • Disk space: 100 GB • Hardware Security Module or Software Security Module	• Red Hat Enterprise Linux (RHEL) 7.5 and above, 64-bit with all packages installed (software development and web server) See Table 4: Application Server Hardware Requirements Example Profiles For Routers and Endpoints, on page 4 for suggested application server resource allocation profiles. • Internet connection When you access IoT FND from a client browser, the browser connects to the Internet to download the necessary data files from the GIS maps provider. • A license to use SafeNet for mesh endpoint security **Note** IoT FND software bundle includes required Java version.
Cisco IoT FND TPS Proxy See Table 3: Tunnel Provisioning Server (TPS), on page 4	• Processor: • Intel Xeon x5680 2.27 GHz (64-bit) • 2 CPUs (virtual cores) • RAM: 4 GB • Hard Disk space: 100 GB	• RHEL 7.5 and above with all packages installed (software development and web server) • Internet connection **Note** IoT FND software bundle includes required Java version.

Component	Minimum Hardware Requirement	Minimum Software Requirements
Database server for IoT FND Scalable to 5,000/5,000,000 endpoints with minimum hardware requirements. See Table 2: Oracle DB Server Hardware Requirements Example Profiles, on page 4 for additional deployment sizes.	• Processor: Intel Xeon x5680 3.33 GHz (64-bit) • 4 CPUs • RAM: 32 GB • Disk space: 150 GB	You will install both Linux and Oracle software on the Database server. Install Linux packages on the Database server before you install Oracle: • RHEL 7.5 and above, 64-bit with all packages installed (software development and web server) **Note** RHEL 8.5 is supported in Cisco IoT FND 4.8.1 release. • Oracle Database 19c Enterprise Edition for IoT FND 4.6.1 and greater. • Oracle Database 18c Enterprise Edition (formerly identified as 12.2c) for IoT FND 4.5.x. 4.6.x and greater. • Oracle Database 12c2 Enterprise Edition Release for IoT FND 4.3.x, 4.4.x, and 4.5.x. • Oracle Database 12cR2 Enterprise Edition Release for IoT FND 4.6.x
Hardware Security Module (HSM)	Luna SA appliance, with client software installed on the IoT FND application servers	Luna SA appliance: • Release 6.10.2 firmware **Note** Contact SafeNet to determine if you can run a higher version. • Release 5.4.7-1 software, plus security patches. Luna SA client software: • Release 5.4.7-1 software.

Component	Minimum Hardware Requirement	Minimum Software Requirements
Software Security Module (SSM)	• RAM: 8 GB • Processor: 2 GHz • 2 CPUs	• RHEL 7.5, 64-bit with all packages installed (software development and web server).

Table 2: Oracle DB Server Hardware Requirements Example Profiles

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	16	100
50/50,000	4	16	200
500/500,000	8	32	500
1,000/1,000,000	12	48	1000
2,000/2,000,000	16	64	1000
5,000/5,000,000	20	96	1000
6,000/6,000,000	20	96	1000

Table 3: Tunnel Provisioning Server (TPS)

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	4	50
50/50,000	2	4	100
500/500,000	2	4	100
1,000/1,000,000	2	4	100
2,000/2,000,000	2	4	100
5,000/5,000,000	2	4	100
6,000/6,000,000	2	4	100

Table 4: Application Server Hardware Requirements Example Profiles For Routers and Endpoints

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	16	100

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
50/50,000	4	16	200
500/500,000	4	16	250
1,000/1,000,000	8	16	250
2,000/2,000,000[1]	8	16	500
5,000/5,000,000 [1]	8	32	500
6,000/6,000,000 [1]	8	32	500

[1]

IR800 Router Only Deployment Requirements

When installing IR800 router only deployments, we recommend using OVA deployments:

https://www.cisco.com/c/en/us/td/docs/routers/connectedgrid/iot_fnd/install/ova/installation_ova.html

Table 5: Application Server Hardware Requirements Example Profile for Routers and LoRa Modules

Nodes(IR800)	CPU (Virtual Cores)	Memory (RAM)	Disk Space (GB)
10,000	10	32	500

Table 6: Database Server Hardware Requirements Example Profile For Routers and LoRa Modules

Nodes(IR800)/LoRa Modules	CPU (Virtual Cores)	Memory (RAM)	Disk Space (GB)
10,000/30,000	10	32	500

Obtaining IoT FND and Cisco Network Register Licenses

- Contact your Cisco partner to obtain the necessary licenses to use IoT FND and Cisco Network Register (CNR).
- Obtain a license to use SafeNet as your Hardware Security Module (HSM) for mesh endpoint security.

[1] [1].Clustered installations RAID 10 is mandatory for deployments of 2 million endpoints and above.

Installing the Linux Packages Required for Installing Oracle

If you are installing a new IoT FND deployment that requires Oracle, you will need to first install the following Linux packages on your FND server, in the order listed, before you install the Oracle database:

1. libaio-devel-0.3.106-5.i386.rpm
2. libaio-devel-0.3.106-5.x86_64.rpm
3. sysstat-7.0.2-11.el5.x86_64.rpm
4. unixODBC-libs-2.2.11-10.el5.i386.rpm
5. unixODBC-libs-2.2.11-10.el5.x86_64.rpm
6. unixODBC-2.2.11-10.el5.i386.rpm
7. unixODBC-2.2.11-10.el5.x86_64.rpm
8. unixODBC-devel-2.2.11-10.el5.i386.rpm
9. unixODBC-devel-2.2.11-10.el5.x86_64.rpm

Obtaining IoT FND RPM Packages

Before you install and set up your IoT FND system, ensure that you have the following packages:

RPM Package	Description
cgms-*version_buildnumber* .x86_64.rpm	Contains the IoT FND installer. This is the main RPM that contains the IoT FND application server itself. Install this package on the IoT FND application servers.
cgms-oracle-*version_number* .x86_64.rpm	Contains the scripts and tools to create the IoT FND Oracle database. This package contains the Oracle database template and management scripts. Install this package on the IoT FND database server system.
cgms-tools-*version_number* .x86_64.rpm	Contains a few optional command-line tools. If needed, install this package on the system running the IoT FND application server.
cgms-ssm-*version_number* .x86_64.rpm	Contains the Software Security Module (SSM). Install this package on the system running the IoT FND application server.
cgms-tpsproxy-*version_number* .x86_64.rpm	Contains the TPS proxy application. Install this package on the IoT FND TPS proxy system.

Important Ensure to configure Network Time Protocol (NTP) before you install the certificates. For more information, refer to Configuring NTP Service, on page 7.

Configuring NTP Service

To configure NTP on your RHEL servers:

Before you begin

Configure all RHEL servers (including all servers that run IoT FND) in your IoT FND deployment to have their NTP service enabled and configured to use the same time servers as the rest of the system.

Caution Before certificates are generated, synchronize the clocks of all system components.

Step 1 Configure the /etc/ntp.conf file.

For example:

Example:

```
cat /etc/ntp.conf
...
# Use the same NTP servers on all our Connected Grid systems.
server 0.ntp.example.com
server 1.ntp.example.com
server 2.ntp.example.com
...
```

Step 2 Restart the NTP daemon and ensure that it is set to run at boot time.

Example:

```
service chrony restart
chkconfig chrony on
```

Step 3 Check the configuration changes by checking the status of the NTP daemon.

This example shows that the system at 192.0.2.1 is configured to be a local NTP server. This server synchronizes its time using the NTP server at 10.0.0.0.

Example:

```
# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*192.0.2.1     198.51.100.1    3 u   309 1024  377    0.694    0.899  0.435
 LOCAL(0)        .LOCL.          10 l   36   64  377    0.000    0.000   0.001
```

For information about configuring NTP on RHEL servers, refer to RHEL documentation.

IoT FND Map View Requirements

When your IoT FND installation is complete, you will need to do the following to ready the Map within the application window.

Note On any device tab, click the Map button in the main pane to display a GIS map of device locations. In its Map View pane, IoT FND uses a Geographic Information System (GIS) map to display device locations. However, before you can use this feature, you must configure your firewall to enable access for all IoT FND operator systems to Cisco-provided GIS map file servers.Note: Only IoT FND operator systems have access to the GIS map file servers.

Note The operator browsers will not have access to other Google sites. No Internet access is required for the IoT FND application server.

You must also assign a fully qualified domain name (FQDN) for each IoT FND server installation and provide Cisco at mailto:%20ask-fnd-pm-external@cisco.com with the following:

- The number of IoT FND installation environments (test and production)
- The FQDN of the IoT FND server
- For cluster deployments, the FQDN of any load balancer in the deployment

Note The FQDN is only used to provision and authorize access to the licensed Cisco IoT FND installation and make API calls to Enterprise Google Map to download the map files. No utility operational data or asset information is ever used (that is, sent over Internet) to retrieve Google map files. Map files are retrieved only using geographic location information.

CHAPTER 2

ISO and RPM Image Verification

Introduction

Starting from Cisco IoT FND 4.9.0, you can verify the ISO and RPM images before the installation or upgrade of IoT FND.

For more information, refer to How to Run the Signature Verification Program, on page 10.

Table 7: IoT FND ISO Image Zip File Contents (iot-fnd-<release>-<build number>-signed)

Zip File Contents	Description
1. iot-fnd-<release>-<build number>.iso	Cisco provided image for which signature is to be verified.
2. iot-fnd-<release>-<build number>.iso.signature	Signature generated for the image.
3. FND_RPM_SIGN-CCO_RELEASE.pem	Cisco signed x.509 end-entity certificate contains the public key that is used to verify the signature. This certificate is chained to Cisco root CA and sub CA posted on https://www.cisco.com/security/pki/
4. cisco_x509_verify_release.py	Signature verification program. After downloading the image, its digital signature, and the x.509 certificate, this program is used to verify the 3-tier x.509 certificate chain and the signature. Certificate chain validation is done by verifying the authenticity of end-entity using Cisco-sourced sub CA and root CA (which the script downloads from Cisco).
5. cisco_x509_verify_release.py.signature	Signature generated for the script cisco_x509_verify_release.py.
6. cisco_openpgp_verify_release.py	Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate.

Zip File Contents	Description
7. cisco_openpgp_verify_release.py.signature	Signature generated for the script cisco_openpgp_verify_release.py.
8. FND-rel-binary.gpg	Open-pgp public key is used for verification of the signed RPM.
9. FND-rel-ascii.gpg	Open-pgp public key is used for verification of the signed RPM.

How to Run the Signature Verification Program

Prerequisites:

- Python 2.7.x
- OpenSSL
- Verification scripts running on customer-premises need internet connectivity to reach Cisco to download root and sub-CA certs

To run a signature verification program:

Step 1 Unzip the file `iot-fnd-<release>-<build number>-signed.zip` and `cd` to the folder `iot-fnd-<release>-<build number>-signed`.

Step 2 Extract the public key from the public cert:

```
openssl x509 -pubkey -noout -in FND_RPM_SIGN-CCO_RELEASE.pem > FND-EE-cert.pubkey
```

Expected Result:

```
FND-EE-cert.pubkey is created under the same folder
```

Step 3 Verify the verification scripts using the public key and signature files.

a)
```
openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
cisco_x509_verify_release.py.signature cisco_x509_verify_release.py
```

Expected Result:

```
Verified OK
```

b)
```
openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
cisco_openpgp_verify_release.py.signature cisco_openpgp_verify_release.py
```

Expected Result:

```
Verified OK
```

Step 4 Verify the ISO file.

```
./cisco_x509_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -s
iot-fnd-<release>-<build number>.iso.signature -i iot-fnd-<release>-<build
number>.iso -v dgst -sha512
```

Expected Result:

```
Retrieving CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...

Successfully retrieved and verified crcam2.cer.

Retrieving SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...

Successfully retrieved and verified innerspace.cer.

Successfully verified root, subca and end-entity certificate chain.

Successfully fetched a public key from FND_RPM_SIGN-CCO_RELEASE.pem.

Successfully verified the signature of iot-fnd-<release>-<build number>.iso using
FND_RPM_SIGN-CCO_RELEASE.pem
```

Step 5 Install the ISO image file.

```
 cd /mnt

mkdir iso

mount -t iso9660 -o loop <path>/iot-fnd-<release>-<build number>.iso /mnt/iso

mkdir /tmp/ISO

cp -pRf /mnt/iso /tmp/ISO

umount /mnt/iso/
```

Step 6 Verify if the delivered binary and ASCII keys have matching fingerprints.

a) `gpg FND-rel-binary.gpg`

Expected Result:

```
pub  2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

b) `gpg FND-rel-ascii.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

Step 7 Verify the binary GPG key against the EE cert.

```
./cisco_openpgp_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -G
FND-rel-binary.gpg
```

Expected Result:

```
Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...

Successfully downloaded crcam2.cer.

Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...

Successfully downloaded innerspace.cer.

Successfully verified Cisco root, subca and end-entity certificate chain.

Successfully fetched a public key from FND_RPM_SIGN-CCO_RELEASE.pem.

Successfully authenticated FND-rel-binary.gpg key using Cisco X.509 certificate trust chain.
```

Step 8 Verify the RPM Signature using the GPG ASCII key.

```
sudo rpm --import FND-rel-ascii.gpg
```

```
rpm -K /tmp/ISO/iso/cgms-<release>-<build number>.x86_64.rpm
```

Expected Result:

```
/tmp/ISO/iso/cgms-<release>-<build number>.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

Step 9 Repeat Step 8 for all the RPMs. Once the RPM is verified, you can install or upgrade the RPM.

CHAPTER 3

Generating and Installing Certificates

This section describes how to generate and install certificates, and includes the following topics:

Information About Certificates

The following topics provide information on certificates:

Types of Certificates

IoT FND uses the following three types of certificates:

1. Device Certificate
2. Web Certificate
3. CSMP Certificate

Role of Certificates

All communications between the CGR1000, IR800s, C800s, ESR C5921s, IR1100, IR8100, IC3000, IXM, and the Cisco Connected IoT Field Network Director (IoT FND) must be authenticated in both directions through mutual authentication. Before mutual authentication occurs, the Cisco IoT FND and the device must each have a certificate that is signed by the same Certificate Authority (CA). You can employ either a root CA or subordinate CA (sub-CA).

For more information on generating certificates for CGRs, refer to Certificate Enrollment Guide for the Cisco 1000 Series Connected Grid Routers.

Generating certificates for IoT FND also involves generating and loading certificates on the IoT FND TPS Proxy (tpsproxy). After generating the certificates, import them into the storage location on the TPS proxy and IoT FND known as the Keystore, on page 14.

Keystore

The keystore provides details for a specific system such as IoT FND server or TPS proxy. IoT FND server and TPS server use Java keystore to store their own device certificate and the corresponding private keys that are referenced using an alias.

The following table lists the certificate types used by IoT FND and the keystore location of the certificates:

IoT FND Certificate Type	Keystore Location
Device Certificate	`cgms_keystore`
Web Certificate	`jbossas.keystore`
CSMP Certificate	HSM or SSM

Note By default, the `cgms_keystore` file does not exist, the file must be created.

The keystore must contain the following information:

- Cisco SUDI or the device manufacturing certificate (private key for the system).
- The root CA certificate of the issuing root CA or sub-CA server.
- The certificate that is generated for FND server or TPS server for TPS.

Note The IoT FND key and the certificates are stored in `cgms_keystore` file in the IoT FND server located in `/opt/cgms/server/cgms/conf` directory.

Generating and Exporting Certificates

Note The IoT FND certificate encrypts data in the database. **Do not lose this certificate!** Loss of this certificate results in some database data that will not be able to be decrypted.

Complete the following procedures to generate and export certificates:

Configuring a Certificate Template for IoT FND and the IoT FND TPS Proxy

On the CA (or subCA) you must create certificate templates to generate certificates for the IoT FND and TPS proxy.

To create a certificate template:

Step 1 Open the Certificate Authority application on a Windows Server 2008 R2 system operating with the Enterprise edition.

The Certificate Authority application is standard on the above noted Windows Server version.

Step 2 Expand the menu to view the Certificate Templates folder.

Step 3 Right-click **Certificate Templates** and choose **Manage** from the context menu.

Step 4 In the right-pane, right-click **Computer**, choose **Duplicate Template** from the context menu, and enter **NMS**.

Step 5 In the Duplicate Template pane, select **Windows Server 2008 Enterprise**.

Step 6 Click **OK**.

Step 7 Click the **NMS Properties** > **General** tab, and do the following:

a) Enter**NMS** in the **Template display name** and **Template name** fields.
b) Enter an appropriate **Validity** period, which defines the lifetime of the certificate.
c) Check the **Publish certificate in Active Directory** check box.
d) Click **OK**.

Step 8 Click the **NMS Properties > Extensions** tab, and do the following:

a) Select **Application Policies** in the Extensions pane.
b) In the Application Policies pane, verify that Client Authentication and Server Authentication appear in the bottom pane.
c) Select **Key Usage** in the Extensions top pane and click **Edit**.
d) In the **Edit Key Usage Extension** pane, clear the **Make this extension critical** check box.
e) Click **OK**.

Step 9 Click the **NMS Properties** > **Request Handling** tab, and do the following:

a) Choose **Signature and encryption** from the Purpose drop-down menu.
b) Check the **Allow private key to be exported** check box.
c) Click **OK**.

Step 10 Click the **NMS Properties** > **Security** tab, and do the following:

a) Select **Administrator** within the Group or user names pane.
b) For each group or user names item listed (such as authenticated users, administrator, domain administrators, enterprise administrators) check the **Allow** check box for all permissions (full control, read, write, enroll, autoenroll).
c) Click **OK**.

Step 11 Click the **NMS Properties** > **Cryptography** tab, and retain the following default settings:

- Algorithm name: RSA
- Minimum key site: 2048
- Cryptographic provider: Requests can use any provider available on the subject computer
- Request hash: SHA256

Step 12 Click **OK**.

Step 13 Click the **NMS Properties > Subject Name** tab, and retain the following default settings:

- Radio button for **Supply in the request radio button** selected

- Check box checked for **Use subject information from existing certificates for autoenrollment renewal requests**

Step 14 Click **OK**.

Note Retain the default settings for the remaining tabs: Superseded Templates, Server, and Issuance Requirements.

Enabling a Certificate Template

To enable the certificate template:

Before you begin

Before you can create a certificate, you must enable the certificate template.

Step 1 Configure a certificate template (see Configuring a Certificate Template for IoT FND and the IoT FND TPS Proxy, on page 14).

Step 2 Open the Certificate Authority application on the Windows Server.

Step 3 Expand the menu to view the Certificate Templates folder.

Step 4 Right-click **Certificate Templates** and choose **New** > **Certificate Template to Issue** from the context menu.

Step 5 In the Enable Certificate Templates window, highlight the new **NMS** template.

Step 6 Click **OK**.

Generating Certificates for IoT FND and the IoT FND TPS Proxy

Follow the same steps for generating a certificate for IoT FND and for the TPS proxy by using the configuration template that you previously created.

Note Go through the steps in this section twice: once to generate the IoT FND certificate, and once to generate the TPS proxy certificate.

In the Certificate Properties window, click the **Subject** tab, and do the following:

- In the **Value** field, add the fully-qualified domain name (FQDN): the value you enter depends on whether you are creating a certificate for the IoT FND or the TPS proxy.

After creating these two certificates, securely transfer the IoT FND certificate to the IoT FND application server, and securely copy the TPS proxy certificate to the TPS proxy server.

To generate a certificate:

Step 1 Configure a certificate template (see Configuring a Certificate Template for IoT FND and the IoT FND TPS Proxy, on page 14).

Step 2 Enable the certificate template (see Enabling a Certificate Template, on page 16).

Step 3 From a server running Windows Server 2008, choose **Start** > **Run** and enter **mmc** to open the MMC console.

Step 4 In the Console 1 window, expand the **Certificates** > **Personal** folders.

Step 5 Right-click **Certificates** and choose **All Tasks** > **Request New Certificate** from the context menu.

Step 6 In the Before You Begin window, click **Next**.

Step 7 In the Select Certificate Enrollment Policy window, choose **Active Directory Enrollment Policy**. Click **Next**.

Step 8 In the Request Certificates window, do the following:

a. Check the **NMS** check box.

b. Click the **More information**... link.

Step 9 In the Certificate Properties window, click the **Subject** tab, and do the following:

a. From the Type drop-down menu, choose **Common name (CN)**.

b. In the **Value** field, add the fully-qualified domain name (FQDN):

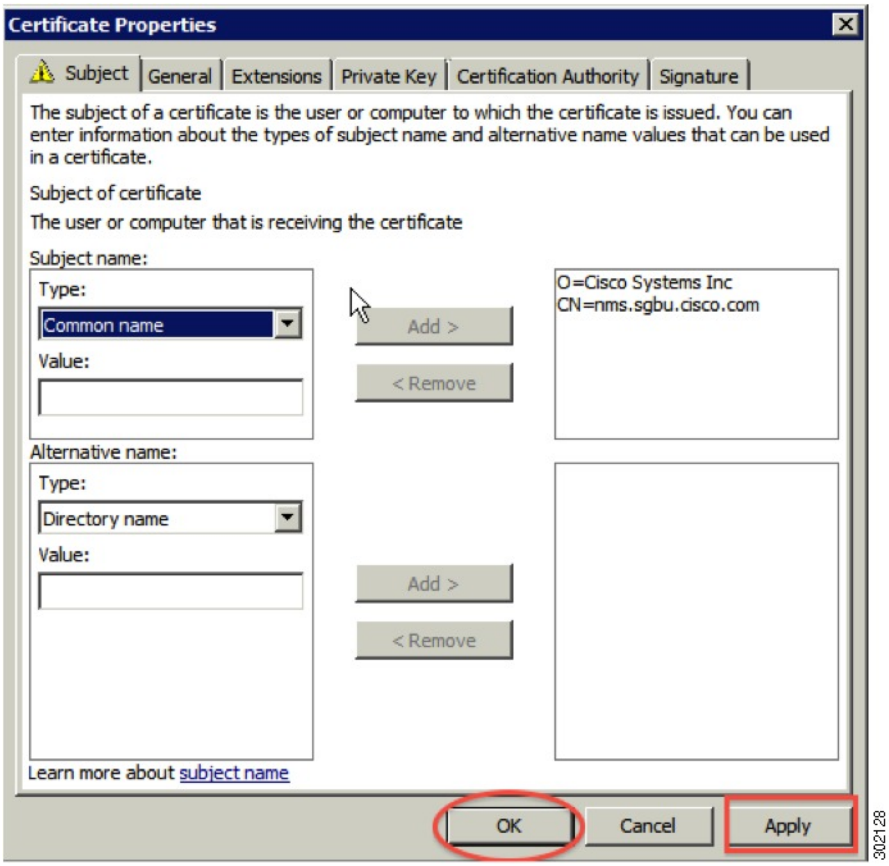
- For IoT FND certificates, enter the FQDN of the IoT FND server for your deployment, for example: CN=nms.sgbu.cisco.com.
- For TPS proxy certificates, enter the FQDN for the TPS proxy for your deployment, for example: CN= tps.sgbu.cisco.com.

c. Click **Add** and the Common Name appears in the right-pane.

d. From the Type drop-down menu, choose **Organization (O)**.

e. In the **Value** field, add the company name or organization for the IoT FND or TPS proxy.

f. Click **Add** and the organization appears in the right-pane.


Certificate Properties

The subject of a certificate is the user or computer to which the certificate is issued. You can enter information about the types of subject name and alternative name values that can be used in a certificate.

Subject of certificate

The user or computer that is receiving the certificate

Subject name:

Type: Common name

Value:

O=Cisco Systems Inc
CN=nms.sgbu.cisco.com

Alternative name:

Type: Directory name

Value:

Step 10 Click **Apply**. Click **OK**.

Step 11 In the Certificate Enrollment window, check the **NMS** check box and click **Enroll**.

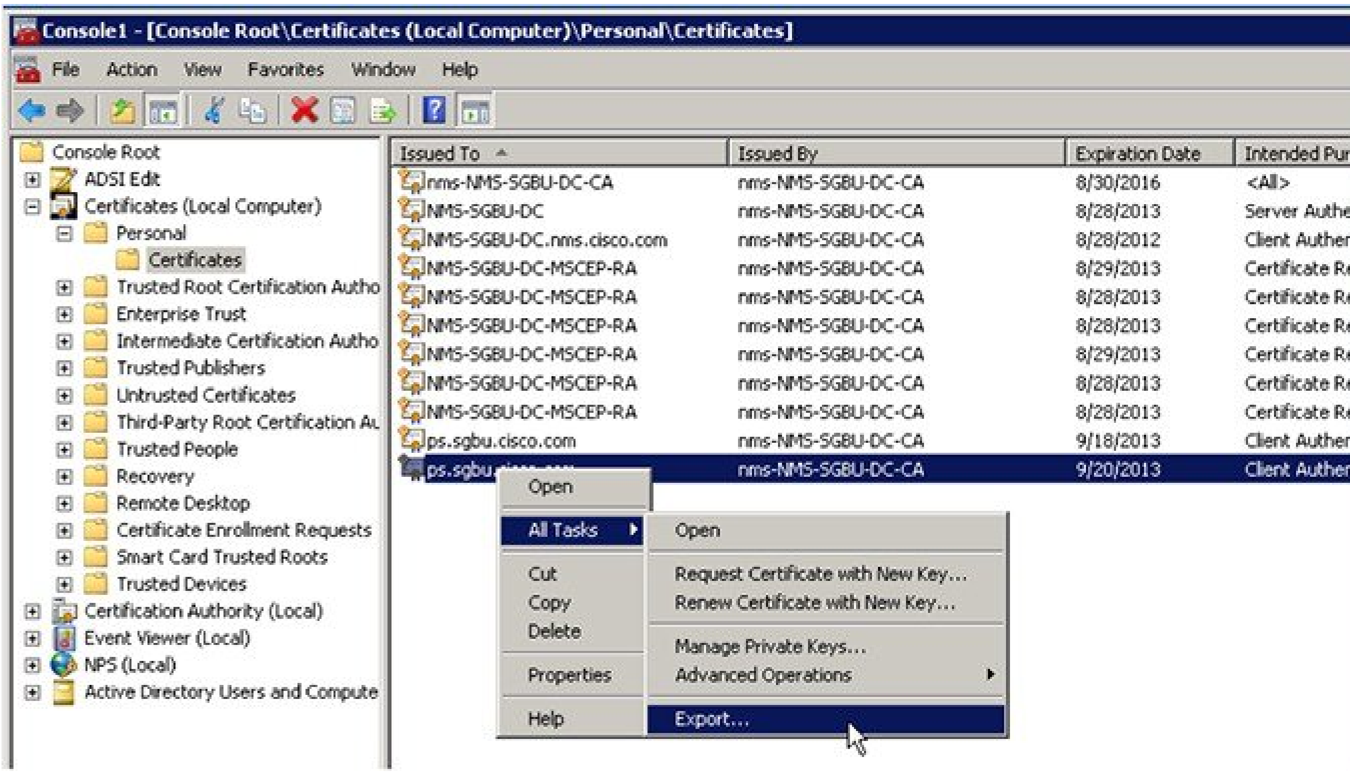
Step 12 After enrollment completes, click **Finish**.

Step 13 In the MMC console (Console 1), expand the **Certificates** folder.

Step 14 Choose **Personal** > **Certificates**.

Step 15 In the Issued To pane, right-click the new certificate and choose **All Tasks** > **Export** fromthe context menu.

The Export Wizard window appears.


Console1 - [Console Root\Certificates (Local Computer)\Personal\Certificates]
File Action View Favorites Window Help
Console Root
ADSI Edit
Certificates (Local Computer)
Personal
Certificates
Trusted Root Certification Autho
Enterprise Trust
Intermediate Certification Autho
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Au
Trusted People
Recovery
Remote Desktop
Certificate Enrollment Requests
Smart Card Trusted Roots
Trusted Devices
Certification Authority (Local)
Event Viewer (Local)
NPS (Local)
Active Directory Users and Compute
Open
All Tasks
Cut
Copy
Delete
Properties
Help
Open
Request Certificate with New Key...
Renew Certificate with New Key...
Manage Private Keys...
Advanced Operations
Export...

Issued To	Issued By	Expiration Date	Intended Pur
nms-NMS-SGBU-DC-CA	nms-NMS-SGBU-DC-CA	8/30/2016	<All>
NMS-SGBU-DC	nms-NMS-SGBU-DC-CA	8/28/2013	Server Authe
NMS-SGBU-DC.nms.cisco.com	nms-NMS-SGBU-DC-CA	8/28/2012	Client Auther
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/29/2013	Certificate R
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/28/2013	Certificate R
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/28/2013	Certificate R
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/29/2013	Certificate R
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/28/2013	Certificate R
NMS-SGBU-DC-MSCEP-RA	nms-NMS-SGBU-DC-CA	8/28/2013	Certificate R
ps.sgbu.cisco.com	nms-NMS-SGBU-DC-CA	9/18/2013	Client Auther
ps.sgbu	nms-NMS-SGBU-DC-CA	9/20/2013	Client Auther

Step 16 Initiate the Export Wizard.

Step 17 At the Export Private Key window, select the **Yes, export the private key** radio button. Click **Next**.

Step 18 At the Export File Format window, do the following:

a. Click the **Personal Information Exchange** radio button.

b. Check the **Include all certificates in the certification path if possible** check box.

This option includes the full certificate chain within the certificate.

c. Click **Next**.

Step 19 In the password window, enter **keystore** and re-enter to confirm.

The password is the default password that the IoT FND and the TPS proxy use to read this file.

Step 20 Click **Next**.

Step 21 In the File to Export window, enter the file name (such as *nms_cert* or *tps_cert*) and click **Next**.

Step 22 In the Completing the Certificate Export Wizard, click **Finish**.

Files with a *.pfx extension are automatically saved to the Desktop. PFX refers to the Personal Information Exchange format, which is also known as PKCS_#12 format. PFX is an industry-standard format that allows certificates and their private keys to be transferred (exported) from one computer to another.

Step 23 Securely transfer the two certificate files (such as *nms_cert.pfx* and *tps_cert.pfx*) from the Windows Desktop to the IoT FND (*nms_cert.pfx*) and TPS proxy *(tps_cert.pfx*), respectively.

Note For heightened security, after a successful transfer delete the *.pfx files from the Windows Desktop and empty the Recycle bin.

Certificate Requirements for IoT FND Server HA Deployment

To generate the certificate for IoT FND server HA deployment, follow the same steps as in the Generating Certificates for IoT FND and the IoT FND TPS Proxy.

Ensure that IoT FND server certificate contents for HA deployment is as given below:

- The Subject — Must have the FQDN of the VIP.

  Example: FNDSERVERVIP.TEST.COM

- The Subject Alternative Name (SAN) — Added must include the FQDN of the VIP.

  Example: FNDSERVERVIP.TEST.COM (same as the subject)

- The Subject Alternative Name — Must NOT have the individual server names.

  Example: It must not contain FNDSERVER1.TEST.COM, FNDSERVER2.TEST.COM

Command Authorization Support

The Cisco Connected Grid Routers (CGRs) are managed by IoT FND over a WAN backhaul connection such as 3G, 4G, or WiMAX. For CG-OS CGRs, you define an OID value to enable administrative privileges for IoT FND.

⚠ **Attention** Defining OID value in the cgms certificate for TPS and FND is required until IoT FND release 4.7 and from IoT FND release 4.8 onwards, it is not required.

The OID for this policy is 1.3.6.1.4.1.9.21.3.3.1. This element appears in the certificate if IoT FND is authorized to issue management commands to the CGR with administrative privileges. When IoT FND communicates with the CGR over a secured session, such as TLS, the CGR can execute these commands as if they were issued by the network administrator.

Enabling Command Authorization Using NMS/TPS Certificates

Follow this procedure to authorize the command authorization (CA) feature of the router, and complete registration with IoT FND.

Step 1 Generate new NMS/TPS certificates (see Generating Certificates for IoT FND and the IoT FND TPS Proxy, on page 16) or renew the existing NMS/TPS certificate (see Renewing Certificates, on page 23).

Step 2 Add an OID value to the CA certificate (see Adding an OID Value to the CA Certificate, on page 21).

Step 3 Generate a new.pfx file for the NMS/TPS certificate (see Generating Certificates for IoT FND and the IoT FND TPS Proxy, on page 16).

Step 4 Stop IoT FND by running the following command:

RHEL Version	Command
8.x	`systemctl stop cgms`
7.x	`service cgms stop`

Note The application typically takes approximately 10 seconds to stop. Run `ps | grep java` to verify that no Java processes are running.

Step 5 Rename the existing cgms_keystore file (for example, cgms_keystore_no_oid).

Step 6 Export the.pfx file to IoT FND and create a new cgms_keystore file (see Using Keytool to Create the cgms_keystore File, on page 32).

Step 7 Install the new certificates (see Installing the Certificates, on page 31).

Step 8 Add the new cgms_keystore file to IoT FND (see Copying the cgms_keystore File to IoT FND, on page 33).

Step 9 Restart IoT FND by running the following command:

RHEL Version	Command
8.x	`systemctl restart cgms`
7.x	`service cgms restart`

Step 10 Register the routers with IoT FND.

Adding an OID Value to the CA Certificate

Note The procedure is applicable only for CG-OS devices and not for Cisco IOS and Cisco IOS-XE devices.

You must add an OID value to the CA certificate to allow IoT FND to use the admin role for command authorization on the router.

To add an OID value to the CA certificate:

Step 1 On the CA server, open a cmd console and type:

```
certutil -setreg policy\EnableRequestExtensionList +1.3.6.1.4.1.9.21.3.3.1
```

Step 2 Restart the CA.

Step 3 In the Select Certificate Enrollment Policy window, choose **Active Directory Enrollment Policy** and click **Next**.

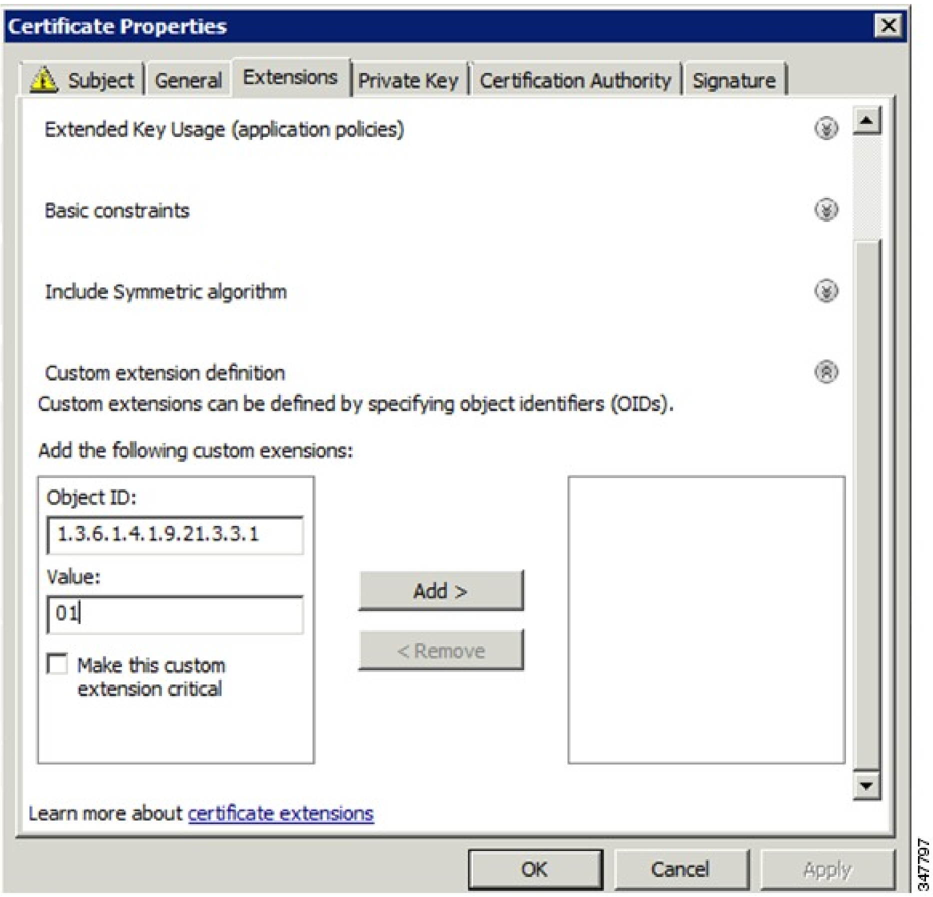
Step 4 In the Request Certificates window, do the following:

a) Check the **NMS** check box

b) Click the **More information**... link

Step 5 In the Certificate Properties window, click the **Subject** tab and complete the fields.

Step 6 In the Certificate Properties window, click the **Extensions** tab and click the **Custom extension definition** button to expand the section.



Custom extensions can be defined by specifying object identifiers (OIDs).

Add the following custom exensions:

Object ID: 1.3.6.1.4.1.9.21.3.3.1

Value: 01

□ Make this custom extension critical

Learn more about certificate extensions

Step 7 Type the following in the **Object ID** field:

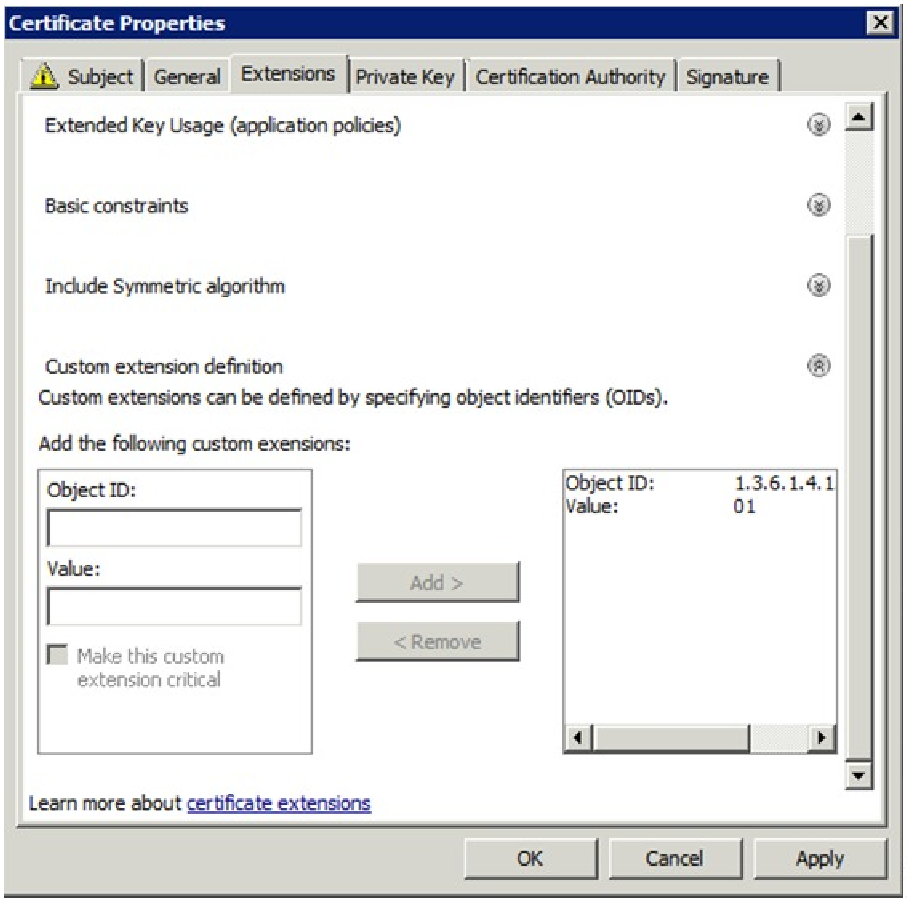
```
1.3.6.1.4.1.9.21.3.3.1
```

Step 8 In the **Value** field, type:

```
01
```

Step 9 Click **Add**.

The OID and Value are added to the field at the right as custom extensions.



Custom extensions can be defined by specifying object identifiers (OIDs).

Add the following custom exensions:

Object ID:

Value:

□ Make this custom extension critical

```
Object ID:        1.3.6.1.4.1
Value:            01
```

Learn more about certificate extensions

Step 10 Ensure that these values are correct, and then click **Apply**.

Renewing Certificates

To renew certificates and add the OID value:

Step 1 From the RSA CA server with the original NMS/TPS certificate, type the following open command at the command prompt:

```
certutil -setreg policy\EnableRequestExtensionList +1.3.6.1.4.1.9.21.3.3.1
```

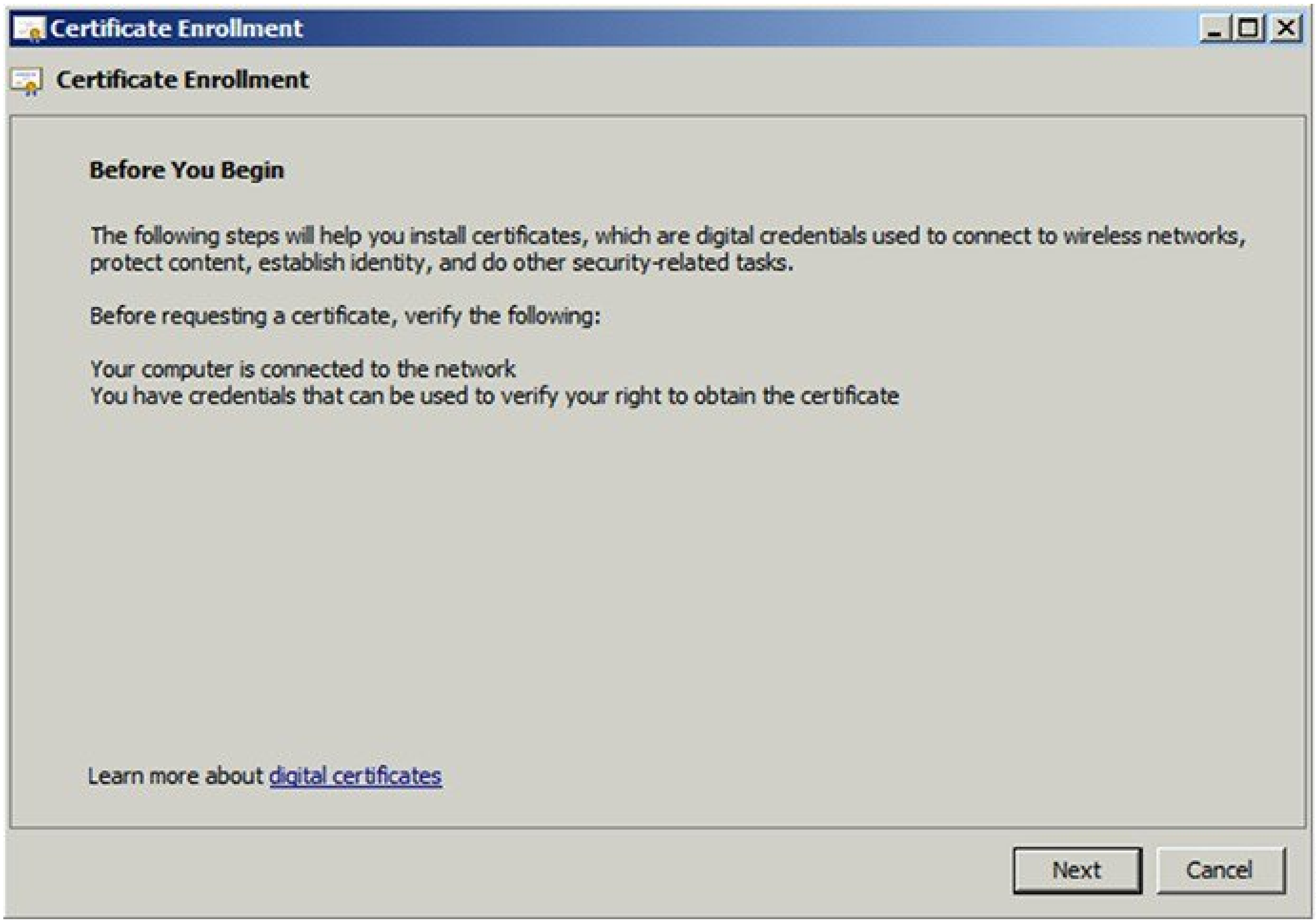
Step 2 Restart the CA server.

Step 3 Open the certificate console in the MMC.

Step 4 Locate the issued NMS/TPS certificate in the Personal folder on the CA server.

Step 5 Right-click on the server icon, and select **All Tasks** > **Advanced Operations** > **Renew This Certificate with the Same Key** option from the context menu.

Step 6 In the Certificate Enrollment window, click **Next**.



Before You Begin

The following steps will help you install certificates, which are digital credentials used to connect to wireless networks, protect content, establish identity, and do other security-related tasks.

Before requesting a certificate, verify the following:

Your computer is connected to the network
You have credentials that can be used to verify your right to obtain the certificate

Learn more about digital certificates

Step 7 Click **Details**.

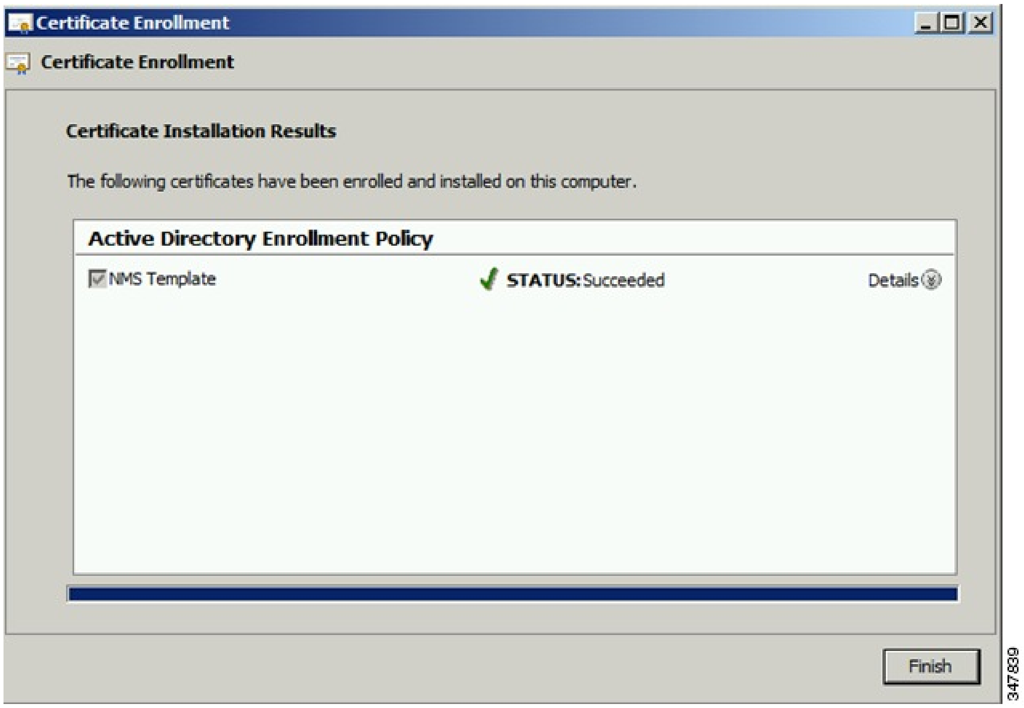
Step 8 Click **Properties**.

Step 9 Enter the OID and its value, and click **OK**.

Step 10 Click **Add**, and then click **OK**.

Step 11 At Request Certificates panel, click **Enroll**.

Step 12 Click **Finish**.



Certificate Enrollment

Certificate Enrollment

Certificate Installation Results

The following certificates have been enrolled and installed on this computer.

Active Directory Enrollment Policy

☑ NMS Template ✓ STATUS: Succeeded Details

Step 13 Verify that the certificate contains the OID value.

Configuring a Custom CA for HSM

This section describes configuring a custom CA for the hardware security module (HSM) for signing CSMP messages sent from IoT FND to mesh devices.

To configure a custom CA for generating HSM certificates:

Before you begin

- Ensure that you install the SafeNet client software version listed in the system requirements in the IoT FND Release Notes on the IOT-FND server.
- You must have your own CA (for example, Microsoft or OpenSSL).

Step 1 Create a new partition on the HSM and assign it to your IoT FND client (see Setting Up the HSM Client, on page 45).

Step 2 Generate a key pair on the HSM and export a CSR for that key pair (see Keystore, on page 14).

All commands run from the Luna client on the IoT FND server. You do not have to log in to the HSM machine.

```
[root@<user>-scaledb bin]# cd /usr/safenet/lunaclient/bin/

# Generate a Key Pair (a set of private and public keys.
You MUST provide explicit labels to the private and public keys)
[root@<user>-scaledb bin]#./cmu generatekeypair -sign=T
-verify=T -labelpublic="nms_public_key" -labelprivate="nms_private_key"
Please enter password for token in slot 1 : ********
```

```
Enter key type - [1] RSA [2] DSA [3] ECDSA : 3 <--- Choose option 3
Enter curve type [1] NISTP 192
[2] NISTP 224
[3] NISTP 256
[4] NISTP 384
[5] NISTP 521

Enter curve type [1] NISTP 192
[2] NISTP 224
[3] NISTP 256 <--- Choose option 3
[4] NISTP 384
[5] NISTP 521
(1 to 5) 3
[root@<user>-scaledb bin]#

# Test if the keypair exists on the HSM partition

[root@<user>-scaledb bin]#./cmu list
Please enter password for token in slot 1 : ********
handle=2000001 label=nms_public_key
handle=2000002 label=nms_private_key

# Now, export a certificate signing request for this keypair.
Note that the specific fields for DN and handle may be different for your HSM.
Fill appropriately.
[root@<user>-scaledb bin]#./cmu requestcertificate
Please enter password for token in slot 1 : ********
Select the private key for the request :

Handler Label
2000002 nms_private_key
Enter handler (or 0 for exit) : 2000002
Enter Subject 2-letter Country Code (C) : US
Enter Subject State or Province Name (S) : CA
Enter Subject Locality Name (L) : San Jose
Enter Subject Organization Name (O) : Cisco Systems Inc.
Enter Subject Organization Unit Name (OU) : IOTSSG
Enter Subject Common Name (CN) : IOT-FND-HSM
Enter EMAIL Address (E) :
Enter output filename : hsm.csr
[root@<user>-scaledb bin]#

# Verify the file exists and has properly formatted content

[root@<user>-scaledb bin]# ls hsm.csr
hsm.csr

[root@<user>-scaledb bin]# cat hsm.csr
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBKzCB0QIBADBvMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExETAPBgNVBAcT
CFNhbiBKb3NlMRowGAYDVQQKExFDaXNjbyBTeXN0ZW1zIEluYzEPMA0GA1UECxMG
SW9UU1NHMRMwEQYDVQQDEwpDRy1OTVMtSFNNMFkwEwYHKoZIzj0CAQYIKoZIzj0D
AQcDQgAESfdlrrcVtzN3Yexj9trlI5qd0w5Sdu8Vj2s17JAF/vPFUOYIw/uXwD6+
bb8vq3WHlA6tmgRbj+FU6G3Bmt/vCqAAMAsGByqGSM49BAEFAANIADBFAiEAroJO
qz3dHA2GLrGzBmUO1vYys642Nkb4B4qyEoUZIGsCIFs0iTUyGQreM1BaSDEPHArZ
RvFlrKo/Zi3c8O4gzFZW
-----END NEW CERTIFICATE REQUEST-----
```

Step 3 Save the generated CSR to your CA and sign the certificate.

Note For CR-Mesh release earlier than 6.2.34 MR or 6.3.20, it is recommended to obtain a 30-year certificate.

Starting from CR-Mesh release 6.2.34 MR or 6.3.20 and later, the EST feature is supported, which automates the certificate renewal process. Therefore, it is not required to obtain a 30-year certificate. For more information on EST, see Configuring Enrollment over Secure Transport. You can use the root CA for IEEE 802.1X authentication for node admission.

Step 4 Copy the signed certificate to the IoT FND server and import it to the HSM.

```
[root@<user>-scaledb bin]#./cmu import
Please enter password for token in slot 1 : ********
Enter input filename : <your file name with signed certificate>

# Verify that the certificate was imported

[root@<user>-scaledb bin]#./cmu list
Please enter password for token in slot 1 : ********
handle=2000001 label=nms_public_key
handle=2000002 label=nms_private_key
handle=2000003 label=IOT-FND-HSM <--- This is my certificate with label = CN
```

Step 5 Run the following commands to configure IoT FND to use the new certificate:

RHEL Version	Commands
8.x	**a.** `systemctl stop cgms` **b.** `systemctl start cgms`
7.x	**a.** `service cgms stop` **b.** `service cgms start`

Example Output:

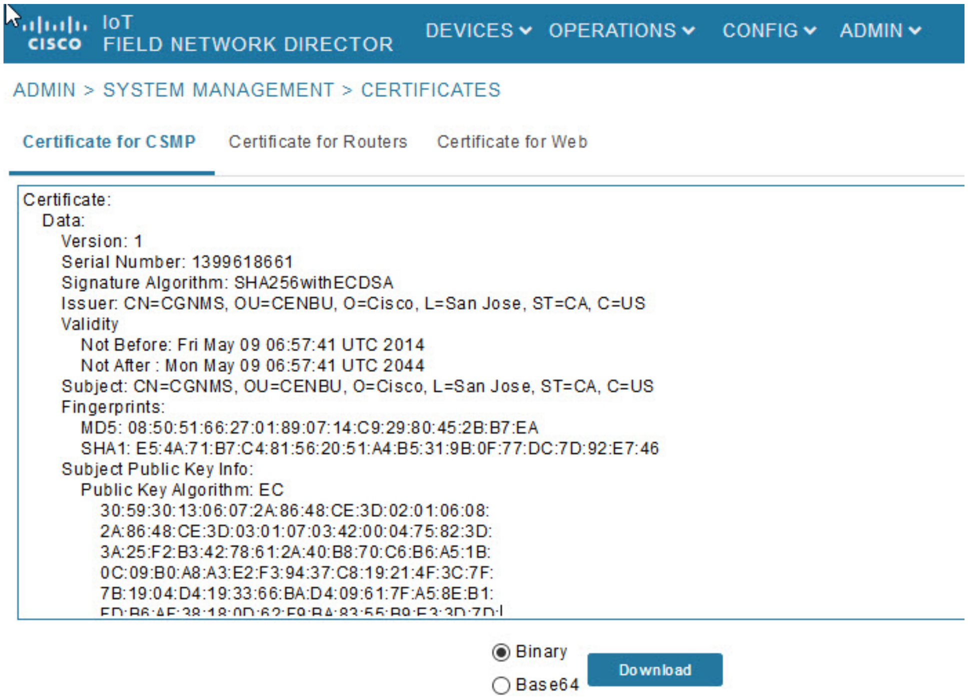
```
[root@kartven2-nms ~]# service cgms stop
[root@kartven2-nms ~]# cd /opt/cgms/server/cgms/conf/

# Add following properties to the cgms.properties file

hsm-private-key-label=nms_private_key <--- private key label you gave to your public key
hsm-public-key-label=nms_public_key <--- public key label you gave to your public key
hsm-cert-label=IOT-FND-HSM <--- label for your signed certificate
hsm-keystore-name=customca-group <--- your HA partition group
hsm-keystore-password=2bVvZsq+vsq94YxuAKdaag== <--- encrypted password for the partition

[root@kartven2-nms conf]# service cgms start
[root@kartven2-nms conf]#
```

Step 6 Verify that the certificate appears on the **Certificates for CSMP** tab (**ADMIN** > **System Management** > **Certificates**).



```
Certificate:
  Data:
    Version: 1
    Serial Number: 1399618661
    Signature Algorithm: SHA256withECDSA
    Issuer: CN=CGNMS, OU=CENBU, O=Cisco, L=San Jose, ST=CA, C=US
    Validity
      Not Before: Fri May 09 06:57:41 UTC 2014
      Not After : Mon May 09 06:57:41 UTC 2044
    Subject: CN=CGNMS, OU=CENBU, O=Cisco, L=San Jose, ST=CA, C=US
    Fingerprints:
      MD5: 08:50:51:66:27:01:89:07:14:C9:29:80:45:2B:B7:EA
      SHA1: E5:4A:71:B7:C4:81:56:20:51:A4:B5:31:9B:0F:77:DC:7D:92:E7:46
    Subject Public Key Info:
      Public Key Algorithm: EC
        30:59:30:13:06:07:2A:86:48:CE:3D:02:01:06:08:
        2A:86:48:CE:3D:03:01:07:03:42:00:04:75:82:3D:
        3A:25:F2:B3:42:78:61:2A:40:B8:70:C6:B6:A5:1B:
        0C:09:B0:A8:A3:E2:F3:94:37:C8:19:21:4F:3C:7F:
        7B:19:04:D4:19:33:66:BA:D4:09:61:7F:A5:8E:B1:
        FD:B6:AF:38:18:0D:62:F9:BA:83:55:B9:E3:3D:7D:
```

Step 7 Configure your mesh nodes to use this certificate for signatures.

Configuring a Custom CA for SSM

This section describes configuring a custom CA for the software security module (SSM) for signing CSMP messages sent from IoT FND to mesh devices.

To configure a custom CA for generating SSM certificates:

Before you begin

- Ensure that you install the SafeNet client software version listed in the system requirements in the IoT FND Release Notes on the IOT-FND server.
- Only SSM versions 2.2.0-37 and above are supported.
- You must have your own CA (for example, Microsoft or OpenSSL).

Step 1 Run the following command to stop the ssm service.

RHEL Version	**Command**
8.x	`systemctl stop ssm`
7.x	`service ssm stop`

Step 2 Use the ssm_setup.sh script to configure a new keypair with a specific alias and generate a CSR:

```
[root@nms-rhel-6-6 ~]# cd /opt/cgms-ssm/bin/
[root@nms-rhel-6-6 bin]# ./ssm_setup.sh

Software Security Module Server

a. Generate a new keyalias with self signed certificate for CSMP

b. Generate a new keypair & certificate signing request for CSMP <--- Choose option 2

c. Import a trusted certificate

d. Change CSMP keystore password

e. Print CG-NMS configuration for SSM

f. Change SSM server port

g. Change SSM-Web keystore password

Select available options. Press any other key to exit.
Enter your choice : 2
Warning: This action will modify ssm_csmp_keystore file. Backup the file before performing this
action.
Do you want to proceed (y/n): y

Enter current ssm_csmp_keystore password :
Enter a new key alias name (8-16): ssmcustomca
Enter key password (8-12):

Enter certificate issuer details

Enter common name CN [Unknown]: IOT-FND-SSM

Enter organizational unit name OU [Unknown]: IOTSSG

Enter organization name O [Unknown]: Cisco Systems Inc.

Enter city or locality name L [Unknown]: San Jose

Enter state or province name ST [Unknown]: CA

Enter country code for this unit C [Unknown]: US


Is [CN=IOT-FND-SSM, OU=IOTSSG, O=Cisco Systems Inc., L=San Jose, ST=CA, C=US] correct (y/n)? :y
Certificate Signing Request file name: /opt/ssmcustomca.csr
Succefully generated keypair with alias ssmcustomca. You can use the CSR from /opt/ssmcustomca.csr
for signature by certificate authority
[root@nms-rhel-6-6 bin]#
```

Step 3 Save the generated CSR to your CA and sign the certificate.

Note For CR-Mesh release earlier than 6.2.34 MR or 6.3.20, it is recommended to obtain a 30-year certificate.

Starting from CR-Mesh release 6.2.34 MR or 6.3.20 and later, the EST feature is supported, which automates the certificate renewal process. Therefore, it is not required to obtain a 30-year certificate. For more information on EST, see Configuring Enrollment over Secure Transport. You can use the root CA for IEEE 802.1X authentication for node admission.

Step 4 Copy the signed certificate to the IoT FND server and import it to the SSM.

Step 5 Use the ssm_setup.sh script to import the two certificates to the SSM keystore:

```
[root@nms-rhel-6-6 bin]# ./ssm_setup.sh

Software Security Module Server

a. Generate a new keyalias with self signed certificate for CSMP

b. Generate a new keypair & certificate signing request for CSMP

c. Import a trusted certificate <--- Choose option 3

d. Change CSMP keystore password

e. Print CG-NMS configuration for SSM

f. Change SSM server port

g. Change SSM-Web keystore password

Select available options. Press any other key to exit

Enter your choice : 3

Enter current ssm_csmp_keystore password :
Enter the alias for import: root
Certificate file name: /opt/ca.crt
Certificate reply was installed in keystore
Succefully imported certificate into alias root
```

Step 6 Use the ssm_setup.sh script to import the signed certificate for the alias:

```
[root@nms-rhel-6-6 bin]# ./ssm_setup.sh

Software Security Module Server

a. Generate a new keyalias with self signed certificate for CSMP

b. Generate a new keypair & certificate signing request for CSMP

c. Import a trusted certificate <--- Choose option 3

d. Change CSMP keystore password

e. Print CG-NMS configuration for SSM

f. Change SSM server port

g. Change SSM-Web keystore password

Select available options.Press any other key to exit

Enter your choice : 3

Enter current ssm_csmp_keystore password :
Enter the alias for import: ssmcustomca
Certificate file name: /opt/ssmcustomca.crt
Certificate reply was installed in keystore
Succefully imported certificate into alias ssmcustomca
```

Step 7 Update the cgms.properties file with the following parameters to configure IoT FND to use this certificate on the SSM for signatures:

```
security-module=ssm
ssm-host=172.27.163.153
ssm-port=8445
ssm-keystore-alias=ssmcustomca
```

```
ssm-keystore-password=GgeQJAOk3fSIH97qJARGRA==
ssm-key-password=GgeQJAOk3fSIH97qJARGRA==
```

Step 8 Verify that the certificate appears on the **Certificates for CSMP** > **ADMIN** > **System Management** > **Certificates** tab.

Step 9 Configure your mesh nodes to use this certificate for signatures.

Exporting the CA Certificate

To export the certificate from the Certificate Authority or subordinate CA to the IoT FND:

Step 1 Open the Certificate Authority application on a Windows Server 2008 R2 system operating with the Enterprise Edition.

Step 2 Expand the menu to view the **Certificates (Local Computer)** > **Personal** > **Certificates** folder.

Step 3 Locate the certificate whose fingerprint matches that in use by the Cisco CGR 1000 and Cisco ASR.

Step 4 Right-click the certificate and choose **All Tasks** > **Export** from the context menu.

Step 5 In the Certificate Export Wizard window, click **Next**.

Step 6 In the Export Private Key window, select the **No, do not export the private key** radio button. Click **Next**.

Step 7 In the Export File Format window, select the **Base-64 encoded X.509 (.CER)** radio button. Click **Next**.

Step 8 In the File to Export window, assign a name for the file that you want to export. Click **Next**.

Step 9 In the File to Export window, enter the file name (such as *ca_cert* or *subca_cert*) and click **Next**.

Step 10 In the Completing the Certificate Export Wizard, click **Finish**.

Files with a *.cer extension are automatically saved to the Desktop.

Step 11 Securely transfer the certificate file (such as *ca_cert.cer*) from the Windows Desktop to IoT FND.

Note For heightened security, after a successful transfer delete the *.cer file from the Windows Desktop and empty the Recycle bin.

Installing the Certificates

You must create a cgms_keystore file on both the servers running IoT FND and IoT FND TPS Proxy.

- **IoT FND** — When creating the cgms_keystore file, you import the IoT FND certificate, its private key, and the certificate chain. After creating the cgms_keystore file, you copy it into a specific directory on the server.
- **IoT FND TPS Proxy** — When you create the cgms_keystore file, you import the IoT FND TPS Proxy certificate, its private key, and the certificate chain. After you create the cgms_keystore file, you copy it into a specific directory on the TPS proxy.

  To create the cgms_keystore file for the TPS proxy and IoT FND, use Keytool and complete the following procedures:

Determine the password to use for the keystore. The examples in this chapter refer to this password as keystore_password.

Using Keytool to Create the cgms_keystore File

To create the cgms_keystore file for both IoT FND and the TPS proxy:

Step 1 As root, view the contents of the.pfx file by entering the following command on the server (IoT FND and TPS proxy):

```
[root@ tps_server
~]# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
```

Note Viewing the.pfx provides the Alias Name required during the import.

Step 2 Enter the keystore password when prompted.

This is the same password entered when creating the.pfx file.

The information that displays (see the following Example) includes the alias_name needed for entering the following command to import the certificates into the cgms_keystore file:

Step 3 Enter the following command to import the certificates into the cgms_keystore file:

```
keytool -importkeystore -v -srckeystore
filename.pfx -srcstoretype pkcs12
-destkeystore cgms_keystore -deststoretype jks -srcalias alias_name
-destalias cgms
-destkeypass
keystore_password
```

Step 4 At the prompt, enter the destination keystore password.

Step 5 Re-enter the keystore password when prompted.

Step 6 Enter the password used when creating the.pfx file (either *nms_cert.pfx* or *tps_cert.pfx*) when prompted for the source keystore password.

Note In this example, **keystore** was the password when we created the *.pfx* file.

Example

To view the *nms_cert.pfx* file and access the Alias name, enter the following commands as root:

Note This example shows the steps for the *nms_cert.pfx*. To view the details on the *tps_cert.pfx* and import the certificates to the TPS proxy, use the same commands but replace the references to *nms_cert.pfx* with *tps_cert.pfx*, and use the Alias name from the *tps_cert.pfx* file.

```
# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
Enter keystore password: keystore
Keystore type: PKCS12
Keystone provider: SunJSSE
Your keystore contains 1 entry
Alias name: le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b
Creation date: March 29.2018
Entry type: PrivateKey Entry
Certificate chain length: 3
Certificate[1]:
Owner: CN=nms.sgbu.cisco.com
Issuer: CN=cisco-RSA-SUBCA-CA, DC=cisco, DC=com
...
```

To import the certificates to the **cgms_keystore** file on IoT FND, enter the following commands as root:

```
# keytool -importkeystore -v
-srckeystore nms_cert.pfx -srcstoretype pkcs12
-destkeystore cgms_keystore -deststoretype jks -srcalias
le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b -destalias cgms
-destkeypass
keystore_password

Enter destination keystore password: keystore_password
Re-enter new password: keystore_password
Enter source keystore password: keystore
...Storing cgms_keystore
```

Note The **storing** *cgms* **_keystore** text indicates successful completion.

Copying the cgms_keystore File to IoT FND

To copy the **cgms_keystore** file into the following IoT FND and TPS proxy directories:

Step 1 For IoT FND, copy the cgms_keystore file to this directory: **/opt/cgms/server/cgms/conf/**

Step 2 For the TPS proxy, copy the cgms_keystore file to this directory: **/opt/cgms-tpsproxy/conf/**

Note For these certificates to be active and enforceable, they must be in the correct directory.

Extracting cgms.pem and cgms.key file from *.pfx file

To extract the cgms.pem and cgms.key file:

Step 1 Enter the following command to import the certificates into the cgms.pem file.

# openssl pkcs12 -in cgms.pfx -nokeys -out cgms.pem

Step 2 Enter the following command to import the certificates into the cgms.key file.

# openssl pkcs12 -in cgms.pfx -nocerts -out cgms.key -nodes

Note Enter the keystore password when prompted. This is the same password entered when creating the.pfx file.

Copying the cgms.pem and cgms.key Files to IoT FND

For IoT FND, copy the cgms.pem and cgms.key file to this directory: **/opt/cgms/server/cgms/conf/**

Importing the CA Certificate

In addition to importing the NMS certificate, you must import the CA or (subCA) certificate to the cgms_keystore.

To import the CA certificate into the cgms_keystore:

Step 1 On the IoT FND application server, log in as root.

Step 2 Change directory to /opt/cgms/server/cgms/conf, where you have placed the cgms_keystore file:

```
# cd /opt/cgms/server/cgms/conf
```

Step 3 Import the CA certificate:

```
# keytool -import -trustcacerts -alias
root -keystore cgms_keystore -file ca_cert.cer
```

A script displays on the screen.

Step 4 Enter the keystore password when prompted.

Step 5 Re-enter the password.

Step 6 Enter **yes** when prompted to trust the certificate.

The certificate is added to the Keystore.

Example

To import the CA certificate, enter the following commands as root:

```
# keytool -import -trustcacerts -alias root -keystore
cgms_keystore -file ca_cert.cer
```

```
Enter keystore password: keystore_password
Owner: CN=SGBUNMSCA-WIN-4BGS4M94L66-CA,DC=SGBUNMSCA,DC=lab,DC=co
Issuer: CN=SGBUNMSCA-WIN-4BGS4M94L66-CA,DC=SGBUNMSCA,DC=lab,DC=co
Serial number:50adbd57e6b136984f9c1512a0eb7174
Valid from: Wed Jan 11 10:58:09 PDT 2018 until: Wed Jan 11:08:59 PDT 2018
Certificate _fingerprints:
MD5: AE:5D:F4:0A:2B:E5:C8:D8:4A:F4:18:56:FD:A7:8D:7D
SHA1: 83:22:12:8C:6A:23:D3:08:2B:00:55:EF:BD:FF:BA:47:97:99:7E:41
Signature algorithm name: SHA1withRSA
Version:3

Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
CA:true
PathLen:2147483647
]
#2:ObjectId: 2.5.29.15 Criticality=false
KeyUsage[
DigitalSignature
Key_CertSign
Crl_Sign
]
#3:ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000:C6 AB 38 CC EE 79 B0 51 3B 4D 13 c8 9A 56 F6 73..8..y.Q;M...V.s
0010:B9 19 FF 7B....
]
]
#4: ObjectId:1.3.6.1.4.1.311.21.1 Criticality=false
Trust this certificate [no] yes
Certificate was added to the keystore.
```

Importing the CA Certificate into TPS Proxy Keystore

Follow the same steps as in Importing the CA Certificate, on page 34 to import the CA certificate into cgms_keystore on the TPS proxy.

Importing the SUDI Certificate

To import the SUDI certificate into cgms_keystore

Step 1 As root, change directory to /opt/cgms/server/cgms/conf, where you have placed the cgms_keystore file.

```
# cd /opt/cgms/server/cgms/conf
```

Step 2 To import the SUDI certificate, enter the following commands.

```
#keytool -import -trustcacerts -alias sudi -keystore cgms_keystore -file ciscosudi/cisco-sudi-ca.pem
```

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

Note It is mandatory to import both the SUDI certificates with 2029 and 2099 expiry periods.

Step 3 Enter the keystore password when prompted and enter yes. The certificate is added to the keystore.

Note For the TPS proxy, copy the cgms_keystore file to this directory:

/opt/cgms-tpsproxy/conf/ciscosudi

Installing Custom Browser Certificates

Default IoT FND installations use a self-signed certificate for HTTP(S) communication using either a client Web browser or the NB API client. If required, you can use certificates signed by your CA servers. This section presents installation procedures for these custom certificates.

This section covers the following topics:

- Installing Custom Certificates in the Browser Client, on page 36
- Importing Custom Certificates with the North Bound API Client (Windows), on page 39
- Importing Custom Certificates with Windows IE, on page 39
- Managing Custom Certificates, on page 42
- Managing North Bound API Events, on page 42

BEFORE YOU BEGIN

- Clear the client browser cache.
- Remove existing certificates for the NMS server (by IP and DNS) on the client browser

  In Firefox for example, select **Preferences** > **Advanced** > **Encryption** > **View Certifications**. Remove the certificates in the list for the respective server.
- Choose a common name to use in the signed certificate.

  This name requires a DNS entry that resolves to the NMS server IP address.
- Generate the new certificates and export them to a.PFX file.

  This file must contain the private keys, public certificate, and CA server certificates.

  See Using Keytool to Create the cgms_keystore File, on page 32 for the procedure to generate the private and public keys for the cgms_keystore file and export them to a.PFX file.

Installing Custom Certificates in the Browser Client

These steps assume that the Java environment has been set to use the Java bundled with FND in `/opt/cgms/jre`.

Step 1 On the NMS server, copy the existing jbossas.keystore, jbossas.keystore.password, vault.keystore, and VAULT.dat files from the `/opt/cgms/server/cgms/conf/` directory to a safe location.

Step 2 Delete the existing jbossas.keystore, jbossas.keystore.password, vault.keystore, and VAULT.dat files from the `/opt/cgms/server/cgms/conf/` directory.

Step 3 Determine the alias in the.PFX file that you plan to import into the new jbossas.keystore file:

```
#keytool -list -v -keystore newcert.pfx -storetype pkcs12
```

Enter the keystore password: keystore_password_when_pfx_file_was_created

```
Keystore type: PKCS12
Keystore provider: SunJSSE

Your keystore contains 1 entry

Alias name: le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0
Creation date: Feb 23, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
…
```

Step 4 Import the new custom certificate, in.pfx file format into a new jbossas.keystore file; and, at the same time change the alias name to **jboss**. Follow the prompts:

```
# keytool -importkeystore -v -srckeystore newcert.pfx -srcstoretype pkcs12
-destkeystore/opt/cgms/server/cgms/conf/jbossas.keystore -deststoretype jks
-srcalias le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0 -destalias
jboss -destkeypass your_keystore_password
Enter destination keystore password: your_keystore_password
Enter source keystore password: keystore_password_when_pfx_file_was_created
[Storing /opt/cgms/server/cgms/conf/jbossas.keystore]
```

Step 5 The keystore password is stored in the `/opt/cgms/server/cgms/conf/VAULT.dat` file.

Perform the following steps to update the keystore password to match the one entered in Step 4 (your_keystore_password).

- IoT FND releases 4.9.x and earlier

- IoT FND releases 4.10.x and later

IoT FND releases 4.9.x and earlier

a) Create a new vault.keystore file.

```
keytool -genseckey -alias vault -storetype jceks -keyalg AES -keysize 128 -storepass
your_keystore_password -keypass your_keystore_password -keystore
/opt/cgms/server/cgms/conf/vault.keystore
```

b) Update the VAULT.dat file with the new password:

```
/opt/cgms/bin/vault.sh -k /opt/cgms/server/cgms/conf/vault.keystore -p your_keystore_password
-e /opt/cgms/server/cgms/conf -i 50 -s 12345678 -v vault -b keystore_pass -a password -x
your_keystore_password
```

Optionally, the iteration and salt values can be modified if desired.

Example Output:

```
=====================================================================
JBoss Vault
JBOSS_HOME:/opt/cgms
JAVA: java
=====================================================================

Dec 20, 2018 3:25:29 PM org.picketbox.plugins.vault.PicketBoxSecurityVault init
INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready
Secured attribute value has been stored in vault.
Please make note of the following:
********************************************
Vault Block:keystore_pass
Attribute Name:password
Configuration should be done as follows:
VAULT::keystore_pass::password::1
********************************************
Vault Configuration in AS7 config file:
********************************************
...
</extensions>
<vault>
<vault-option name=="KEYSTORE_URL"
value="/opt/cgms/server/cgms/conf/vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-
VKsAwH928fwt.3H2qUwOG"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="50"/>
<vault-option name="ENC_FILE_DIR" value="/opt/cgms/server/cgms/conf/"/>
</vault><management>...
********************************************
```

where vault.keystore contains the reference to VAULT.dat and VAULT.dat stores and hides the jboss keystore password. This command creates a new VAULT.dat file that contains the new jboss.keytsore password.

IoT FND releases 4.10.x and later

a) Create a new vault.keystore file:

```
/opt/cgms/jre/bin/keytool -genseckey -alias vault -storetype jceks
-keyalg AES -keysize 256 -storepass keystore -dname "CN=IoTFND, OU=IoT, O=Cisco Systems,
L=San Jose, ST=CA, C=US" -keypass keystore -validity 730 -keystore vault.keystore
```

b) Update the VAULT.dat file with the new password:

```
/opt/cgms/bin/vault.sh --keystore /opt/cgms/server/cgms/conf/vault.keystore
 --keystore-password keystore --alias vault --vault-block keystore_pass
--attribute password --sec-attr keystore --enc-dir /opt/cgms/server/cgms/conf/
 --iteration 50 --salt 12345678 -n
```

Optionally, the iteration and salt values can be modified if desired.

Expected Output:

```
============================================================

  JBoss Vault

  JBOSS_HOME: /opt/cgms

  JAVA: /opt/cgms/jre/bin/java

=========================================================================

WFLYSEC0047: Secured attribute value has been stored in Vault.
Please make note of the following:
********************************************
Vault Block:keystore_pass
Attribute Name:password
Configuration should be done as follows:
VAULT::keystore_pass::password::1
********************************************
WFLYSEC0048: Vault Configuration commands in WildFly for CLI:
********************************************
For standalone mode:
/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"/opt/cgms/server/cgms/conf/vault.keystore"),
("KEYSTORE_PASSWORD" => "MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR" =>
"/opt/cgms/server/cgms/conf/")])
********************************************
For domain mode:
/host=the_host/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"/opt/cgms/server/cgms/conf/vault.keystore"),
("KEYSTORE_PASSWORD" => "MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR" =>
 "/opt/cgms/server/cgms/conf/")])
********************************************
```

where vault.keystore contains the reference to VAULT.dat and VAULT.dat stores and hides the jboss keystore password. This command creates a new VAULT.dat file that contains the new jboss.keytsore password.

Step 6 Copy /opt/cgms/standalone/configuration/standalone.xml and standalone-cluster.xml to a safe location.

Step 7 Update the /opt/cgms/standalone/configuration/standalone.xml or standalone-cluster.xml file.

Depending on whether the FND server is standalone or clustered, update the respective file accordingly.

a) Replace the keystore password with the output in step 5.

```
<vault>
<vault-option name="KEYSTORE_URL"
value="/opt/cgms/server/cgms/conf/vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-VKsAwH928fwt.3H2qUwOG"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="50"/>
<vault-option name="ENC_FILE_DIR" value="/opt/cgms/server/cgms/conf/"/>
</vault>
```

b) Edit the standalone.xml or standalone-cluster.xml and replace the existing <vault> section with the above. Save and exit.

Step 8 Restart IoT FND by running the following command:

RHEL Version	Command
8.x	`systemctl restart cgms`
7.x	`service cgms restart`

Step 9 Use your browser to connect to the NMS server.

Step 10 Accept and add the new certificates.

Step 11 Use your browser to log in to IoT FND.

Importing Custom Certificates with the North Bound API Client (Windows)

For an NB API client running on a Windows Server, import the CA public certificate to the Certificate Store on your local computer. Matching CA public certificates ensures that the client machine communicates with IoT FND using the NB API client.

Importing Custom Certificates with Windows IE

To import custom certificates with windows IE:

Step 1 In IE, enter the https URL address of the NMS server.

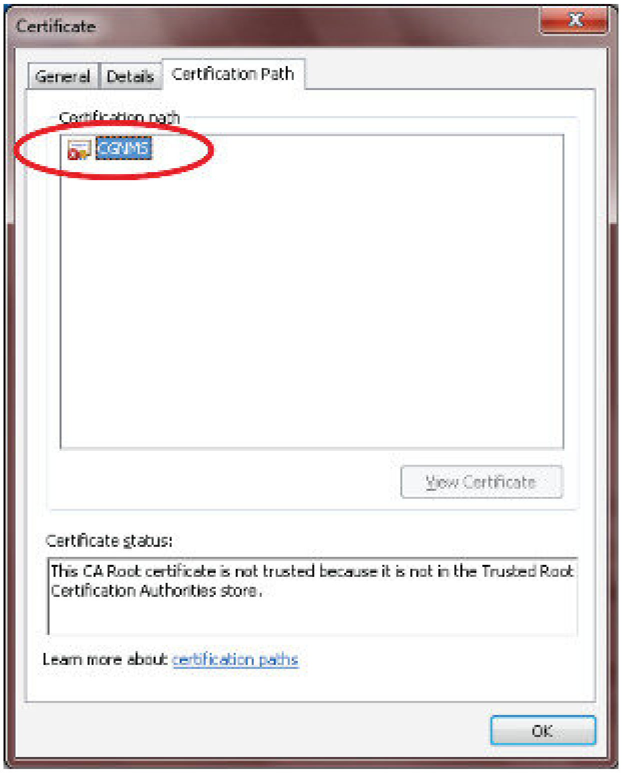
The URL name must match the Common Name on the NMS Server certificate.

Step 2 In the Security Alert window, click **OK**.

Step 3 In the security certificate warning window, click the **Continue to this Website (Not Recommended)** link.

Step 4 In the Security Alert window, click **OK**.

Step 5 Click the **Certificate error** section of the address bar.



Certification path

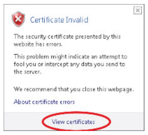
Certificate status:

This CA Root certificate is not trusted because it is not in the Trusted Root Certification Authorities store.

Learn more about certification paths

Step 6 In the Certificate Invalid window, click **View certificates**.

The Certificate window lists the device certificate issued to the NMS server and signed by the issuing CA (or sub CA) server.



Certificate Invalid

The security certificate presented by this website has errors.

This problem might indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage.

About certificate errors

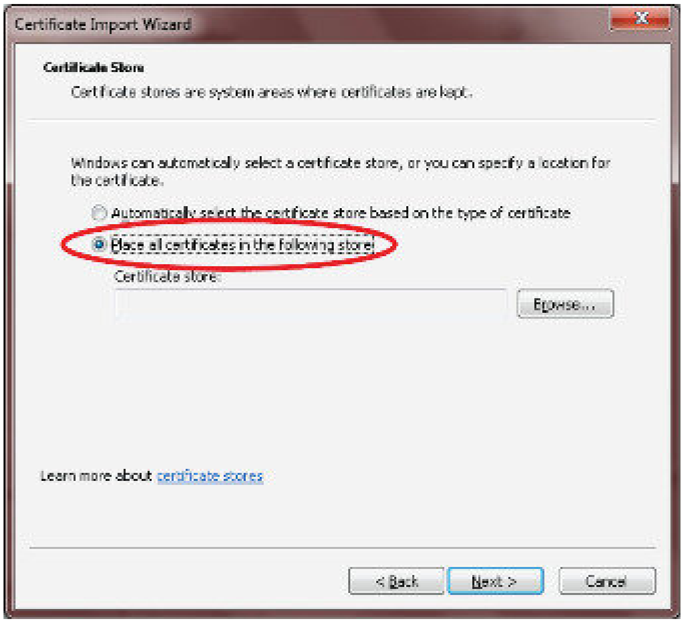
Step 7 Select the **Certification Path** tab, and look for the invalid certificate (that is, the one with a red cross).

Step 8 Select the invalid certificate, and select the **General** tab.

Step 9 Click **Install Certificate**.

Step 10 In the Certificate Install Wizard window, click **Next**.

Step 11 Select the **Place all certificates in the Following Store** option, and then click **Browse**.



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

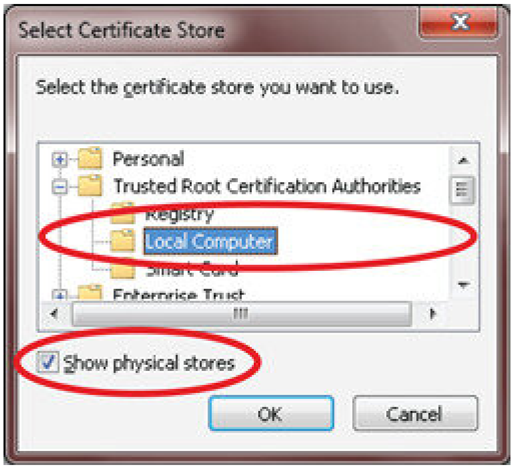
○ Automatically select the certificate store based on the type of certificate

● Place all certificates in the following store

Certificate store:

Learn more about certificate stores

Step 12 In the Certificate Store window, check the **Show physical stores** check box, open the Trusted Root Certification Authorities folder, select **Local Computer**, and then click **OK**.



Select Certificate Store

Select the certificate store you want to use.

- Personal
- Trusted Root Certification Authorities
  - Registry
  - Local Computer
  - Smart Card
- Enterprise Trust

☑ Show physical stores

Step 13 Click **Next**.

Step 14 Click **Finish**.

Step 15 Click **OK**.

Step 16 In the Certificate window, click **Install Certificate**.

Step 17 Select the **Place all certificates in the Following Store** option, and then click **Browse**.

Step 18 In the Certificate Store window, check the **Show physical stores** check box, open the Trusted Root Certification Authorities folder, select **Local Computer**, and then click **OK**.

Step 19 Click **Next**.

Step 20 Click **Finish**.

Step 21 Click **OK**.

Step 22 In the Certificate window, click **OK**.

Step 23 Repeat the previous steps if the Certificate error section of the address bar still appears.

- Ensure that the device certificate issued to the NMS server and signed by the issuing CA (or sub CA) displays server in the Certificate window.
- Select the Certification Path tab and verify that all certificates in the path are valid (that is, there are no red crosses on the certificates).

Step 24 Close and restart the browser.

Step 25 Enter the IoT FND server secure URL in the address bar.

The IoT FND login page displays without the security screen.

Managing Custom Certificates

Step 1 Back up the following files that are overwritten when you upgrade or perform a fresh installation of IoT FND:

a) In the /opt/cgms/server/cgms/conf/ directory:

- jbossas.keystore.password
- jbossas.keystore

b) In the /opt/cgms/server/cgms/deploy/ directory:

- security-service.xml file

  This is the file where you added the salt value in Installing Custom Certificates in the Browser Client, on page 36).

c) In the /opt/cgms/server/cgms/conf directory:

- VAULT.dat
- vault.keystore

Step 2 Perform the IoT FND upgrade or new installation. Refer to the appropriate installation chapter within this guide:

Installing Cisco IoT FND-RPM for the First Time - Oracle Deployment

Step 3 Copy the above files to their respective folders, and restart IoT FND.

Managing North Bound API Events

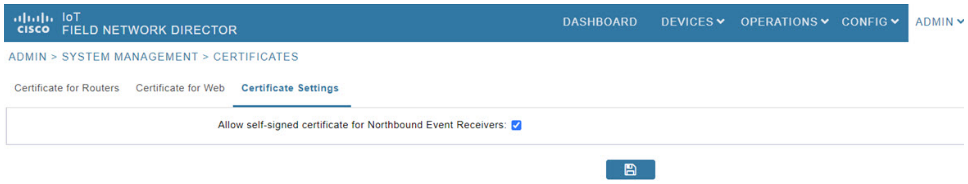
The North Bound (NB) API client can send events using HTTPS. NB API clients must subscribe to IoT FND by providing a valid HTTPS URL over which IoT FND sends events.

The NB API HTTPS communication uses either Self-Signed Certificate or Custom CA Certificate.

Self-Signed Certificate

By default, a new self-signed FND Web certificate is generated in `jbossas.kesytore` everytime when FND is upgraded or installed.

If you are using self-signed certificates, then navigate to **ADMIN** > **SYSTEM MANAGEMENT** > **CERTIFICATES** and check the **Allow self-signed certificate for Northbound Event Receivers** check box.



Allow self-signed certificate for Northbound Event Receivers: ☑

Custom CA Certificate

NB API HTTPS communication with FND can also be done using custom CA certificates. In this case, the custom CA certificate for FND Web installed in `jbossas.keystore` is used.

The `jbossas.keystore` must contain the following entries:

- Issuing CA Certificate:
  - If the FND Web certificate is issued by the sub CA server, then the certificate of the sub CA server is installed in the keystore.
  - If the certificate is issued by the root CA server, then the certificate of the root CA server is installed in the keystore.
- The root CA certificate of the sub-CA server.
- The certificate that is generated for FND Web server by issuing sub CA server or root CA server.

Note Starting from IoT FND 4.4.0 release, the issuer CA certificate also needs to be installed in the `jbossas.keystore` in addition to the Web certificate installed for FND from custom CA.

Configuring IoT FND to Access the Keystore

After you create `cgms_keystore` and import the NMS and CA certificates to it, configure IoT FND to access the `cgms_keystore` file.

To set the keystore password:

Step 1 Stop IoT FND.

Step 2 Run the `setupCgms.sh` script:

```
pwd
/opt/cgms/bin ./setupCgms.sh


06-12-2018 10:21:39 PDT: INFO: ========== CG-NMS Setup Started - 2018-06-12-10-21-39 =====
06-12-2018 10:21:39 PDT: INFO: Log file: /opt/cgms/bin/../server/cgms/log/cgms_setup.log
Are you sure you want to setup CG-NMS (y/n)? y
06-12-2018 10:21:39 PDT: INFO: User response: y
...
Do you want to change the keystore password (y/n)? y
06-12-2018 10:21:52 PDT: INFO: User response: y
Enter keystore password: keystore_password
Re-enter keystore password: keystore_password
06-12-2018 10:21:59 PDT: INFO: Configuring keystore password. This may take a while.
Please wait...
06-12-2018 10:22:00 PDT: INFO: Keystore password configured.
...
```

This script saves the password set in the `cgms.properties` file.

Step 3 Start IoT FND.

Tip To protect the cgms_keystore and cgms.properties files, set their permissions to root read only.

Caution Protect your system! Ensure that only root has access to the IoT FND server. Your firewall should only allow SSH access from internal hosts.

Configuring the TPS Proxy to Access the Keystore

To configure the TPS proxy to access the keystore:

Step 1 Change to the tpsproxy bin directory:

```
cd /opt/cgms-tpsproxy/bin
```

Step 2 Convert your chosen password into encrypted form:

```
[root@fndnms bin]#./encryption_util.sh
Usage: EncryptionUtil <encrypt|decrypt> <data>

[root@fndnms bin]#./encryption_util.sh encrypt test
Kq+bA+oLBlo=
```

Step 3 Copy the encrypted password into the tpsproxy.properties file:

a) Open the file for editing.

```
cd /opt/cgms-tpsproxy/conf
emacs tpsproxy.properties
```

b) Add this line to the file:

```
cgms-keystore-password-hidden=keystore_password
```

In this example, the encrypted *keystore_password* is `“7jlXPniVpMvat+TrDWqh1w==”`

Step 4 Restart the TPS proxy by running the following command:

RHEL Version	Command
8.x	`systemctl restart tpsproxy`

RHEL Version	Command
7.x	`service tpsproxy restart`

Setting Up the HSM Client

Complete the following procedures to set up the HSM client:

- Installing the HSM Client on the IoT FND Server, on page 45
- Configuring the HSM HA Client, on page 51

Note If your installation uses SSM for CSMP-based messaging, see the "Installing Cisco IoT FND RPM for The First Time" chapter.

Installing the HSM Client on the IoT FND Server

The Hardware Security Module (HSM) works as a security server listening at port 1792. For IoT FND to communicate with HSM:

1. Install the HSM client on the IoT FND server.
2. Configure the HSM client to have the certificate for HSM.
3. Upload the certificate to HSM.

This section describes how to install and configure the HSM client, assuming that HSM is at 172.16.0.1 and the client at 172.31.255.254.

To install and set up an HSM client:

Step 1 Get the HSM client package, unpack it, and run the installation script:

```
sh install.sh
```

Step 2 Create the client certificate:

```
cd /usr/safenet/lunaclient/bin/
```

Step 3 Create the client certificate:

```
./vtl createCert -n  ip_address_of_hsm_client
```

Step 4 Download the HSM certificate from the HSM server:

```
scp admin@ip_address_of_hsm_server
:server.pem.
```

Step 5 Upload the client certificate to the HSM server:

```
scp../cert/client/ip_address_of_hsm_client.pem
admin@ip_address_of_hsm_server:.
```

Step 6 Load the HSM certificate:

```
vtl addServer -n
ip_address_of_hsm_server -c server.pem.
```

Step 7 Ensure that the HSM server is added:

```
vtl listServer
```

Step 8 From the HSM client, use SSH to log in to the HSM server:

```
ssh admin@ip_address_of_hsm_server
Last login: Mon Aug 15 15:36:43 2018 from 10.27.164.171
Luna SA 5.0.1-2 Command Line Shell - Copyright (c) 2001-2010 SafeNet, Inc. All rights reserved.
[TestLunaSA1] lunash:>
```

Step 9 Use SSH to perform these steps on the HSM server:

a) Add the client to the HSM server:

```
[TestLunaSA1] lunash:>client register -c hsm_client_name -i ip_address_of_hsm_client
'client register' successful. Command Result : 0 (Success)
```

b) List the clients that are defined on the server and ensure that the client was added:

```
[TestLunaSA1] lunash:>client list
registered client 1: cg-nms
registered client 2: hsm_client_name
Command Result : 0 (Success)
```

c) Assign the client to a partition:

```
[TestLunaSA1] lunash:>client assignPartition -c hsm_client_name
-p partition_name
'client assignPartition' successful.
Command Result : 0 (Success)
```

d) Log out of HSM.

Step 10 On the server running the HSM client, verify the HSM client installation:

```
vtl verify
The following Luna SA Slots/Partitions were found:
Slot Serial # Label
==== ======== =====
1 151285008 TestPart1
```

For a fresh install of IoT FND and HSM integration, the CSMP certificate appears in the IoT FND UI only when an endpoint/meter is added to IoT FND, irrespective of whether the meter/endpoint is registered to IoT FND or not.

Note You can also add a dummy entry for meter/endpoint, if there is no real endpoint or meter to add at the point of testing CSMP certificate display.

Apart from the CSMP certificate displayed in the GUI, you can also use the following methods to verify if IoT FND can access and retrieve the CSMP certificate from HSM:

- **Method 1**

  Run the following command:

  ```
  cat /opt/cgms/server/cgms/log/server.log | grep -i HSM
  ```

  If you get the below message, then IoT FND and HSM communication is successful, and IoT FND can retrieve the public key.

  ```
  %IOTFND-6-UNSPECIFIED: %[ch=HSMKeyStore][sev=INFO]
  [tid=MSC service thread 1-3]:
  Retrieved public key: 3059301306072a8648ce3d020106082a8648ce3d03010703
  420004d914167514ec0a110
  f3170eef742a000572cea6f0285a3074db87e43da398ab016e40ca4be5b888c26c4
  fe91106cbf685a04b0f61d599826bdbcff25cf065d24
  ```

- **Method 2**

  Run the following command. The cmu list command checks if IoT FND can see two objects stored in HSM partition, namely private keys and CSMP certificate.

  ```
  [root@iot-fnd ~]# cd /usr/safenet/lunaclient/bin
  [root@iot-fnd bin]# ./cmu list
  Certificate Management Utility (64-bit) v7.3.0-165.
  Copyright (c) 2018 SafeNet. All rights reserved.

  Please enter password for token in slot 0 : *******

  handle=2000001 label=NMS_SOUTHBOUND_KEY
  handle=2000002 label=NMS_SOUTHBOUND_KEY--cert0
  You have new mail in /var/spool/mail/root
  ```

Step 11 After the HSM client installation completes, run the test suite ckdemo.

```
ckdemo
Ckdemo is the property of SafeNet Inc and is provided to our customers for
diagnostic and development purposes only. It is not intended for use in
production installations. Any re-distribution of this program in whole or
in part is a violation of the license agreement.

CrystokiConnect() (modified on Oct 18 2018 at 20:57:53)

*** CHRYSTOKI DEMO - SIMULATION LAB ***


Status: Doing great, no errors (CKR_OK)

TOKEN FUNCTIONS
( 1) Open Session ( 2) Close Session ( 3) Login
( 4) Logout ( 5) Change PIN ( 6) Init Token
( 7) Init Pin ( 8) Mechanism List ( 9) Mechanism Info
(10) Get Info (11) Slot Info (12) Token Info
(13) Session Info (14) Get Slot List (15) Wait for Slot Event
(18) Factory Reset (19) CloneMofN
```

```
OBJECT MANAGEMENT FUNCTIONS
(20) Create object (21) Copy object (22) Destroy object
(23) Object size (24) Get attribute (25) Set attribute
(26) Find object (27) Display Object
SECURITY FUNCTIONS
(40) Encrypt file (41) Decrypt file (42) Sign
(43) Verify (44) Hash file (45) Simple Generate Key
(46) Digest Key
HIGH AVAILABILITY RECOVERY FUNCTIONS
(50) HA Init (51) HA Login
KEY FUNCTIONS
(60) Wrap key (61) Unwrap key (62) Generate random number
(63) Derive Key (64) PBE Key Gen (65) Create known keys
(66) Seed RNG (67) EC User Defined Curves
CA FUNCTIONS
(70) Set Domain (71) Clone Key (72) Set MofN
(73) Generate MofN (74) Activate MofN (75) Generate Token Keys
(76) Get Token Cert(77) Sign Token Cert(78) Generate CertCo Cert
(79) Modify MofN (86) Dup. MofN Keys (87) Deactivate MofN
(88) Get Token Certificates (112) Set Legacy Cloning Domain
OTHERS
(90) Self Test
(94) Open Access (95) Close Access
(97) Set App ID (98) Options (100) LKM Commands
OFFBOARD KEY STORAGE:
(101) Extract Masked Object (102) Insert Masked Object
(103) Multisign With Value (104) Clone Object
(105) SIMExtract (106) SIMInsert
(107) SimMultiSign (118) Extract Object
(119) Insert Object
SCRIPT EXECUTION:
(108) Execute Script (109) Execute Asynchronous Script
(110) Execute Single Part Script
CLUSTER EXECUTION:
(111) Get Cluster State
SRK FUNCTIONS:
(200) SRK Get State (201) SRK Restore (202) SRK Resplit
(203) SRK Zeroize (204) SRK Enable/Disable

( 0) Quit demo

Enter your choice : 1

Slots available:
slot#1 - LunaNet Slot
slot#2 - Luna UHD Slot
slot#3 - Luna UHD Slot
slot#4 - Luna UHD Slot
Select a slot: 1

SO[0] or normal user[1]?

You must enter a number between 0 and 1: 1

Status: Doing great, no errors (CKR_OK)

TOKEN FUNCTIONS
( 1) Open Session ( 2) Close Session ( 3) Login
( 4) Logout ( 5) Change PIN ( 6) Init Token
( 7) Init Pin ( 8) Mechanism List ( 9) Mechanism Info
(10) Get Info (11) Slot Info (12) Token Info
(13) Session Info (14) Get Slot List (15) Wait for Slot Event
(18) Factory Reset (19) CloneMofN
OBJECT MANAGEMENT FUNCTIONS
```

```
(20) Create object (21) Copy object (22) Destroy object
(23) Object size (24) Get attribute (25) Set attribute
(26) Find object (27) Display Object
SECURITY FUNCTIONS
(40) Encrypt file (41) Decrypt file (42) Sign
(43) Verify (44) Hash file (45) Simple Generate Key
(46) Digest Key
HIGH AVAILABILITY RECOVERY FUNCTIONS
(50) HA Init (51) HA Login
KEY FUNCTIONS
(60) Wrap key (61) Unwrap key (62) Generate random number
(63) Derive Key (64) PBE Key Gen (65) Create known keys
(66) Seed RNG (67) EC User Defined Curves
CA FUNCTIONS
(70) Set Domain (71) Clone Key (72) Set MofN
(73) Generate MofN (74) Activate MofN (75) Generate Token Keys
(76) Get Token Cert(77) Sign Token Cert(78) Generate CertCo Cert
(79) Modify MofN (86) Dup. MofN Keys (87) Deactivate MofN
(88) Get Token Certificates (112) Set Legacy Cloning Domain
OTHERS
(90) Self Test
(94) Open Access (95) Close Access
(97) Set App ID (98) Options (100) LKM Commands
OFFBOARD KEY STORAGE:
(101) Extract Masked Object (102) Insert Masked Object
(103) Multisign With Value (104) Clone Object
(105) SIMExtract (106) SIMInsert
(107) SimMultiSign (118) Extract Object
(119) Insert Object
SCRIPT EXECUTION:
(108) Execute Script (109) Execute Asynchronous Script
(110) Execute Single Part Script
CLUSTER EXECUTION:
(111) Get Cluster State
SRK FUNCTIONS:
(200) SRK Get State (201) SRK Restore (202) SRK Resplit
(203) SRK Zeroize (204) SRK Enable/Disable

( 0) Quit demo

Enter your choice : 3
Security Officer[0]
Crypto-Officer [1]
Crypto-User [2]: 1
Enter PIN : 9JT5-WMYG-E5FE-TExs

Status: Doing great, no errors (CKR_OK)

TOKEN FUNCTIONS
( 1) Open Session ( 2) Close Session ( 3) Login
( 4) Logout ( 5) Change PIN ( 6) Init Token
( 7) Init Pin ( 8) Mechanism List ( 9) Mechanism Info
(10) Get Info (11) Slot Info (12) Token Info
(13) Session Info (14) Get Slot List (15) Wait for Slot Event
(18) Factory Reset (19) CloneMofN
OBJECT MANAGEMENT FUNCTIONS
(20) Create object (21) Copy object (22) Destroy object
(23) Object size (24) Get attribute (25) Set attribute
(26) Find object (27) Display Object
SECURITY FUNCTIONS
(40) Encrypt file (41) Decrypt file (42) Sign
(43) Verify (44) Hash file (45) Simple Generate Key
(46) Digest Key
HIGH AVAILABILITY RECOVERY FUNCTIONS
```

```
(50) HA Init (51) HA Login
KEY FUNCTIONS
(60) Wrap key (61) Unwrap key (62) Generate random number
(63) Derive Key (64) PBE Key Gen (65) Create known keys
(66) Seed RNG (67) EC User Defined Curves
CA FUNCTIONS
(70) Set Domain (71) Clone Key (72) Set MofN
(73) Generate MofN (74) Activate MofN (75) Generate Token Keys
(76) Get Token Cert(77) Sign Token Cert(78) Generate CertCo Cert
(79) Modify MofN (86) Dup. MofN Keys (87) Deactivate MofN
(88) Get Token Certificates (112) Set Legacy Cloning Domain
OTHERS
(90) Self Test
(94) Open Access (95) Close Access
(97) Set App ID (98) Options (100) LKM Commands
OFFBOARD KEY STORAGE:
(101) Extract Masked Object (102) Insert Masked Object
(103) Multisign With Value (104) Clone Object
(105) SIMExtract (106) SIMInsert
(107) SimMultiSign (118) Extract Object
(119) Insert Object
SCRIPT EXECUTION:
(108) Execute Script (109) Execute Asynchronous Script
(110) Execute Single Part Script
CLUSTER EXECUTION:
(111) Get Cluster State
SRK FUNCTIONS:
(200) SRK Get State (201) SRK Restore (202) SRK Resplit
(203) SRK Zeroize (204) SRK Enable/Disable

( 0) Quit demo

Enter your choice : 27

Enter handle of object to display (-1 to list available objects) :

You must enter a number between -1 and 10000000: -1

No objects found

Enter handle of object to display (-1 to list available objects) :

You must enter a number between -1 and 10000000:

You must enter a number between -1 and 10000000:

You must enter a number between -1 and 10000000: 0
ERROR: Can not find object with handle 0

Status: C_GetObjectSize returned error. (CKR_OBJECT_HANDLE_INVALID)

TOKEN FUNCTIONS
( 1) Open Session ( 2) Close Session ( 3) Login
( 4) Logout ( 5) Change PIN ( 6) Init Token
( 7) Init Pin ( 8) Mechanism List ( 9) Mechanism Info
(10) Get Info (11) Slot Info (12) Token Info
(13) Session Info (14) Get Slot List (15) Wait for Slot Event
(18) Factory Reset (19) CloneMofN
OBJECT MANAGEMENT FUNCTIONS
(20) Create object (21) Copy object (22) Destroy object
(23) Object size (24) Get attribute (25) Set attribute
(26) Find object (27) Display Object
SECURITY FUNCTIONS
(40) Encrypt file (41) Decrypt file (42) Sign
```

```
(43) Verify (44) Hash file (45) Simple Generate Key
(46) Digest Key
HIGH AVAILABILITY RECOVERY FUNCTIONS
(50) HA Init (51) HA Login
KEY FUNCTIONS
(60) Wrap key (61) Unwrap key (62) Generate random number
(63) Derive Key (64) PBE Key Gen (65) Create known keys
(66) Seed RNG (67) EC User Defined Curves
CA FUNCTIONS
(70) Set Domain (71) Clone Key (72) Set MofN
(73) Generate MofN (74) Activate MofN (75) Generate Token Keys
(76) Get Token Cert(77) Sign Token Cert(78) Generate CertCo Cert
(79) Modify MofN (86) Dup. MofN Keys (87) Deactivate MofN
(88) Get Token Certificates (112) Set Legacy Cloning Domain
OTHERS
(90) Self Test
(94) Open Access (95) Close Access
(97) Set App ID (98) Options (100) LKM Commands
OFFBOARD KEY STORAGE:
(101) Extract Masked Object (102) Insert Masked Object
(103) Multisign With Value (104) Clone Object
(105) SIMExtract (106) SIMInsert
(107) SimMultiSign (118) Extract Object
(119) Insert Object
SCRIPT EXECUTION:
(108) Execute Script (109) Execute Asynchronous Script
(110) Execute Single Part Script
CLUSTER EXECUTION:
(111) Get Cluster State
SRK FUNCTIONS:
(200) SRK Get State (201) SRK Restore (202) SRK Resplit
(203) SRK Zeroize (204) SRK Enable/Disable

( 0) Quit demo

Enter your choice : 0

Exiting GESC SIMULATION LAB
```

Configuring the HSM HA Client

Note You must perform the steps in this section even if you only have one HSM server. You must also create a group that contains the HSM server.

To configure the HSM HA client:

Step 1 Configure the HSM client so that it connects with both HSM servers, as described in Installing the HSM Client on the IoT FND Server, on page 45.

Step 2 Change to the `/usr/safenet/lunaclient/bin/` directory:

`/usr/safenet/lunaclient/bin/`

Step 3 Create a group that contains only the partition of the first HSM server by running this command and providing the serial number (*serial_num*) of the HSM server obtained by running the **`./vtl verify`** command (Installing the HSM Client on the IoT FND Server, on page 45), the name of the group (*group_name*), and the password (*prtn_password*) for accessing the partition:

```
./vtl haAdmin newGroup -serialNum serial_num
-label group_name -password prtn_password
```

For example:

```
./vtl haAdmin newGroup -serialNum 151285008
-label testGroup1 -password TestPart1

Warning: There are 2 objects currently on the new member.
Do you wish to propagate these objects within the HA
group, or remove them?
Type 'copy' to keep and propagate the existing
objects, 'remove' to remove them before continuing,
or 'quit' to stop adding this new group member.
> copy

New group with label "testGroup1" created at group number 1151285008.
Group configuration is:

HA Group Label: testGroup1
HA Group Number: 1151285008
Synchronization: enabled
Group Members: 151285008
Needs sync: no
```

Step 4 Add the partition of the second HSM to the group.

For example:

```
./vtl haAdmin addMember -group testGroup1
-serialNum 151268008 -password TestPart1
Member 151268008 successfully added to group testGroup1. New group
configuration is:

HA Group Label: testGroup1
HA Group Number: 1151285008
Synchronization: enabled
Group Members: 151285008, 151268008
Needs sync: yes

Please use the command 'vtl haAdmin -synchronize' when
you are ready to replicate data between all members of the
HA group. (If you have additional members to add, you may
wish to wait until you have added them before synchronizing
to save time by avoiding multiple synchronizations.)
```

Step 5 Verify that both partitions can be listed:

```
./vtl haAdmin -listGroups

If you would like to see synchronization data for group testGroup1,
please enter the password for the group members. (Press enter to
skip the synchronization check):
> *********

HA Group Label: testGroup1
HA Group Number: 1151285008
Synchronization: enabled
Group Members: 151285008, 151268008
Needs sync: yes

HA auto recovery: disabled
HA logging: disabled
```

Step 6 Enable HA auto recovery:

```
[root@localhost bin]#./vtl haAdmin -autoRecovery
```

```
vtl haAdmin -autoRecovery [ -retry
<count> | -interval <seconds> ] -retry
<retry count>-interval <seconds>
```

- Set the *retry* value between -1 and 500 where, -1 is an infinite number of retries and 0 disables auto recovery.
- Specify the auto recovery poll *interval* in seconds.

Step 7 Enable HA.

```
./vtl haAdmin -HAOnly -enable
```

Configuring the HSM Group Name and Password

The HSM Group name and password is provided by Cisco at manufacture.

To allow the HSM Group name and password to be configured by the user:

Step 1 Edit the **cgms.properties** file to add the following properties:

a) hsm-keystore-name ***<name>***
b) hsm-keystore-password ***<encrypted password>***

Tip You can use the same HSM server for multiple IoT FND installations by creating multiple partitions on the HSM server, configuring the HSM client, and specifying the partition name and partition password in the cgms.properties file.

Step 2 Save the cgms.properties file.

Step 3 To apply these changes, start the cgms service:

RHEL Version	Command
8.x	`systemctl start cgms`
7.x	`service cgms start`

CHAPTER 4

Installing Cisco IoT FND-RPM for the First Time - Oracle Deployment

This chapter provides an overview of the steps required to install Cisco IoT Field Network Director (Cisco IoT FND) software and the supporting application and database hardware servers within your network for the first time. This guide focuses on installation of Red-hat Package Manager (RPM) deployments for large-scale Advanced Metering Infrastructure (AMI) use cases. These deployment options allow for a la carte deployment of each head-end component with dedicated hardware. With this design, you have the flexibility to horizontally scale the Cisco IoT FND to support up to 11 million endpoints.

Note For an overview of the features and functionality of the application and details on how to configure features and manage the Cisco IoT Field Network Director after its installation, refer to the Cisco IoT Field Network Director User Guides, Releases 4.3.x. 4.4.x, 4.5.x or 4.6.x.

Note Review the Before You Install Field Network Director chapter in this guide and the relevant FND Release Notes before you install Oracle software to ensure you are installing the correct version.

Note Cisco IoT FND Releases 4.6.1 and later support Oracle 19c Enterprise Edition.

Only Cisco IoT FND Release 4.5.x and Cisco IoT FND Release 4.6.x support Oracle18c Enterprise Edition.

Cisco IoT FND Release 4.4.x and Cisco IoT FND Release 4.3.x support Oracle 12c and 11g Enterprise Editions.

IoT FND Installation Overview

Complete the following procedures to install IoT FND for the first time:

- Installing and Setting Up the IoT FND Database
- Installing and Setting Up IoT FND
- Installing and Configuring the IoT FND TPS Proxy
- Backing Up and Restoring the IoT FND Database

Installing and Setting Up the IoT FND Database

Complete the following procedures to finish your IoT FND installation:

- Installation and Setup Overview
- Downloading and Unpacking Oracle Database
- Running the Oracle Database Installer
- Setting Up the IoT FND Database
- Additional IoT FND Database Topics

Installation and Setup of IoT FND Database Overview

The following topics provide an overview of IoT FND database deployment:

- Single-Server Database Deployment, on page 56
- High Availability Database Server Deployment, on page 57

Single-Server Database Deployment

To install and set up IoT FND database for a single-server database deployment:

- Log in to the database server.
- Downloading and Unpacking Oracle Database.
- Running the Oracle Database Installer.
- Setting Up the IoT FND Database.

High Availability Database Server Deployment

To install and set up IoT FND database for HA:

- Log in to the primary IoT FND database server.
- Downloading and Unpacking Oracle Database.
- Running the Oracle Database Installer.
- Log in to the standby database server
- Downloading and Unpacking Oracle Database.
- Running the Oracle Database Installer.

Downloading and Unpacking Oracle Database

To download the Oracle database:

Before you begin

Note Before downloading and installing the Oracle Database, ensure that the /tmp folder can handle, at a minimum, a 5GB file. After you complete the full installation and are working with the FND user interface, you may need to zip up log files sets and the larger /tmp folder will be of value.

Step 1 Log in to your server as root.

Step 2 Download Oracle18c Enterprise Edition Release. (Only FND 4.5.x and FND 4.6.x support this version).

Step 3 To avoid display-related errors when installing the Oracle Database software, as root run this command:

```
# xhost + local:oracle
```

Step 4 Create the **oracle** user and **dba** group:

```
# groupadd dba
# adduser -d /home/oracle -g dba -s /bin/bash oracle
```

Step 5 Unpack the Oracle Database zip archives.

```
p10404530_121020_Linux-x86-64_1of7.zip
p10404530_121020_Linux-x86-64_2of7.zip
p10404530_121020_Linux-x86-64_3of7.zip
p10404530_121020_Linux-x86-64_4of7.zip
p10404530_121020_Linux-x86-64_5of7.zip
p10404530_121020_Linux-x86-64_6of7.zip
p10404530_121020_Linux-x86-64_7of7.zip
```

Running the Oracle Database Installer

To install the Oracle database:

Before you begin

Note Before running the Oracle installer, disable the firewall.

Step 1 Switch to user **oracle** and run the Oracle database installer:

```
# su - oracle
# setenv DISPLAY <desktop>
# path_to_DB_installation_folder/database/runInstaller
```

Step 2 Click **Yes**, and then click **Next**.

Step 3 Click **Install database software only**, and then click **Next**.

Step 4 Click **Single instance database installation**, and then click **Next**.

Step 5 Select **English** as the language in which the database runs, and then click **Next**.

Step 6 Click **Enterprise Edition 6.4GB (Oracle18c)**, and then click **Next**.

Step 7 Select the following two default installation values, Oracle Base and Software Location 12.1.0), and then click **Next**.

- Oracle Base — **/home/oracle/app/oracle**
- Software Location —**/home/oracle/app/oracle/product/12.1.0/dbhome_1**

Later you will create the environment variables ORACLE_BASE and ORACLE_HOME based on the values of the Oracle Base and Software Location properties.

Step 8 On the **Create Inventory** page, keep the default values, and then click **Next**.

- Inventory Directory — **/home/oracle/app/oraInventory**
- oraInventory_Group Name —**dba**

Step 9 On the **Privileged Operating System Groups** page, keep the default values, and then click **Next**.

- Database Administrator (OSDBA) group — **dba**
- Database Operator (OSOPER) group —**dba**
- Database Backup and Recovery (OSBACKUPDBA) group —**dba (18c only)**
- Data Guard administrative (OSDGDBA) group — **dba (18c only)**
- Encryption Key Management administrative (OSKMDBA) group —**dba (18c only)**

Step 10 (optional) On the **Perform Prerequisite Checks** page, install any required software or run supplied scripts

The installer might require the installation of additional software based on your system kernel settings, and may also instruct you to run scripts to configure your system and complete the database installation.

Note If no missing packages are noted or you see the message “This is a prerequisite condition to test whether the package “ksh” is available on the system, check the Ignore All box.

Step 11 After installing any missing packages, click **Fix & Check Again**.

Keep doing this until all requirements are met.

Caution Do not ignore errors on this page. If there are errors during database installation, IoT FND may not function properly.

Step 12 Click **Next**.

Step 13 On the **Summary** page, verify the database settings, and then click Install (18c) to start the installation process.

Step 14 At the prompts, run the supplied configuration scripts.

Because the installer runs as the user *oracle*, it cannot perform certain installation operations that require root privileges. For these operations, you will be prompted to run scripts to complete the installation process. When prompted, open a terminal window and run the scripts as root.

Step 15 If the installation succeeds, click **Close** on the **Finish** page.

Note If performing a new installation of Oracle 18c or upgrading from Oracle 11g, you must install the Oracle 18c. Go to Mandatory Installing 18c Patch Only Supported on FND 4.5.x and 4.6.x, on page 59.

Mandatory Installing 18c Patch Only Supported on FND 4.5.x and 4.6.x

For all new Oracle 18c database installations and all Oracle 11g upgrades, you must install the 18c patch.

To install the patch:

Step 1 Stop IoT FND application if running.

Step 2 Stop Oracle service if running.

Step 3 Run the following commands to verify inventory of installed Oracle software components and patches. No patches are applied at this stage. The following displays at the end: There are no interim patches installed in this Oracle Home.

```
/home/oracle/app/oracle/product/12.1.0/dbhome_1/OPatch/opatch lsinventory -details

Oracle Interim Patch Installer version 12.1.0.1.3
Copyright (c) 2016, Oracle Corporation.  All rights reserved.

Oracle Home       : /home/oracle/app/oracle/product/12.1.0/dbhome_1
Central Inventory : /home/oracle/app/oraInventory   from           :
/home/oracle/app/oracle/product/12.1.0/dbhome_1/oraInst.loc
OPatch version    : 12.1.0.1.3
OUI version       : 12.1.0.2.0
Log file location :
/home/oracle/app/oracle/product/12.1.0/dbhome_1/cfgtoollogs
/opatch/opatch2016-02-25_10-37-50AM_1.log

Lsinventory Output file location : /home/oracle/app/oracle/product/12.1.0/dbhome_1
/cfgtoollogs/opatch/lsinv/lsinventory2016-02-25_10-37-50AM.txt

--------------------------------------------------------------------------------

Installed Top-level Products (1):
Oracle Database 18c                                                   12.1.0.2.0
There are 1 products installed in this Oracle Home.
Installed Products (135):
Assistant Common Files                                                12.1.0.2.0
Buildtools Common Files                                               12.1.0.2.0
Cluster Verification Utility Common Files                             12.1.0.2.0
Database Configuration and Upgrade Assistants                         12.1.0.2.0
Database Migration Assistant for Unicode                             12.1.0.2.0
Database SQL Scripts                                                 12.1.0.2.0
Database Workspace Manager                                           12.1.0.2.0
DB TOOLS Listener                                                    12.1.0.2.0
Deinstallation Tool                                                  12.1.0.2.0
Enterprise Edition Options                                           12.1.0.2.0
Expat libraries                                                       2.0.1.0.2
Generic Connectivity Common Files                                    12.1.0.2.0
Hadoopcore Component                                                 12.1.0.2.0
HAS Common Files                                                     12.1.0.2.0
HAS Files for DB                                                     12.1.0.2.0
Installation Common Files                                            12.1.0.2.0
Installation Plugin Files                                            12.1.0.2.0
Installer SDK Component                                              12.1.0.2.0J
Accelerator (COMPANION)                                             12.1.0.2.0
Java Development Kit                                                 1.6.0.75.0
LDAP Required Support Files                                          12.1.0.2.0
OLAP SQL Scripts                                                     12.1.0.2.0
Oracle Advanced Security                                             12.1.0.2.0
Oracle Application Express                                           12.1.0.2.0
Oracle Bali Share                                                    11.1.1.6.0
Oracle Call Interface (OCI)                                          12.1.0.2.0
Oracle Clusterware RDBMS Files                                       12.1.0.2.0
Oracle Configuration Manager                                         10.3.8.1.1
Oracle Configuration Manager Client                                  10.3.2.1.0
Oracle Configuration Manager Deconfiguration                         10.3.1.0.0
Oracle Containers for Java                                           12.1.0.2.0
Oracle Context Companion                                             12.1.0.2.0
Oracle Core Required Support Files                                   12.1.0.2.0
Oracle Core Required Support Files for Core DB                       12.1.0.2.0
Oracle Core XML Development Kit                                      12.1.0.2.0
Oracle Data Mining RDBMS Files                                       12.1.0.2.0
Oracle Database 18c                                                  12.1.0.2.0
Oracle Database 18c                                                  12.1.0.2.0
Oracle Database 18c Multimedia Files                                 12.1.0.2.0
Oracle Database Deconfiguration                                      12.1.0.2.0
Oracle Database Gateway for ODBC                                     12.1.0.2.0
Oracle Database Plugin for Oracle Virtual Assembly Builder           12.1.0.2.0
Oracle Database User Interface                                       11.0.0.0.0
Oracle Database Utilities                                            12.1.0.2.0
Oracle Database Vault option                                         12.1.0.2.0
Oracle DBCA Deconfiguration                                          12.1.0.2.0
Oracle Extended Windowing Toolkit                                    11.1.1.6.0
Oracle Globalization Support                                         12.1.0.2.0
Oracle Globalization Support                                         12.1.0.2.0
Oracle Globalization Support For Core                                12.1.0.2.0
Oracle Help for Java                                                 11.1.1.7.0
Oracle Help Share Library                                            11.1.1.7.0
Oracle Ice Browser                                                   11.1.1.7.0
Oracle Internet Directory Client                                     12.1.0.2.0
Oracle Java Client                                                   12.1.0.2.0
Oracle Java Layout Engine                                            11.0.0.0.0
Oracle JDBC Server Support Package                                   12.1.0.2.0
Oracle JDBC/OCI Instant Client                                       12.1.0.2.0
Oracle JDBC/THIN Interfaces                                          12.1.0.2.0
Oracle JFC Extended Windowing Toolkit                                11.1.1.6.0
Oracle JVM                                                           12.1.0.2.0
Oracle JVM For Core                                                  12.1.0.2.0
Oracle Label Security                                                12.1.0.2.0
Oracle LDAP administration                                           12.1.0.2.0
Oracle Locale Builder                                                12.1.0.2.0
Oracle Message Gateway Common Files                                  12.1.0.2.0
Oracle Multimedia                                                    12.1.0.2.0
Oracle Multimedia Client Option                                      12.1.0.2.0
Oracle Multimedia Java Advanced Imaging                              12.1.0.2.0
Oracle Multimedia Locator                                            12.1.0.2.0
Oracle Multimedia Locator Java Required Support Files                12.1.0.2.0
Oracle Multimedia Locator RDBMS Files                                12.1.0.2.0
Oracle Net                                                           12.1.0.2.0
Oracle Net Java Required Support Files                               12.1.0.2.0
Oracle Net Listener                                                  12.1.0.2.0
Oracle Net Required Support Files                                    12.1.0.2.0
Oracle Net Services                                                  12.1.0.2.0
Oracle Netca Client                                                  12.1.0.2.0
Oracle Notification Service                                          12.1.0.2.0
Oracle Notification Service (eONS)                                   12.1.0.2.0
Oracle Notification Service for Instant Client                       12.1.0.2.0
Oracle ODBC Driver                                                   12.1.0.2.0
Oracle ODBC Driverfor Instant Client                                 12.1.0.2.0
Oracle OLAP                                                          12.1.0.2.0
Oracle OLAP API                                                      12.1.0.2.0
Oracle OLAP RDBMS Files                                              12.1.0.2.0
Oracle One-Off Patch Installer                                       12.1.0.1.2
Oracle Partitioning                                                  12.1.0.2.0
Oracle Programmer                                                    12.1.0.2.0
Oracle Quality of Service Management (Client)                        12.1.0.2.0
Oracle R Enterprise Server Files                                     12.1.0.2.0
Oracle RAC Deconfiguration                                           12.1.0.2.0
Oracle RAC Required Support Files-HAS                                12.1.0.2.0
Oracle Real Application Testing                                      12.1.0.2.0
Oracle Recovery Manager                                              12.1.0.2.0
Oracle Security Developer Tools                                      12.1.0.2.0
Oracle Spatial and Graph                                             12.1.0.2.0
Oracle SQL Developer                                                 12.1.0.2.0
Oracle Starter Database                                              12.1.0.2.0
Oracle Text                                                          12.1.0.2.0
Oracle Text ATG Language Support Files                               12.1.0.2.0
Oracle Text for Core                                                 12.1.0.2.0
Oracle Text Required Support Files                                   12.1.0.2.0
Oracle Universal Connection Pool                                     12.1.0.2.0
Oracle Universal Installer                                           12.1.0.2.0
Oracle USM Deconfiguration                                           12.1.0.2.0
Oracle Wallet Manager                                                12.1.0.2.0
Oracle XML Development Kit                                           12.1.0.2.0
Oracle XML Query                                                     12.1.0.2.0
oracle.swd.oui.core.min                                     12.1.0.2.0
Parser Generator Required Support Files                              12.1.0.2.0
Perl Interpreter                                                     5.14.1.0.0
Perl Modules 5.14.1.0.0
PL/SQL                                                               12.1.0.2.0
PL/SQL Embedded Gateway                                              12.1.0.2.0
Platform Required Support Files                                      12.1.0.2.0
Precompiler Common Files                                             12.1.0.2.0
Precompiler Common Files for Core                                    12.1.0.2.0
Precompiler Required Support Files 12.1.0.2.0
Precompilers                                                         12.1.0.2.0
RDBMS Required Support Files                                         12.1.0.2.0
RDBMS Required Support Files for Instant Client                      12.1.0.2.0
RDBMS Required Support Files Runtime                                 12.1.0.2.0
Required Support Files                                               12.1.0.2.0
Sample Schema Data                                                   12.1.0.2.0
Secure Socket Layer  12.1.0.2.0
SQL*Plus                                                             12.1.0.2.0
SQL*Plus Files for Instant Client                                    12.1.0.2.0
SQL*Plus Required Support Files                                      12.1.0.2.0
SQLJ Runtime                                                         12.1.0.2.0
SSL Required Support Files for InstantClient                         12.1.0.2.0
Tracle File Analyzer                                                 12.1.0.2.0
```

```
XDK Required Support Files                                           12.1.0.2.0
XML Parser for Java                                                  12.1.0.2.0
XML Parser for Oracle JVM                                            12.1.0.2.0
There are 135 products installed in this Oracle Home.

There are no Interim patches installed in this Oracle Home.
```

Step 4 Apply the patch.

a) On the database machine. Copy the patch file : “p20830993_121020_Linux-x86-64.zip”

b) Run a prerequisite check. It should pass.

```
$ cd /home/oracle/patches/20830993/
$ /home/oracle/app/oracle/product/12.1.0/dbhome_1/OPatch
/opatch prereq CheckConflictAgainstOHWithDetail -ph./
Oracle Interim Patch Installer version 12.1.0.1.3
Copyright (c) 2016, Oracle Corporation.  All rights reserved.

PREREQ session

Oracle Home       : /home/oracle/app/oracle/product/12.1.0/dbhome_1
Central Inventory : /home/oracle/app/oraInventory
   from           : /home/oracle/app/oracle/product/12.1.0/dbhome_1/oraInst.loc
OPatch version    : 12.1.0.1.3
OUI version       : 12.1.0.2.0
Log file location :/home/oracle/app/oracle/product/12.1.0/dbhome_1
/cfgtool logs/opatch/opatch2016-02-25_10-48-48AM_1.log

Invoking prereq "checkconflictagainstohwithdetail"

Prereq "checkConflictAgainstOHWithDetail" passed.

OPatch succeeded.
```

c) Apply the patch.

```
$ /home/oracle/app/oracle/product/12.1.0/dbhome_1
/OPatch/opatch apply
Oracle Interim Patch Installer version 12.1.0.1.3
Copyright (c) 2016, Oracle Corporation.  All rights reserved.

Oracle Home       : /home/oracle/app/oracle/product/12.1.0/dbhome_1
Central Inventory : /home/oracle/app/oraInventory   from           :
/home/oracle/app/oracle/product/12.1.0/dbhome_1/oraInst.loc
OPatch version    : 12.1.0.1.3
OUI version       : 12.1.0.2.0
Log file location: /home/oracle/app/oracle/product/12.1.0/dbhome_1
/cfgtoollogs/opatch/20830993_Feb_25_2016_10_53_25/apply2016-02-25_10-53-25AM_1.log

Applying interim patch '20830993' to OH '/home/oracle/app/oracle/product/12.1.0/dbhome_1'
Verifying environment and performing prerequisite checks...
All checks passed.

Please shutdown Oracle instances running out of this ORACLE_HOME on the local system.
(Oracle Home = '/home/oracle/app/oracle/product/12.1.0/dbhome_1')

Is the local system ready for patching? [y|n]
y
User Responded with: Y
Backing up files...

Patching component oracle.rdbms, 12.1.0.2.0...

Verifying the update...
Patch 20830993 successfully applied
```

```
Log file location:/home/oracle/app/oracle/product/12.1.0/dbhome_1/cfgtoollogs
/opatch/ 20830993_Feb_25_2016_10_53_25/apply2016-02-25_10-53-25AM_1.log

OPatch succeeded.
```

d) Run Opatch utility to verify that the patch is now recognized. Notice the mention of "Interim Patch" at the end of following output.

```
$ /home/oracle/app/oracle/product/12.1.0/dbhome_1/OPatch/opatch
lsinventory -details
Oracle Interim Patch Installer version 12.1.0.1.3
Copyright (c) 2016, Oracle Corporation.  All rights reserved.

Oracle Home       : /home/oracle/app/oracle/product/12.1.0/dbhome_1
Central Inventory : /home/oracle/app/oraInventory
   from           : /home/oracle/app/oracle/product/12.1.0/dbhome_1/oraInst.loc
OPatch version    : 12.1.0.1.3
OUI version       : 12.1.0.2.0
Log file location : /home/oracle/app/oracle/product/12.1.0/dbhome_1
/cfgtoollogs/opatch/opatch2016-02-25_11-05-19AM_1.log

Lsinventory Output file location : /home/oracle/app/oracle/product/12.1.0/dbhome_1
/cfgtoollogs/opatch/lsinv/lsinventory2016-02-25_11-05-19AM.txt

-------------------------------------------------------------------------
-------

Installed Top-level Products (1):
Oracle Database 18c                                                  12.1.0.2.0
There are 1 products installed in this Oracle Home.

Installed Products (135):

Assistant Common Files                                               12.1.0.2.0
Buildtools Common Files                                              12.1.0.2.0
Cluster Verification Utility Common Files                            12.1.0.2.0
Database Configuration and Upgrade Assistants                        12.1.0.2.0
Database Migration Assistant for Unicode                             12.1.0.2.0
Database SQL Scripts                                                 12.1.0.2.0
Database Workspace Manager                                           12.1.0.2.0
DB TOOLS Listener                                                    12.1.0.2.0
Deinstallation Tool                                                  12.1.0.2.0
Enterprise Edition Options                                           12.1.0.2.0
Expat libraries                                                       2.0.1.0.2
Generic Connectivity Common Files                                    12.1.0.2.0
Hadoopcore Component                                                 12.1.0.2.0
HAS Common Files                                                     12.1.0.2.0
HAS Files for DB                                                     12.1.0.2.0
Installation Common Files                                            12.1.0.2.0
Installation Plugin Files                                            12.1.0.2.0
Installer SDK Component                                              12.1.0.2.0
JAccelerator (COMPANION)                                             12.1.0.2.0
Java Development Kit                                                 1.6.0.75.0
LDAP Required Support Files                                          12.1.0.2.0
LAP SQL Scripts                                                     12.1.0.2.0
Oracle Advanced Security                                             12.1.0.2.0
Oracle Application Express                                           12.1.0.2.0
Oracle Bali Share                                                    11.1.1.6.0
Oracle Call Interface (OCI)                                          12.1.0.2.0
Oracle Clusterware RDBMS Files                                       12.1.0.2.0
Oracle Configuration Manager                                         10.3.8.1.1
Oracle Configuration Manager Client                                  10.3.2.1.0
Oracle Configuration Manager Deconfiguration                         10.3.1.0.0
Oracle Containers for Java                                           12.1.0.2.0
Oracle Context Companion                                             12.1.0.2.0
Oracle Core Required Support Files                                   12.1.0.2.0
Oracle Core Required Support Files for Core DB                       12.1.0.2.0
Oracle Core XML Development Kit                                      12.1.0.2.0
Oracle Data Mining RDBMS Files                                       12.1.0.2.0
Oracle Database 18c                                                  12.1.0.2.0
Oracle Database 18c                                                  12.1.0.2.0
Oracle Database 18c Multimedia Files                                 12.1.0.2.0
Oracle Database Deconfiguration                                      12.1.0.2.0
Oracle Database Gateway for ODBC                                     12.1.0.2.0
Oracle Database Plugin for Oracle Virtual Assembly Builder           12.1.0.2.0
Oracle Database User Interface                                       11.0.0.0.0
Oracle Database Utilities                                            12.1.0.2.0
Oracle Database Vault option                                         12.1.0.2.0
Oracle DBCA Deconfiguration                                          12.1.0.2.0
Oracle Extended Windowing Toolkit                                    11.1.1.6.0
Oracle Globalization Support                                         12.1.0.2.0
Oracle Globalization Support                                         12.1.0.2.0
Oracle Globalization Support For Core                                12.1.0.2.0
Oracle Help for Java                                                 11.1.1.7.0
Oracle Help Share Library                                            11.1.1.7.0
Oracle Ice Browser                                                   11.1.1.7.0
Oracle Internet Directory Client                                     12.1.0.2.0
Oracle Java Client                                                   12.1.0.2.0
Oracle Java Layout Engine                                            11.0.0.0.0
Oracle JDBC Server Support Package                                   12.1.0.2.0
Oracle JDBC/OCI Instant Client                                       12.1.0.2.0
Oracle JDBC/THIN Interfaces                                          12.1.0.2.0
Oracle JFC Extended Windowing Toolkit                                11.1.1.6.0
Oracle JVM                                                           12.1.0.2.0
Oracle JVM For Core                                                  12.1.0.2.0
Oracle Label Security                                                12.1.0.2.0
Oracle LDAP administration                                           12.1.0.2.0
Oracle Locale Builder                                                12.1.0.2.0
Oracle Message Gateway Common Files                                  12.1.0.2.0
Oracle Multimedia                                                    12.1.0.2.0
Oracle Multimedia Client Option                                      12.1.0.2.0
Oracle Multimedia Java Advanced Imaging                              12.1.0.2.0
Oracle Multimedia Locator                                            12.1.0.2.0
Oracle Multimedia Locator Java Required Support Files                12.1.0.2.0
Oracle Multimedia Locator RDBMS Files                                12.1.0.2.0
Oracle Net                                                           12.1.0.2.0
Oracle Net Java Required Support Files                               12.1.0.2.0
Oracle Net Listener                                                  12.1.0.2.0
Oracle Net Required Support Files                                    12.1.0.2.0
Oracle Net Services                                                  12.1.0.2.0
Oracle Netca Client                                                  12.1.0.2.0
Oracle Notification Service                                          12.1.0.2.0
Oracle Notification Service (eONS)                                   12.1.0.2.0
Oracle Notification Service for Instant Client                       12.1.0.2.0
Oracle ODBC Driver                                                   12.1.0.2.0
Oracle ODBC Driverfor Instant Client                                 12.1.0.2.0
Oracle OLAP                                                          12.1.0.2.0
Oracle OLAP API                                                      12.1.0.2.0
Oracle OLAP RDBMS Files                                              12.1.0.2.0
Oracle One-Off Patch Installer                                       12.1.0.1.2
Oracle Partitioning                                                  12.1.0.2.0
Oracle Programmer                                                    12.1.0.2.0
Oracle Quality of Service Management (Client)                        12.1.0.2.0
Oracle R Enterprise Server Files                                     12.1.0.2.0
Oracle RAC Deconfiguration                                           12.1.0.2.0
Oracle RAC Required Support Files-HAS                                12.1.0.2.0
Oracle Real Application Testing                                      12.1.0.2.0
Oracle Recovery Manager                                              12.1.0.2.0
Oracle Security Developer Tools                                      12.1.0.2.0
Oracle Spatial and Graph                                             12.1.0.2.0
Oracle SQL Developer                                                 12.1.0.2.0
Oracle Starter Database                                              12.1.0.2.0
Oracle Text                                                          12.1.0.2.0
Oracle Text ATG Language Support Files                               12.1.0.2.0
Oracle Text for Core                                                 12.1.0.2.0
Oracle Text Required Support Files                                   12.1.0.2.0
Oracle Universal Connection Pool                                     12.1.0.2.0
Oracle Universal Installer                                           12.1.0.2.0
Oracle USM Deconfiguration                                           12.1.0.2.0
Oracle Wallet Manager                                                12.1.0.2.0
Oracle XML Development Kit                                           12.1.0.2.0
Oracle XML Query                                                     12.1.0.2.0
oracle.swd.oui.core.min                                              12.1.0.2.0
Parser Generator Required Support Files                              12.1.0.2.0
Perl Interpreter                                                     5.14.1.0.0
Perl Modules                                                         5.14.1.0.0
PL/SQL                                                               12.1.0.2.0
PL/SQL Embedded Gateway                                              12.1.0.2.0
Platform Required Support Files                                      12.1.0.2.0
Precompiler Common Files                                             12.1.0.2.0
Precompiler Common Files for Core                                    12.1.0.2.0
Precompiler Required Support Files                                   12.1.0.2.0
Precompilers                                                         12.1.0.2.0
RDBMS Required Support Files                                         12.1.0.2.0
RDBMS Required Support Files for Instant Client                      12.1.0.2.0
RDBMS Required Support Files Runtime                                 12.1.0.2.0
Required Support Files                                               12.1.0.2.0
Sample Schema Data                                                   12.1.0.2.0
Secure Socket Layer                                                 12.1.0.2.0
SQL*Plus                                                             12.1.0.2.0
SQL*Plus Files for Instant Client                                    12.1.0.2.0
SQL*Plus Required Support Files                                      12.1.0.2.0
SQLJ Runtime                                                         12.1.0.2.0
SSL Required Support Files for InstantClient                         12.1.0.2.0
Tracle File Analyzer                                                 12.1.0.2.0
XDK Required Support Files                                           12.1.0.2.0
XML Parser for Java                                                  12.1.0.2.0
XML Parser for Oracle JVM                                            12.1.0.2.0
There are 135 products installed in this Oracle Home.

Interim patches (1) :

Patch  20830993     : applied on Thu Feb 25 10:53:50 PST 2016
Unique Patch ID:  18912657
Created on 13 May 2015, 00:37:38 hrs PST8PDT
   Bugs fixed:     20830993
Files Touched:
/qksvc.o --> ORACLE_HOME/lib/libserver12.a
 ins_rdbms.mk --> ORACLE_HOME/rdbms/lib/ioracle
Patch Location in Inventory:
/home/oracle/app/oracle/product/12.1.0/dbhome_1/inventory/oneoffs/20830993
Patch Location in Storage area:
/home/oracle/app/oracle/product/12.1.0/dbhome_1/.patch_storage/20830993_May_13_2015_00_37_38
------------------------------------------------------------------------
--------
Process complete.
```

Continue to Setting Up the IoT FND Database, on page 66

Setting Up the IoT FND Database

Complete the following procedures to set up the IoT FND database:

- IoT FND Database Setup Overview
- Defining Oracle Database Environment Variables
- Installing IoT FND Oracle Database Scripts
- Creating the IoT FND Oracle Database
- Starting the IoT FND Oracle Database

IoT FND Database Setup Overview

To set up the IoT FND database:

1. Defining Oracle Database Environment Variables
2. Installing IoT FND Oracle Database Scripts
3. Creating the IoT FND Oracle Database
4. Starting the IoT FND Oracle Database

Defining Oracle Database Environment Variables

Before installing the IoT FND Oracle database, switch to the **oracle** user account and define the following Oracle database environment variables.

Variable	Description
ORACLE_BASE	Defines the path to the Oracle root directory on your system. For example: `$ export ORACLE_BASE=/home/oracle/app/oracle` If this variable is not set, the IoT FND setup script displays an error.
ORACLE_HOME	Defines the path to the Oracle home of the IoT FND database. For example: `$ export ORACLE_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome_1` **Note** Do not have any trailing backslashes in the ORACLE_HOME environment variable.
PATH	Defines the path to the Oracle binaries. For example: `$ export PATH=$PATH:$ORACLE_HOME/bin`
LD_LIBRARY_PATH	Defines the path to the libraries. For example: `$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH`

Variable	Description

ORACLE_SID	Defines the Oracle System ID (SID).
	If you are only using one database server or installing an HA deployment, set this variable on the *primary* database server to **cgms**:
	`$ export ORACLE_SID=cgms`
	If deploying a standby database server, set this variable on the *standby* database server to **cgms_s**:
	`$ export ORACLE_SID=cgms_s`
	If this variable is not set, the IoT FND setup script displays an error.

You can set these variables manually, as shown in the following example:

On a Single or Primary Database Server	On a Standby Database Server

`$ su - oracle`
`$ export ORACLE_BASE=/home/oracle/app/oracle`
`$ export ORACLE_HOME=/home/oracle/app/oracle/`
`product/12.1.0/dbhome_1`
`$ export PATH=$PATH:$ORACLE_HOME/bin`
`$ export LD_LIBRARY_PATH=$ORACLE_HOME/`
`lib:$LD_LIBRARY_PATH`
`$ export ORACLE_SID=cgms`
	`$ su - oracle`
	`$ export ORACLE_BASE=/home/oracle/app/oracle`
	`$ export ORACLE_HOME=/home/oracle/app/oracle/`
	`product/12.1.0/dbhome_1`
	`$ export PATH=$PATH:$ORACLE_HOME/bin`
	`$ export LD_LIBRARY_PATH=$ORACLE_HOME/`
	`lib:$LD_LIBRARY_PATH`
	`$ export ORACLE_SID=cgms_s`

Installing IoT FND Oracle Database Scripts

IoT FND is packaged with scripts and Oracle database templates.

To install the Oracle scripts on your Oracle server:

Step 1 Log in as the root user.

Step 2 Securely copy the IoT FND Oracle script RPM to your Oracle server:

```
$ scp cgms-oracle-version_number
.x86_64.rpm root@oracle-machine:~
$ rpm -ivh cgms-oracle-version_number
.x86_64.rpm
```

Step 3 Run the script installCgmsOracleScripts.sh from the path /opt/cgms-oracle/scripts/

Creating the IoT FND Oracle Database

To create the IoT FND Oracle database in a single-database-server deployment, run the setupCgmsDb.sh script as the user *oracle*. This script starts the Oracle Database and creates the IoT FND database.

This script creates the user **cgms_dev** used by IoT FND to access the database. The default password for this user account is **cgms123**. The default password for the sys DBA account is **cgmsDba123**.

Note We strongly recommend that you change all default passwords. Do not use special characters such as, @, #, !, or + when using the encryption_util.sh script. The script cannot encrypt special characters.

Note If the DB server is used by other applications in addition to IOT FND and cgms DB is also created on that DB server, then use Oracle provided tools like RMAN for DB management tasks.

Note This script might run for several minutes. To check the setup progress, run the command:*$ tail -f /tmp/cgmsdb_setup.log*

```
$ su - oracle
$ export DISPLAY=localhost:0
$ cd $ORACLE_BASE/cgms/scripts
$ ./setupCgmsDb.sh
09-13-2018 10:38:07 PDT: INFO: ======== CGMS Database Setup Started ==========
09-13-2018 10:38:07 PDT: INFO: Log file: /tmp/cgmsdb_setup.log

Are you sure you want to setup CG-NMS database (y/n)? y

09-13-2018 10:38:08 PDT: INFO: User response: y
09-13-2018 10:38:08 PDT: INFO: CGMS database does not exist.
Enter new password for SYS DBA:
Re-enter new password for SYS DBA:
09-13-2018 10:38:14 PDT: INFO: User entered SYS DBA password.

Enter new password for CG-NMS database:
Re-enter new password CG-NMS database:
09-13-2018 10:38:18 PDT: INFO: User entered CG-NMS DB password.
09-13-2018 10:38:18 PDT: INFO: Stopping listener...
09-13-2018 10:38:18 PDT: INFO: Listener already stopped.
09-13-2018 10:38:18 PDT: INFO: Deleting database files...
09-13-2018 10:38:18 PDT: INFO: Creating listener...
09-13-2018 10:38:19 PDT: INFO: Listener creation completed successfully.
09-13-2018 10:38:19 PDT: INFO: Configuring listener...
09-13-2018 10:38:19 PDT: INFO: Listener successfully configured.
09-13-2018 10:38:19 PDT: INFO: Creating database. This may take a while. Please be patient...
09-13-2018 10:42:55 PDT: INFO: Database creation completed successfully.
09-13-2018 10:42:55 PDT: INFO: Updating /etc/oratab...
09-13-2018 10:42:55 PDT: INFO: /etc/oratab updated.
09-13-2018 10:42:55 PDT: INFO: Configuring database...
09-13-2018 10:42:56 PDT: INFO: Starting listener...
09-13-2018 10:42:56 PDT: INFO: Listener start completed successfully.
09-13-2018 10:42:56 PDT: INFO: Starting database configuration...
09-13-2018 10:43:17 PDT: INFO: Database configuration completed successfully.
09-13-2018 10:43:17 PDT: INFO: Starting Oracle...
09-13-2018 10:43:17 PDT: INFO: Starting Oracle in mount state...
ORACLE instance started.

Total System Global Area 1.6836E+10 bytes
Fixed Size 2220032 bytes
Variable Size 8589934592 bytes
Database Buffers 8187281408 bytes
Redo Buffers 56487936 bytes
Database mounted.
```

```
09-13-2018 10:43:26 PDT: INFO: Opening database for read/write...

Database altered.

09-13-2018 10:43:29
PDT: INFO: ========== CGMS Database Setup Completed Successfully ==========
```

Starting the IoT FND Oracle Database

To start the IoT FND Oracle database:

Step 1 Run the script:

```
$ su - oracle
$ cd $ORACLE_BASE/cgms/scripts
$ ./startOracle.sh
```

Step 2 Configure a cron job that starts IoT FND database at bootup by running this script:

```
./installOracleJob.sh
```

Additional IoT FND Database Topics

The following procedures discuss database management:

- Stopping the IoT FND Oracle Database
- Removing the IoT FND Database
- Upgrading the IoT FND Database
- Changing the SYS DBA and IoT FND Database Passwords
- IoT FND Database Helper Scripts

Stopping the IoT FND Oracle Database

Typically, you do not have to stop the Oracle database during the installation procedure. However, if it becomes necessary to stop the Oracle database, use the stop script in the scripts directory:

```
su - oracle
cd $ORACLE_BASE/cgms/scripts
./stopOracle.sh
...
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
...
```

Removing the IoT FND Database

Caution The following script is destructive. Do not use this script during normal operation.

To remove the IoT FND database, run this script:

```
cd $ORACLE_BASE/cgms/scripts
./deleteCgmsDb.sh
```

Upgrading the IoT FND Database

To upgrade the IoT FND database:

Step 1 Add the database files (a total of 15 files).

```
ALTER TABLESPACE USERS ADD DATAFILE
'&oracle_base/oradata/&sid_caps/users<02 to 15>.dbf'
SIZE 5M AUTOEXTEND ON;
```

This is required for scaling the system.

Step 2 Enable block-change tracking (required for incremental backup):

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE
'&oracle_base/oradata/&sid_caps/rman_change_track.f' REUSE;
```

Step 3 Disable parallel execution:

```
set parallel_max_servers = 0 scope=both
```

Caution The incremental IoT FND backup script enables the Oracle block-change tracking feature to improve backup performance. To take advantage of this feature, delete your IoT FND database and run the setupCgmsDb.sh script before performing the first incremental backup. To avoid losing data, run these commands:

```
sqlplus sys/password@cgms as sysdba
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE
'/home/oracle/app/oracle/oradata/CGMS/rman_change_track.f' REUSE;
exit;
```

Changing the SYS DBA and IoT FND Database Passwords

To change default IoT FND database password for the CGMS_DEV user:

Step 1 On the IoT FND server, run the setupCgms.sh script and change the password for the CGMS_DEV account.

Caution The password for the IoT FND database and the cgnms_dba user password must match or IoT FND cannot access the database.

```
# cd /opt/cgms/bin
#./setupCgms.sh
...
Do you want to change the database password (y/n)? y
09-13-2018 17:15:07 PDT: INFO: User response: y
Enter database password:
Re-enter database password:
09-13-2018 17:15:31 PDT: INFO: Configuring database password.
This may take a while. Please wait...
09-13-2018 17:15:34 PDT: INFO: Database password configured.
...
```

For information about running the setupCgms.sh script, see Setting Up IoT FND , on page 79

Step 2 On the Oracle server, run the change_password.sh script and change the password for the CGMS_DEV account:

```
$ ./change_password.sh
09-13-2018 10:48:32 PDT: INFO: ======== Database Password Util Started ==========
09-13-2018 10:48:32 PDT: INFO: Log file: /tmp/cgms_oracle.log

Are you sure you want to change CG-NMS database password (y/n)? y
09-13-2018 10:48:33 PDT: INFO: User response: y

Enter current password for SYS DBA:
Re-enter current password for SYS DBA:
09-13-2018 10:48:41 PDT: INFO: User entered current SYS DBA password.
Enter new password for SYS DBA:
Re-enter new password for SYS DBA:
09-13-2018 10:48:54 PDT: INFO: User entered SYS DBA password.

Enter new password for CG-NMS database:
Re-enter new password CG-NMS database:
09-13-2018 10:49:03 PDT: INFO: User entered CG-NMS DB password.
User altered.
...
```

Note As root, you can also use this script to change the password for the sys user (SYS DBA).

Step 3 On the IoT FND server, run the cgms_status.sh script to verify the connection between IoT FND and the IoT FND database:

RHEL Version	Command
8.x	`systemctl status cgms`
7.x	`service cgms status`

Example Output:

```
# service cgms status
09-06-2018 18:51:20 PDT: INFO: CG-NMS database server: localhost
09-06-2018 18:51:21 PDT: INFO: CG-NMS database connection verified.
```

IoT FND Database Helper Scripts

Table 8: IoT FND Database Helper Scripts describes helper IoT FND database scripts available in the $ORACLE_BASE/cgms/scripts/ directory.

Note Cisco provides helpful scripts to enable you to perform few tasks easily as FND application talks to DB and interacts with the DB server. However, DB administration tasks are NOT responsibility of Cisco and has to be taken care by your DB administrator.

Table 8: IoT FND Database Helper Scripts

Script	Description
change_password.sh	Use this script to change the passwords for the database administration and IoT FND database user accounts. The IoT FND database user account is used by IoT FND to access the database.
backup_archive_log.sh	Use this script to back up the archive logs.
backupCgmsDb.sh	Use this script to back up the IoT FND database. This script supports full and incremental backups.
restoreCgmsDb.sh	Use this script to restore the IoT FND database from a backup.
setupCgmsDb.sh	Use this script to set up IoT FND database.
startOracle.sh	Use this script to start the IoT FND database.
stopOracle.sh	Use this script to stop the IoT FND database.
setupStandbyDb.sh	(IoT FND database HA installations only) Use this script to set up the standby database server.
setupHaForPrimary.sh	(IoT FND database HA installations only) Use this script to set up the primary database server.
getHaStatus.sh	Run this script to verify that the database is set up for HA.

Note These IoT FND database helper scripts will work if the DB for FND is installed by following the procedure given in the Cisco IoT Field Network Director Installation Guide - Oracle Deployment, Releases 4.3.x and Later.

The helper scripts will not work if custom listener is configured. For example, the ./setupcgmsDB.sh script will set variables and parameters for listeners in listener.ora, tnsnames.ora, and other environment variables or parameters. If the listener.ora and tnsnames.ora are changed, then the helper scripts like restoreCgmsDb.sh might not work. The IoT FND database helper scripts will not work for Oracle RAC clusters. For custom install of DB and also for Oracle RAC clusters, Oracle provided tools like RMAN can be used for backup and restore.

For IoT FND database HA installations, the backup DB uses Oracle Data Guard and has to be setup as given in the Cisco IoT Field Network Director Post-Installation Guide. If `Fast-Start Failover` is not enabled as shown in Setting Up the Observer and a custom setup is done for backup DB, for example by duplicating the DB, then the helper scripts (setupStandbyDb.sh, setupHaForPrimary.sh, getHaStatus.sh) for backup DB also might not work.

Installing and Setting Up the SSM (Utility Deployment)

The Software Security Module (SSM) is a low-cost alternative to a Hardware Security Module (HSM). IoT FND uses the CSMP protocol to communicate with meters, DA Gateway (IR500 devices), and range extenders.

SSM uses Cisco to provide cryptographic services such as signing and verifying CSMP messages, and CSMP Keystore management. SSM ensures Federal Information Processing Standards (FIPS) compliance, while providing services. You install SSM on the IoT FND application server or other remote server. SSM remote-machine installations use HTTPS to securely communicate with IoT FND.

This section describes SSM installation and set up, including:

- Installing or Upgrading the SSM Server
- Uninstalling the SSM Server
- Integrating SSM and IoT FND

With the SSM server installed, configured, and started and with IoT FND configured for SSM, you can view the CSMP certificate on **Admin** > **Certificates** > **Certificate for CSMP**.

Note See the "Setting Up the HSM Client" section in the Generating and Installing Certificates chapter of this book for information on the Hardware Security Module (HSM).

Prerequisites

Ensure that the installation meets the hardware and software requirements listed in the IoT FND Release Notes
.

Installing or Upgrading the SSM Server

To install the SSM server:

Step 1 Run the cgms-ssm-<version>-<release>.<architecture>.rpm rpm script:

```
[root@VMNMS demossm]# rpm -Uvh cgms-ssm-<version>.x86_64.rpm
Preparing... ########################################### [100%]
1:cgms-ssm ########################################### [100%]
```

Step 2 Get the IoT FND configuration details for the SSM. SSM ships with following default credentials:

- ssm_csmp_keystore password: **ciscossm**
- csmp alias name: **ssm_csmp**
- key password: **ciscossm**
- ssm_web_keystore password: **ssmweb**

```
[root@VMNMS demossm]# cd /opt/cgms-ssm/bin/
[root@VMNMS bin]# ./ssm_setup.sh

Software Security Module Server
1. Generate a new keyalias with self signed certificate for CSMP
2. Generate a new keypair & certificate signing request for CSMP
3. Import a trusted certificate
4. Change CSMP keystore password
5. Print CG-NMS configuration for SSM
6. Change SSM server port
7. Change SSM-Web keystore password
```

```
Select available options.Press any other key to exit
Enter your choice :
```

Step 3 Enter 5 at the prompt, and complete the following when prompted:

```
Enter current ssm_csmp_keystore password :ciscossm
Enter alias name : ssm_csmp
Enter key password :ciscossm

security-module=ssm
ssm-host=<Replace with IPv4 address of SSM server>
ssm-port=8445
ssm-keystore-alias=ssm_csmp
ssm-keystore-password=NQ1/zokip4gtUeUyQnUuNw==
ssm-key-password=NQ1/zokip4gtUeUyQnUuNw==
```

Step 4 To connect to this SSM server, copy paste the output from Step 3into the cgms.properties file.

Note You must include the IPv4 address of the interface for IoT FND to use to connect to the SSM server.

Step 5 (Optional) Run the ssm_setup.sh script to:

- Generate a new key alias with self-signed certificate for CSMP
- Change SSM keystore password
- Change SSM server port
- Change SSM-Web keystore password

Note If you perform any of the above operations, you must run the SSM setup script, select “Print CG-NMS configuration for SSM,” and copy and paste all details into the cgms.properties file.

Step 6 Start the SSM server by running the following command:

RHEL Version	Command
8.x	`systemctl start ssm`
7.x	`service ssm start`

Example Output:

```
[root@VMNMS ~]# service ssm start
Starting Software Security Module Server: [ OK ]
```

Monitoring SSM Log Files

You can monitor SSM logs in /opt/cgms-ssm/log/ssm.log

The default metrics report interval is 900 secs (15 min.), which is the minimum valid value. Only servicing metrics are logged. If there are no metrics to report, no messages are in the log.

You can change the metrics report interval by setting the **ssm-metrics-report-interval** field (in secs) in the /opt/cgms-ssm/conf/ssm.properties file.

Note: Your SSM server must be up and running before starting the IoT FND server.

Uninstalling the SSM Server

This section presents steps to completely uninstall the SSM server, including the steps for a fresh installation.

Note: Do not use this procedure for upgrades. Use the procedure in Installing or Upgrading the SSM Server, on page 73.

To uninstall the SSM server:

Step 1 Stop the SSM server by running the following command:

RHEL Version	Command
8.x	`systemctl stop ssm`
7.x	`service ssm stop`

Step 2 Copy and move the /opt/cgms-ssm/conf directory and contents to a directory outside of /opt/cgms-ssm.

Step 3 Uninstall the cgms-ssm rpm:

```
rpm -e cgms-ssm
```

Fresh installations only

Step 4 Install a new SSM server.

Step 5 Copy and overwrite the /opt/cgms-ssm/conf directory with the contents moved in Copy and move the /opt/cgms-ssm/conf directory and contents to a directory outside of /opt/cgms-ssm.

Integrating SSM and IoT FND

Note: You must install and start the SSM server before switching to SSM.

To switch from using the Hardware Security Module (HSM) for CSMP-based messaging and use the SSM:

Step 1 Run the following command to stop IoT FND.

RHEL Version	Command
8.x	`systemctl stop cgms`

RHEL Version	Command
7.x	`service cgms stop`

Step 2 Run the ssm_setup.sh script on the SSM server.

Step 3 Select option 3 to print IoT FND SSM configuration.

Step 4 Copy and paste the details into the cgms.properties to connect to that SSM server.

EXAMPLE

```
security-module=ssm
ssm-host=127.107.155.85
ssm-port=8445
ssm-keystore-alias=ssm_csmp
ssm-keystore-password=NQ1/zokip4gtUeUyQnUuNw==
ssm-key-password=NQ1/zokip4gtUeUyQnUuNw==
```

Step 5 To set up the HSM, specify the following properties in the cgms.properties file (see also Setting Up the HSM Client, on page 45 in the Generating and Exporting Certificates, on page 14 chapter):

```
security-module=ssm/hsm (required; hsm: Hardware Security Module default.)
hsm-keystore-name=testGroup1 (optional; hsm partition name; testGroup1 default)
hsm-keystore-password=TestPart1 (optional; encrypted hsm partition password;
 TestPart1 default)
```

Step 6 Ensure that the SSM up and running and you can connect to it.

Step 7 Start IoT FND.

Integrating SSM and IoT FND

Note You must install and start the SSM server before switching to SSM.

To switch from using the Hardware Security Module (HSM) for CSMP-based messaging and use the SSM:

Step 1 Run the following command to stop IoT FND.

RHEL Version	Command
8.x	`systemctl stop cgms`
7.x	`service cgms stop`

Step 2 Run the ssm_setup.sh script on the SSM server.

Step 3 Select option 3 to print IoT FND SSM configuration.

Step 4 Copy and paste the details into the cgms.properties to connect to that SSM server.

EXAMPLE

```
security-module=ssm
ssm-host=127.107.155.85
ssm-port=8445
```

```
ssm-keystore-alias=ssm_csmp
ssm-keystore-password=NQ1/zokip4gtUeUyQnUuNw==
ssm-key-password=NQ1/zokip4gtUeUyQnUuNw==
```

Step 5 To set up the HSM, specify the following properties in the cgms.properties file (see also Setting Up the HSM Client, on page 45 in the Generating and Exporting Certificates, on page 14 chapter):

```
security-module=ssm/hsm (required; hsm: Hardware Security Module default.)
hsm-keystore-name=testGroup1 (optional; hsm partition name; testGroup1 default)
hsm-keystore-password=TestPart1 (optional; encrypted hsm partition password;
 TestPart1 default)
```

Step 6 Ensure that the SSM up and running and you can connect to it.

Step 7 Start IoT FND.

Installing and Setting Up IoT FND

Complete the following procedures to finish your IoT FND installation:

- Installation and Setup Overview
- Installing IoT FND
- Setting Up IoT FND
- Starting IoT FND
- Checking IoT FND Status
- Running the IoT FND Database Migration Script
- Accessing the IoT FND Web GUI

Prerequisites

To install IoT FND, first obtain the IoT FND installation RPM:

```
cgms-version_number
.x86_64.rpm
```

Note Ensure that /etc/hosts and /etc/resolv.conf files are correctly configured on the IoT FND server.

Installation and Setup Overview

These topics provide an overview of the two types of IoT FND installations:

- Single-Server Deployment
- Cluster Deployment (HA)

Single-Server Deployment

To install and set up IoT FND for a single-server deployment:

- Log in to the RHEL server that will host IoT FND.
- Installing IoT FND.
- Setting Up IoT FND.
- Running the IoT FND Database Migration Script.
- Checking IoT FND Status.
- Accessing the IoT FND Web GUI

Cluster Deployment (HA)

To install and set up IoT FND for HA deployments, repeat the steps in Single-Server Deployment, but only run the IoT FND database migration script once.

Setting up a Cluster on CG-NMS Versions Greater than 2.1

A unique cluster for CG-NMS versions greater than 2.1.x is identified by the tuple (UDP_MULTICAST_ADDR,UDP_MULTICAST_PORT).

Note HA_PARTITION_NAME is not honored now. However, a new parameter CLUSTER_BIND_ADDR is required and it should be set to the IP address of the server that is reachable by other servers in the cluster.

These settings must be set in the /opt/cgms/bin/cgms.conf file. Restart all cluster members after you put these settings on EACH of them.

Note By default, JBOSS starts forming cluster over 228.11.11.11 and port 45688.

Example:

CLUSTER_BIND_ADDR=2.2.55.25

UDP_MULTICAST_ADDR=FFFF::228.11.11.21

UDP_MULTICAST_PORT=45691

Note If you have multiple clusters on the same network, you must configure a different multicast IP and Port pair for each of the clusters.

Installing IoT FND

To install the IoT FND application:

Step 1 Run the IoT FND installation RPM:

```
$ rpm -ivh cgms-version.x86_64.rpm
```

Step 2 Verify installation and check the RPM version:

```
$ rpm -qa | grep -i cgms
cgms-1.0
```

Setting Up IoT FND

To set up IoT FND, run the setupCgms.sh script.

Note If deploying a IoT FND server cluster, the setupCgms.sh script must be run on every node in the cluster.

Caution The IoT FND certificate encrypts data in the database. The setupCgms.sh script runs database migration, which requires access to the IoT FND certificate in the keystore. You must set up certificates before running setupCgms.sh. The script results in an error if it migrates the database and cannot access the certificate (see Generating and Exporting Certificates, on page 14).

Caution Ensure that the database password entered while running the setupCgms.sh script is valid. If you enter an invalid password multiple times, Oracle might lock your user account. You can unlock your account on the database server.

Note For more information about unlocking your password, see “Unlocking the IoT FND Database Password” in the Troubleshooting chapter of the IoT Field Network Director User Guide, Release 4.3.x.

This example uses the setupCgms.sh script to set up a single-server IoT FND system that uses one database.

```
# cd /opt/cgms/bin

# ./setupCgms.sh
07-10-2023 17:10:00 IST: INFO: ========== IoT-FND  Setup Started - 2018-09-13-17-10-00
==========
07-10-2023 17:10:00 IST: INFO: Log file: /opt/cgms/bin/../server/cgms/log/cgms_setup.log
Are you sure you want to setup CG-NMS (y/n)? y

07-10-2023 17:10:02 IST: INFO: User response: y
Do you want to change the database settings (y/n)? y

07-10-2023 17:10:05 IST: INFO: User response: y
Enter database server IP address [example.com]: 128.107.154.246
```

```
07-10-2023 17:11:02 IST: INFO: Database server IP: 128.107.154.246
Enter database server port [1522]:
07-10-2023 17:11:07 IST: INFO: Database server port: 1522
Enter database SID [cgms]:
07-10-2023 17:11:12 IST: INFO: Database SID: cgms
Do you wish to configure another database server for this CG-NMS ? (y/n)? n

07-10-2023 17:11:18 IST: INFO: User response: n
07-10-2023 17:11:18 IST: INFO: Configuring database settings. This may take a while. Please
 wait ...
07-10-2023 17:11:19 IST: INFO: Database settings configured.
Do you want to change the database password (y/n)? y

07-10-2023 17:15:07 IST: INFO: User response: y
Enter database password:
Re-enter database password:
07-10-2023 17:15:31 IST: INFO: Configuring database password. This may take a while. Please
 wait ...
07-10-2023 17:15:34 IST: INFO: Database password configured.
Do you want to change the keystore password (y/n)? n

07-10-2023 17:16:18 IST: INFO: User response: n
Do you want to change the web application 'root' user password (y/n)? n

07-10-2023 17:16:34 IST: INFO: User response: n
Do you want to change IPAM and PSK Settings (y/n)?
07-10-2023 17:16:34 IST: INFO: User response: y
Do you want to use Internal IP Address Management (IPAM) for Loopback (y/n)?

07-10-2023 17:16:34 IST: INFO: User response: y

07-10-2023 17:16:45 IST: Configuring Preferences settings for IPAM. This may take a while.

Please wait…
07-10-2023 17:16:45 IST: Preferences Settings for IPAM completed successfully
Do you want to manage Tunnels using Unique Pre-Shared Keys (y/n)?
07-10-2023 17:16:34 PDT: INFO: User response: y

07-10-2023 17:16:45 IST: Configuring Preferences settings for Tunnel Mgmt. This may take a
 while.
Please wait…
07-10-2023 17:16:45 IST: Preferences Settings for Tunnel Mgmt completed successfully
Do you want to change the FTP settings (y/n)? n

07-10-2023 17:16:45 IST: INFO: User response: n
07-10-2023 17:16:45 IST: INFO: ========== IoT-FND Setup Completed Successfully ==========
```

The setupCgms.sh script lets you configure these settings:

- Configuring Database Settings
- Configuring Database HA
- Configuring the IoT FND Database Password
- Configuring the Keystore Password
- Configuring the Web root User Password
- Configuring FTPS Settings

Configuring Database Settings

To configure the database settings, the setupCgms.sh script prompts you for this information:

- IP address of the primary IoT FND database server
- Port number of the IoT FND database server

  Press Enter to accept the default port number (1522).
- Database System ID (SID), which is cgms for the primary database server

  Press Enter to accept the default SID (cgms). This SID identifies the server as the primary database server.

```
Do you want to change the database settings (y/n)? y

09-13-2018 17:10:05 PDT: INFO: User response: y
Enter database server IP address [example.com]: 128.107.154.246

09-13-2018 17:11:02 PDT: INFO: Database server IP: 128.107.154.246
Enter database server port [1522]:
09-13-2018 17:11:07 PDT: INFO: Database server port: 1522
Enter database SID [cgms]:
09-13-2018 17:11:12 PDT: INFO: Database SID: cgms
```

Configuring Database HA

To configure the standby database settings, the setupCgms.sh script prompts you for the following information:

- IP address of the standby IoT FND database server
- Port number of the standby IoT FND database server

  Enter **1522**.
- Database System ID (SID), which is cgms for the primary database server

  Enter **cgms_s**. This SID identifies the server as the standby database server.

```
Do you wish to configure another database server for this CG-NMS ? (y/n)? y

09-13-2018 17:11:18 PDT: INFO: User response: y
Enter database server IP address []: 128.107.154.20

09-13-2018 17:11:02 PDT: INFO: Database server IP: 128.107.154.20

Enter database server port []: 1522

09-13-2018 17:11:07 PDT: INFO: Database server port: 1522
Enter database SID []: cgms_s

09-13-2018 17:11:12 PDT: INFO: Database SID: cgms_s
09-13-2018 17:11:18 PDT: INFO: Configuring database settings. This may take a while. Please
 wait ...
09-13-2018 17:11:19 PDT: INFO: Database settings configured.
```

For information about setting up database HA, see “Setting Up IoT FND Database for HA” in the following guide: Cisco IoT Field Network Director Post-Installation Guide - Release 4.3.x - High Availability and Tunnel Provisioning

Configuring the IoT FND Database Password

When prompted to change the IoT FND database password, enter the password of the CGMS_DEV account on the database server. If using the default password, do not change the database password now.

```
Do you want to change the database password (y/n)? y

09-13-2018 17:15:07 PDT: INFO: User response: y

Enter database password:
Re-enter database password:

09-13-2018 17:15:31 PDT: INFO: Configuring database password. This may take a while. Please
 wait ...
09-13-2018 17:15:34 PDT: INFO: Database password configured.
```

Configuring the Keystore Password

To configure the keystore password:

```
Do you want to change the keystore password (y/n)? y
09-13-2018 10:21:52 PDT: INFO: User response: y

Enter keystore password: keystore_password
Re-enter keystore password: keystore_password

09-13-2018 10:21:59 PDT: INFO: Configuring keystore password. This may take a while.
Please wait ...
09-13-2018 10:22:00 PDT: INFO: Keystore password configured.
```

Configuring the Web root User Password

To change the password of the root user account that lets you access the IoT FND browser-based interface, enter **y** and provide the password:

```
Do you want to change the web application 'root' user password (y/n)? n
09-13-2018 17:16:34 PDT: INFO: User response: n
```

Configuring FTPS Settings

If deploying a cluster, provide the FTPS settings required for downloading logs. FTPS securely transfers files between cluster nodes. If the FTPS settings are not configured, you can only download logs from the IoT FND node where you are currently logged in.

```
Do you want to change the FTP settings (y/n)? y
09-13-2018 17:16:45 PDT: INFO: User response: y

Enter FTP user password:
Re-enter FTP user password:

09-13-2018 17:16:49 PDT: INFO: Configuring FTP settings. This may take a while. Please wait
 ...
09-13-2018 17:16:57 PDT: INFO: FTP settings configuration completed successfully
```

Running the IoT FND Database Migration Script

IoT FND uses a special database migration system that lets you quickly migrate your IoT FND database without having to perform a database dump and restore. Each database migration creates or modifies some of the tables in the IoT FND database so that IoT FND can keep a record of migrations already performed.

Before launching IoT FND the first time, run the database migration script to set up the IoT FND tables in the database:

```
#cd /opt/cgms/bin
#./db-migrate
```

Note This script runs for a few minutes before launching IoT FND for the first time. Running this script after upgrading to a new version of IoT FND takes longer depending on the amount of data in the IoT FND database.

Note If deploying a IoT FND server cluster, run the db-migrate script on only one cluster node.

The **db-migrate** command prompts you for the database password. The default password is **cgms123**.

Caution Ensure that the password entered while running the db-migrate script is the correct password. If you enter an incorrect password multiple times, Oracle might lock your user account. If so, you have to unlock your account on the database server. Follow the steps below to unlock your password:

- If you enter an incorrect IoT FND Database password multiple times, Oracle locks your user account. Unlock your password using the Oracle software, as shown in this example:

```
# su - oracle
# sqlplus sys/<database_password>@cgms as sysdba
alter user cgms_dev account unlock;
exit;.
```

Accessing the IoT FND Web GUI

IoT FND has a self-signed certificate for its Web GUI. You must add a security exception in your browser to access the IoT FND GUI. Once you start IoT FND, you can access its web GUI at:

The initial default username is root; the password is **root123**.

IoT FND uses the default password of **root123** unless the password was changed when the setup script ran.

For more information on the setup script, see Setting Up IoT FND .

Note If the IoT FND includes the Hardware Security Module (HSM), the Firefox browser will not connect to IoT FND. To work around this issue, open Firefox Preferences, navigate to **Advanced**, and click the **Encryption** tab. Under Protocols, clear the **Use TLS 1.0** check box. Reconnect to IoT FND and ensure that the page loaded properly.

HTTPS Connections

IoT FND only accepts TLSv1.2 based HTTPS connections. To access the IoT FND GUI, you must enable the TLSv1.2 protocol to establish an HTTPS connection with the IoT FND.

Note IoT FND Release 2.1.1-54 and later do not support TLSv1.0 or TLSv1.1 based connections.

First-Time Log In Actions

This section explains the settings that are required when you log in for the first time.

Changing the Password

When you log in to IoT FND for the first time, a popup window prompts you to change the password.

Note IoT FND supports a maximum 32-character password length.

1. Enter your New password.
2. Re-enter the new password in the Confirm Password field.
3. Click Change Password.

Configuring the Time Zone

To configure the time zone, follow these steps:

Step 1 From the *username* drop-down menu (top right), choose **Time Zone**.

Step 2 Select a time zone.

Step 3 Click **Update Time Zone** .

Step 4 Click **OK**.

Changing the Sorting Order of Columns

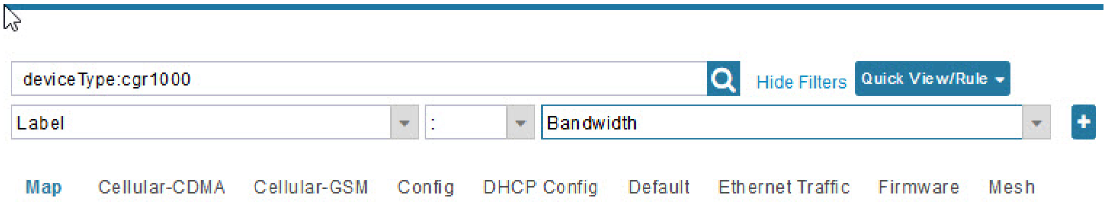
For pages that display lists under a column heading (such as a list of routers) you can change the sort order (ascending or descending) by toggling the triangle icon in the column heading.

Filtering Lists

IoT FND lets you define filters on the DEVICES and OPERATIONS pages.

- To define a filter, click Show Filters to the right of the search field to open a filter definition panel (shown below). After you define the search parameters in the field, click the magnifying glass icon to start search. Results display beneath the filter field.

  In the following example, typing the search string **deviceType:cgmesh status:up** in the Search Devices field lists the mesh endpoint devices with an Up status.



- Click **Hide Filters** to close the search field.

Setting User Preferences for User Interface

You can define what items display in the user interface by selecting the Preferences option under the <*user name*> drop-down menu (top right).

In the User Preferences panel that displays, you can select those items (listed below) that you want to display by checking the box next to that option. Click Apply to save.

User Preference options include:

- Show chart on events page
- Show summary counts on events/issues page
- Enable map:
- Default to map view
- Show device type and function on device pages: Routers, Endpoints, Head End Routers, Servers

Logging Out

Click **Log Out** in the <*user name*> drop-down menu (top right).

IoT FND CLIs

This section addresses key command-line interface (CLI) commands used to manage IoT FND:

- Starting IoT FND
- Checking IoT FND Status
- Stopping IoT FND
- Restarting IoT FND
- IoT FND Log File Location

- IoT FND Helper Scripts
- Uninstalling IoT FND

Starting IoT FND

To start IoT FND, run the following command:

RHEL Version	Command
8.x	`systemctl start cgms`
7.x	`service cgms start`

To configure IoT FND so that it runs automatically at boot time, run this command:

```
chkconfig cgms on
```

Checking IoT FND Status

Before you can start IoT FND, check its connection to the IoT FND database by running the following command:

RHEL Version	Command
8.x	`systemctl status cgms`
7.x	`service cgms status`

Example Output:

```
# service cgms status
IoT-FND Version 4.3.0-78
07-05-2018 15:02:43 PDT: INFO: IoT-FND database server: 2.2.55.8
07-05-2018 15:02:44 PDT: INFO: IoT-FND database connection verified.
07-05-2018 15:02:46 PDT: INFO: IoT-FND application server is up and running.
07-05-2018 15:02:47 PDT: INFO: IoT-FND is up and running.
```

This command provides the IP address or hostname and status of the IoT FND database, and also verifies the connection to the IoT FND database. If the connection is not verified, you cannot start IoT FND.

Stopping IoT FND

To stop IoT FND, run the following command:

RHEL Version	Command
8.x	`systemctl stop cgms`
7.x	`service cgms stop`

Note The application typically takes approximately 10 seconds to stop. Run `ps | grep java` to verify that no Java processes are running.

Restarting IoT FND

To restart IoT FND, run the following command:

RHEL Version	Command
8.x	`systemctl restart cgms`
7.x	`service cgms restart`

IoT FND Log File Location

The IoT FND log file (server.log) is located in the /opt/cgms/server/cgms/log directory.

Uninstalling IoT FND

Note This deletes all IoT FND local installation configuration settings and installation files (for example, the keystore with your certificates).

Tip If you plan to reinstall IoT FND, copy your current keystore and certificate files to use to overwrite the keystore and certificate files included with the install package.

To remove the IoT FND application, run these commands:

```
#rpm -e cgms
# rm -rf /opt/cgms
```

Cleaning up the IoT FND Database

To clean up the IoT FND database:

1. (HA database configurations) Stop the Observer server.
2. (HA database configurations) Run the $ORACLE_BASE/cgms/scripts/ha/deleteStandbyDb.sh script to delete the standby database.
3. (HA database configurations) Run the $ORACLE_BASE/cgms/scripts/ha/deletePrimaryDbHa.sh script to delete the HA configuration from primary database.

4. Run the $ORACLE_BASE/cgms/scripts/deleteCgmsDb.sh script to delete primary database.

IoT FND Log File Location

The IoT FND log file (server.log) is located in the /opt/cgms/server/cgms/log directory.

IOT FND Helper Scripts

The following describes the helper IoT FND scripts in the /opt/cgms/bin/ directory

Script	Description
deinstall_cgms_watchdog.sh	Uninstalls the watchdog script.
install_cgms_watchdog.sh	Installs the watchdog script.
mcast_test.sh	Tests the communication between cluster members.
password_admin.sh	Changes or resets the user password used to access IoT FND.
print_cluster_view.sh	Prints cluster members.

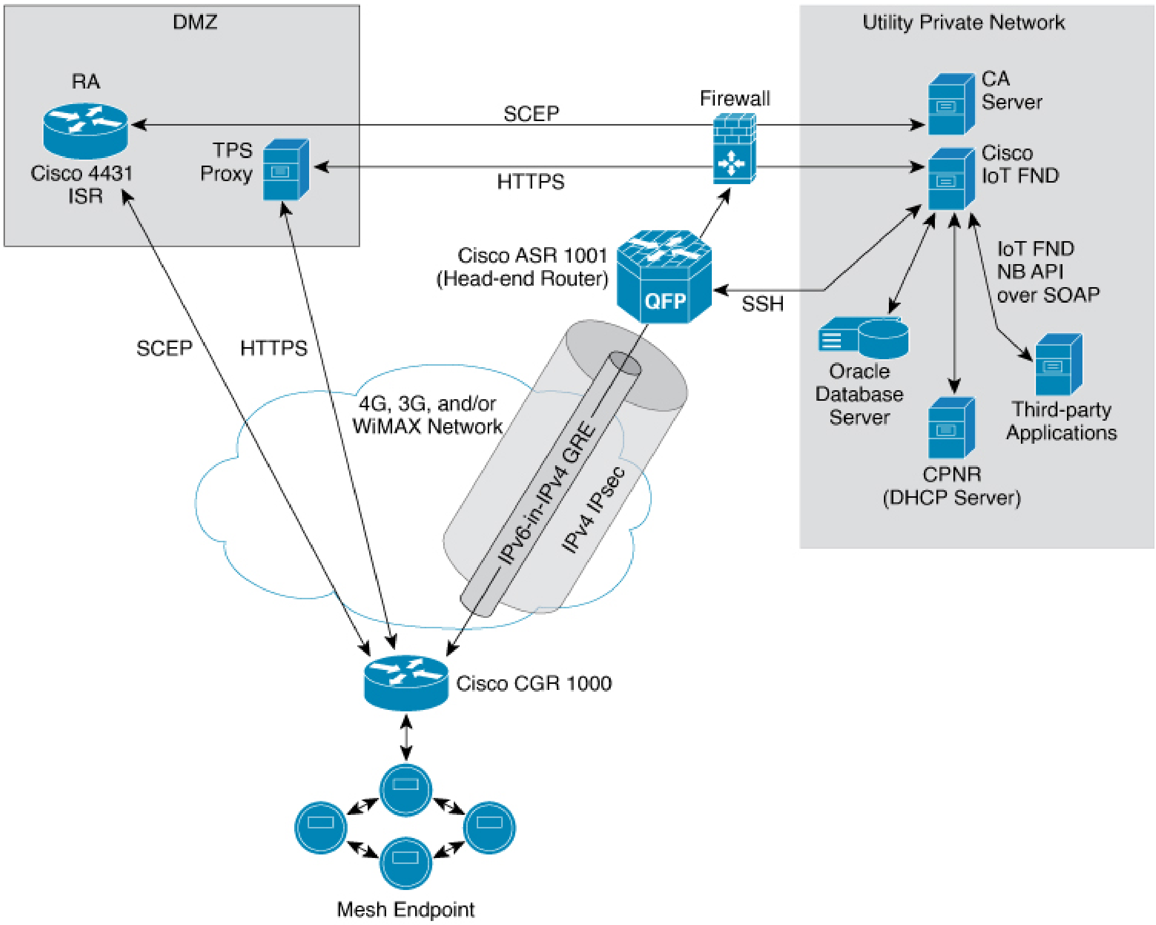
Installing and Configuring the IoT FND TPS Proxy

The first use of the optional TPS proxy is typically when a field area router sends an inbound request to initialize the portion of Zero Touch Deployment (ZTD) handled by IoT FND. IoT FND operates behind a firewall and does not have a publicly reachable IP address. When field area routers (such as CGRs) contact IoT FND for the first time, IoT FND requires that they use the TPS proxy. This server lets these routers contact the IoT FND application server to request tunnel provisioning. See “Managing Tunnel Provisioning” in the Cisco IoT Field Network Director Post-Installation Guide - Release 4.3.x, 4.4.x, 4.5.x and 4.6.x.

The TPS proxy does not have its own GUI. You must edit the properties in the **cgnms.properties** and **tpsproxy.properties-template** files for HTTPS outbound tunnel provisioning requests so that IoT FND recognizes them as requests from the TPS proxy.

After provisioning the tunnel (s), the field area routers can contact IoT FND directly without using the TPS proxy. IoT FND is notified of the exact certificate subject from the proxy certificate, and then authenticates that the HTTPS inbound requests are coming from the TPS proxy.

Figure 1: Zero Touch Deployment Architecture



Setting Up TPS Proxy

To configure the proxy-server settings:

Before you begin

Install the cgms-tpsproxy RPM package Java application on a separate (TPS proxy) server to act as a stateless extension of IoT FND outside the firewall. The TPS proxy can be a Red Hat Enterprise Linux (RHEL) server (see TPS proxy system requirements in the IoT FND Release Notes). The cgnms-tpsproxy application runs as a daemon on the server and requires the following configuration parameters:

- URL of the IoT FND server (to forward inbound requests).
- IP address of the IoT FND server, as part of a whitelist (approved list) for forwarding outbound requests.

Before you install the TPS proxy, obtain the TPS proxy installation package:

```
cgms-tpsproxy-version_number.x86_64.rpm
```

Step 1 Configure a RHEL server to use as the TPS proxy.

Step 2 Connect this RHEL server so that it can be reached while outside the firewall.

Step 3 Configure the TPS proxy using the template file:

```
ssh root@tps_proxy_server
cd /opt/cgms-tpsproxy/conf
cp tpsproxy.properties-template tpsproxy.properties
```

Note Edit the cgnms.properties and tpsproxy.properties files after running the encryption_util.sh script during IoT FND TPS Proxy Enrollment, on page 90

Step 4 Edit the tpsproxy.properties file to add the following lines defining the inbound and outbound addresses for the IoT FND application server:

```
[root@cgr-centos57 conf]# cat tpsproxy.properties-template
inbound-proxy-destination=https://nms_domain_name:9120
outbound-proxy-allowed-addresses=nms_ip_address
cgms-keystore-password-hidden=<obfuscated password>
```

Note You must edit the properties in the cgnms.properties and tpsproxy.properties-template files for HTTPS outbound tunnel provisioning requests so that IoT FND recognizes them as requests from the TPS proxy.

What to do next

Configuring the TPS Proxy Firewall

To configure the TPS proxy firewall:

- Set up a firewall rule to allow HTTPS connections from the TPS proxy to the IoT FND server on port 9120 (for HTTPS inbound requests).
- Set up a firewall rule to allow HTTPS connections from the IoT FND server to the TPS proxy on port 9122 (for HTTPS outbound requests).

IoT FND TPS Proxy Enrollment

The enrollment process for the TPS proxy is the same as the IoT FND enrollment process. The certification authority (CA) that signs the certificate of the IoT FND application server must also sign the certificate of the TPS proxy. The certificate of the TPS proxy is stored in a Java keystore and is similar to the IoT FND certificate.

For the enrollment process, consider these scenarios:

- Fresh installation
  - If the keystore password is the same as the default password, change the default password.

Note We strongly recommend that you change all default passwords. Do not use special characters such as, @, #, !, or + as the encryption_util.sh script cannot encrypt special characters.

- If the keystore password is different from default password, run the encryption_util.sh script and copy the encrypted password to the properties file.

Note Edit the cgnms.properties and tpsproxy.properties files after running the encryption_util.sh script.

- Upgrade

  Regardless of whether you are using the default password or a custom one, the upgrade process encrypts the password in the /opt/cgms-tpsproxy/conf/tpsproxy.properties file.

For information on IoT FND enrollment, refer to the Generating and Exporting Certificates section in the Generating and Exporting Certificates, on page 14 chapter of this guide.

To enroll the terminal TPS proxy:

Step 1 Create a **cgms_keystore** file .

Step 2 Add your certifications to this file.

Step 3 Copy the file to the **/opt/cgms-tpsproxy/conf** directory.

Configuring IoT FND to Use the TPS Proxy

You must edit the properties in the cgnms.properties and tpsproxy.properties-template files for HTTPS outbound tunnel provisioning requests so that IoT FND recognizes them as requests from the TPS proxy. The TPS proxy logs all inbound and outbound requests.

Note If the properties in the cgnms.properties and tpsproxy.properties-template files are not set, IoT FND does not recognize the TPS proxy, drops the forwarded request, and considers it from an unknown device.

Note The following examples employ variable not mandatory values, and are provided as examples only.

To configure IoT FND to use the TPS proxy:

Step 1 Open an SSH connection to the IoT FND server:

```
ssh root@nms_machine
cd /opt/cgms/server/cgms/conf/
```

Note Edit the cgnms.properties and tpsproxy.properties files after running the encryption_util.sh script during IoT FND TPS Proxy Enrollment, on page 90

.

Step 2 Edit the **cgms.properties** file to add lines identifying the TPS proxy IP address, domain name, and user subjects in the cgdm-tpsproxy-subject property:

Note The cgdm-tpsproxy-subject property must match the installed TPS proxy certificate.

```
cgdm-tpsproxy-addr=proxy_server_IP_address
cgdm-tpsproxy-subject=CN="common_name", OU="organizational_unit", O="
organization", L="location", ST="state", C="country"
```

Note Use quotes around comma-separated strings.

Starting the IoT FND TPS Proxy

Start the TPS proxy after it is installed, configured, and enrolled.

To start the TPS proxy, run the start script:

RHEL Version	Command
8.x	`systemctl start tpsproxy`
7.x	`service tpsproxy start`

The TPS proxy log file is located at `/opt/cgms-tpsproxy/log/tpsproxy.log`.

Note For information, see TPS Proxy Validation.

TPS Proxy Validation

The TPS proxy logs all HTTPS inbound and outbound requests in the TPS proxy log file located at /opt/cgms-tpsproxy/log/tpsproxy.log

The following entry in the TPS proxy tpsproxy.log file defines inbound requests for a CGR:

```
73: cgr-centos57: May 21 2014 01:05:20.513 -0700: %CGMS-6-UNSPECIFIED:
 %[ch=TpsProxyServlet-49dc423f][eid=CGR1240/K9+JAF1732ARCJ][ip=192.168.201.5]
[sev=INFO][tid=qtp46675819-29]: Inbound proxy request from [192.168.201.5]
 with client certificate subject [CN=CGRJAF1732ARCJ.example.com,
SERIALNUMBER=PID:CGR1240/K9 SN:JAF1732ARCJ]
```

This message entry in the TPS proxy tpsproxy.log file indicates that the TPS successfully forwarded the message to IoT FND:

```
74: cgr-centos57: May 21 2014 01:05:20.564 -0700: %CGMS-6-UNSPECIFIED:
%[ch=TpsProxyServlet-49dc423f][sev=INFO]
[tid=com.cisco.cgms.tpsproxy.TpsProxyServlet-49dc423f-22]:
Completed inbound proxy request from [192.168.201.5]
 with client certificate subject [CN=CGRJAF1732ARCJ.example.com,
SERIALNUMBER=PID:CGR1240/K9 SN:JAF1732ARCJ]
```

The following entry in the IoT FND server log file identifies the TPS proxy:

```
Request came from proxy
Using forwarded client subject (CN=cg-cgr-1, SERIALNUMBER=PID:CGR1240/K9 SN:JSJ15220047)
 for authentication
```

The following entry in the TPS proxy tpsproxy.log file defines outbound requests:

```
%CGMS-6-UNSPECIFIED: %[ch=TpsProxyOutboundHandler][ip=192.168.205.5]
[sev=INFO][tid=qtp257798932-15]: Outbound proxy request from [192.168.205.5]
 to [192.168.201.5:8443]
```

The following entry in the IoT FND server log file identifies the HTTPS connection:

```
Using proxy at 192.168.201.6:9122 to send to
https://192.168.201.4:8443/cgdm/mgmt commands:
```

Backing Up and Restoring the IoT FND Database

The following topics demonstrate how IoT FND supports both full and incremental database backups:

- Before You Begin
- Creating a Full Backup of the IoT FND Database
- Scheduling a Full IoT FND Backup
- Restoring a IoT FND Backup

Before You Begin

Before backing up your IoT FND database:

- Download and install the latest cgms-oracle-version_number .x86_64.rpm package.
- Copy the scripts, templates, and tools folders from the /opt/cgms-oracle folder to the $ORACLE_BASE/cgms folder.
- Set the ownership of the files and folders you copied to oracle:dba.

Creating a Full Backup of the IoT FND Database

Full backups back up all the blocks from the data file. Full backups are time consuming and consume more disk space and system resources than partial backups.

IoT FND lets you perform full hot backups of IoT FND database. In a hot backup, IoT FND and the IoT FND database are running during the backup

Note The destination backup directory must be writable by the oracle user and have enough space for the IoT FND data.

To create a backup file of the IoT FND software:

Step 1 On the IoT FND database server, open a CLI window.

Step 2 Switch to the user oracle:

```
su - oracle
```

Step 3 Change directory to the location of the IoT FND backup script (backupCgmsDb.sh):

```
cd /home/oracle/app/oracle/cgms/scripts
```

Step 4 Run the backup script and specify the destination folder. For example, to store the backup data in the /home/oracle/bkp folder, enter this command:

./backupCgmsDb.sh full /home/oracle/bkp08-03-2018 15:54:10 PST: INFO: ========== CGMS Database Backup Started ==========08-03-2018 15:54:10 PST: INFO: Log file: /tmp/cgms_backup_restore.logAre you sure you want to backup CG-NMS database (y/n)?**y**

Step 5 Enter y to begin the backup process.

Scheduling a Full IoT FND Backup

To schedule a full IoT FND backup to run daily at 1:00 AM (default setting):

Note The destination backup directory must be writable by the oracle user and have enough space for the IoT FND data.

Step 1 On the IoT FND database server, open a CLI window.

Step 2 Switch to the user *oracle* :

```
su - oracle
```

Step 3 Change directory to the location of the IoT FND backup script (backupCgmsDb.sh):

```
cd /home/oracle/app/oracle/cgms/scripts
```

Step 4 Run the backup script and specify the destination folder.

To change the backup scheduling interval, edit the installCgmsBackupJob.sh script before running it.For example, to store the backup data in /home/oracle/bkp, enter this command:

```
./installCgmsBackupJob.sh /home/oracle/bkp
```

To delete the backup job, enter these commands:

```
cd /home/oracle/app/oracle/cgms/scripts
./deinstallCgmsBackupJob.sh
```

Restoring a IoT FND Backup

Perform database backups and restores using the scripts provided in the cgms-oracle.rpm package. If using the supplied scripts, backups and restores only work if performed on the same Oracle database version.

Note Backups from Oracle version 12.1.0 can only be restored on v12.1.0 if using the supplied scripts. Backups do not work across different versions of Oracle, for example, a backup taken on 12.1.0 cannot be restored on a different version of a future v12.x version using the supplied scripts. If a database upgrade from v12.1.0 to a future v12.x version is required, follow the Oracle upgrade procedure. Refer to the Oracle upgrade document and website.

IoT FND supports restoring IoT FND backups on the same host or different host. If you choose to restore IoT FND backups on a different host, ensure that the host runs the same version of the Oracle database software and that IoT FND database on the destination host was created using the setupCgmsDb.sh script.

Note IoT FND does not support cross-platform backups.

To restore a IoT FND backup:

Step 1 Run the following command to stop IoT FND.

RHEL Version	Command
8.x	`systemctl stop cgms`
7.x	`service cgms stop`

Step 2 Switch to the user oracle, change directories to the script location, and stop Oracle:

```
su - oracle
```

```
cd /home/oracle/app/oracle/cgms/scripts
```

```
./stopOracle.sh
```

Step 3 To restore the IoT FND database, run the command:

```
./restoreCgmsDb.sh full-backup-file
```

Tip Performing a restore from a full backup can be time consuming. For large deployments, we recommend restoring the database from incremental backups.

To restore IoT FND database from an incremental backup, run these commands and specify the path to last incremental backup file:

```
su - oracle
```

```
cd /home/oracle/app/oracle/cgms/scripts
```

```
./restoreCgmsDb.sh last-incr1-backup-file
```

The restore script might display these errors:

```
06-08-2018 13:12:56 PDT: INFO: Import completed successfully
06-08-2018 13:12:56 PDT: INFO: Shared memory file system. Required (1K-blocks): 6084456,
Available (1K-blocks): 4083180
06-08-2018 13:12:56 PDT: ERROR: Insufficient shared memory file system. Increase your
shared memory file system before restoring this database.
06-08-2018 13:12:56 PDT: ERROR: ========== CGMS Database Restore Failed ==========
06-08-2018 13:12:56 PDT: ERROR: Check log file for more information.
```

To avoid these errors, increase the size of the shared memory file system:

```
###### as "root" user
###### Following command allocates 6G to shm. Adjust size as needed.
# umount tmpfs
# mount -t tmpfs tmpfs -o size=6G /dev/shm

###### Edit /etc/fstab and replace defaults as shown below
tmpfs /dev/shm tmpfs size=6G 0 0
```

Step 4 Start Oracle:

```
./startOracle.sh
```

Step 5 Change directories to /opt/cgms and run the db-migrate script:

```
$ cd /opt/cgms
$ bin/db-migrate
```

When you restore a IoT FND database, the restore script restores the database to the IoT FND version the database was using. An error returns if you restore an old database to a newer version of IoT FND. Run the migrate script to ensure that the database runs with the current version of IoT FND.

Step 6 Start IoT FND by running the following command:

RHEL Version	Command
8.x	`systemctl start cgms`
7.x	`service cgms start`

Note For disaster recovery, perform a clean restore. The script starts by deleting the current IoT FND database:

```
$ su - oracle
$ cd /home/oracle/app/oracle/cgms/scripts
$ ./deleteCgmsDb.sh
INFO: ======== CGMS Database Deletion Started - 2011-10-16-07-24-09 ==========
INFO: Log file: /tmp/cgmsdb_setup.log
INFO: Deleting database. This may take a while. Please be patient...
INFO: Delete database completed successfully
INFO: ========== CGMS Database Deletion Completed Successfully
- 2011-10-16-07-25-01 ==========
```

Note If a clean restore is not required, use the Oracle tool to restore the database.

Backing Up the IoT FND Database Incrementally

Incremental backups only back up data file blocks that changed since the previous specified backup. IoT FND supports two incremental backup levels, and an hourly log backup:

- incr0–Base backup for subsequent incremental backups. This is similar to a full backup. For large deployments (millions of mesh endpoints and several thousand routers such as CGR1000 and IR800), run incr0 backups twice a week.
- incr1–Differential backup of all blocks changed since the last incremental backup. For large deployments (millions of mesh endpoints and several thousand routers), run incr1 backups once a day.

Note An incr0 backup must run before an incr1 backup to establish a base for the incr1 differential backup.

- Hourly archivelog backup–The Oracle Database uses archived logs to record all changes made to the database. These files grow over time and can consume a large amount of disk space. Schedule the backup_archive_log.sh script to run every hour. This script backs up the database archive (.arc) log files, stores them on a different server, and deletes the source archivelog files to free space on the database server.

Tip Before performing any significant operation that causes many changes in the IoT FND database (for example, importing a million mesh endpoints or uploading firmware images to mesh endpoints), perform am incr0 backup. After the operation completes, perform another incr0 backup, and then resume the scheduled incremental backups.

Performing an Incremental Backup

Note The destination backup directory must be writable by the oracle user and have enough space for the IoT FND data.

To perform an incremental backup:

Step 1 On the IoT FND database server, open a CLI window.

Step 2 Switch to the user *oracle* and change directory to the location of the IoT FND backup script:

```
su - oracle
cd /home/oracle/app/oracle/cgms/scripts
```

Step 3 Run the backup script and specify the incremental backup level and the destination folder where the backup data is stored (for example, /home/oracle/bkp). For example, to perform an incr0 backup to/home/oracle/bkp, enter the command:

```
./backupCgmsDb.sh incr0 /home/oracle/bkp
```

To perform an incr1 backup, enter the command:

```
./backupCgmsDb.sh incr1 /home/oracle/bkp
```

CHAPTER 5

Upgrading IoT FND

This section contains the following IoT FND upgrade topics:

Pre-Upgrade Checklist

The section identifies the tasks that can be performed before you begin your upgrade to ensure a successful upgrade and limited downtime.

- Back up application directory. For example, if you want to upgrade `cgms RPM`, then you must back up the `/opt/cgms` folder. For more information, refer to Installing or Upgrading the SSM Server, on page 73

> **Note** After upgrade, the manual changes made to the application scripts are lost.

- Back up database. For more information, refer to Creating a Full Backup of the IoT FND Database, on page 93, Backing Up the IoT FND Database Incrementally, on page 97

Verifying Certificates and System Requirements

This section describes how to verify certificates and the system requirements for the upgrade procedure.

- Generating and Exporting Certificates, on page 14

- System Requirements

Upgrading IoT FND and IoT FND TPS Proxy

Note It is not necessary to stop the database during normal upgrades. All upgrades are in-place.

Note For virtual IoT FND installations using custom security certificates, see Managing Custom Certificates, on page 42 before performing upgrade.

Caution Run the following steps sequentially.

To upgrade the IoT FND application:

Step 1 Obtain the new IoT FND ISO from Cisco.

Step 2 Extract the **`cgms rpms`** into a directory from the FND release ISO file.

Step 3 Run **`rpm -qa | grep cgms`** to get the list of rpms installed in the application server.

Step 4 Run the following command to stop IoT FND.

RHEL Version	Command
8.x	`systemctl stop cgms`
7.x	`service cgms stop`

Note The application typically takes approximately 10 seconds to stop.

Step 5 Run **`ps | grep java`** to verify that no Java processes are running.

Step 6 Run the following command to make sure that the **`cgms`** service has stopped.

RHEL Version	Command
8.x	`systemctl status cgms`
7.x	`service cgms status`

Step 7 Run the following script to upgrade the IoT FND RPM.

IoT FND Release	Command

Upgrade to 4.11.0 from any earlier release.	`rpm -Uvh <new_cgms_rpm_filename> --force`
	For example, to upgrade to IoT FND release 4.11.0, run the following command.
	`rpm -Uvh cgms-4.11.0-46.x86_69.rpm --force`

IoT FND Release	Command
Upgrades prior to release 4.11.0.	`rpm -Uvh <new_cgms_rpm_filename>`

Note We recommend you to upgrade all the installed **`rpms`** with the same FND version that you will upgrade to. The new *rpm* files overwrite the existing files in **`/opt/cgms`**.

Note For TPS Proxy, the rpm is bundled in the IoT FND ISO file. Extract the rpm file and copy the rpm to a TPS server. Run the rpm upgrade command `rpm -Uvh cgms-tpsproxy-<fndversion>.rpm` on TPS server.

Step 8 Run **`./db-migrate`** in **`/opt/cgms/bin`** directory to upgrade the database.

Note Ensure that you run the **`db-migrate`** script after each upgrade.

Step 9 Enter the database password when prompted.

Note The default password is **cgms123**.

Step 10 Run the following command to start IoT FND.

RHEL Version	Command
8.x	`systemctl start cgms`
7.x	`service cgms start`

Note You can also use the RHEL (Red Hat Enterprise Linux) GUI to start the IoT FND service (**ADMIN** > **System Management** > **Server Settings** > **Services**).

Post-Upgrade Checklist

This section describes the tasks that you have to perform post upgrade:

Note Any manual changes made to the **`cgms`** scripts are lost post upgrade; therefore, you have to make the changes again.

- Run **`setupCgms`** script to reconfigure FND.

Note The **`setupCgms`** script provides information on new configurations that are part of this FND upgrade.

- Run **`DB migrate`** script to upgrade the database.
- Start **`cgms service`** to monitor the status.

Upgrade FND in HA Configuration or Clustered Mode

This section provides the tasks for upgrading IoT FND in high-availability (HA) configuration or clustered mode:

- Upgrade Oracle DB. For more information, refer to Upgrading the IoT FND Database, on page 70.
- Stop all application servers that are part of the cluster.
- Upgrade all the FND applications.
- Run **`db-migrate`** post upgrade in one of the application servers.
- Start FND service one by one.

CHAPTER 6

High Availability Deployment for IoT FND

This section describes how high availability is achieved for IoT FND:

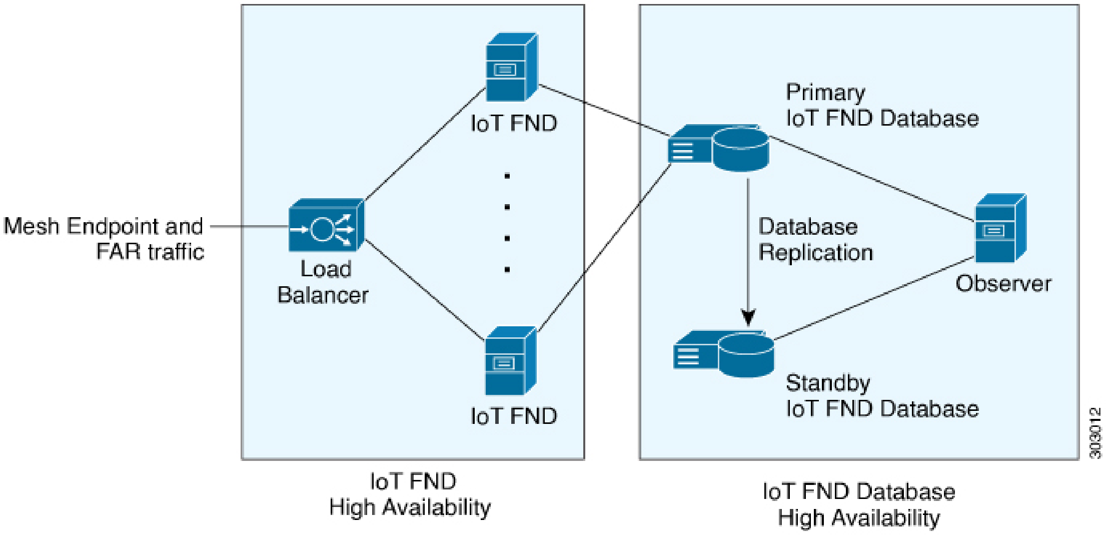
Overview of High Availability Deployment for IoT FND

IoT FND is a critical application for monitoring and managing a connected grid. This chapter discusses on IoT FND solution High Availability (HA), the different components, how it can be achieved, and what configurations are required to achieve it at various levels.

IoT FND provides two main levels of HA:

- IoT FND Server HA—This is achieved by connecting multiple IoT FND servers to a load balancer. Traffic originating at Mesh Endpoints (ME), Field Area Routers (FAR), and Aggregation Services Routers (ASR) goes to the load balancer, which uses a load balancing algorithm to distribute the load among the IoT FND cluster servers.
- IoT FND Database HA—This is achieved by configuring two IoT FND Database servers: a primary server and a standby (or secondary) server. When the primary database receives new data it sends a copy to the standby database. A separate system runs the Observer (the Observer can also run on the standby server), which is a program that monitors the IoT FND Database servers. If the primary database fails, the Observer configures the standby server as the new primary database. IoT FND Database HA works in single and cluster IoT FND server deployments.

Note To set up an Observer server which runs Oracle 12c on a separate server (distinct from the IoT FND Database servers), refer to the following instructions found on the online Oracle Help Center for Oracle 12c (12.1.0.2): Documentation Library.



High Availability Guidelines and Limitations

- IoT FND HA refers to FND server HA. HA for other IoT FND solution components like FAR, ASR, load balancer etc, has to be considered during design for IoT FND Solution HA.
- Zero service downtime is targeted by IoT FND HA, but it is not guaranteed.
- All IoT FND nodes must on the same subnet.
- All IoT FND nodes must run on similar hardware.
- All IoT FND nodes must run the same software version.
- Run the IoT FND setup script (/opt/cgms/bin/setupCgms.sh) on all the nodes.
- Run the DB migration script (/opt/cgms/bin/db-migrate) on only one node.
- The /opt/cgms/bin/print_cluster_view.sh script displays information about IoT FND cluster members.

High Availability of IoT FND Solution Components

IoT FND has mandatory components as well as optional components. FND application server itself and the database server form the mandatory components, however, all other components come under optional because some components are required ONLY as per need, for example, Hardware Security Module (HSM) or Software Security Module (SSM) is required only if end points have to be managed.

Note There are multiple options for IoT FND deployment. IoT FND can also be deployed without tunnels, for example, if you use a private Access Point Name (APN) network or already have a encrypted Multiprotocol Label Switching (MPLS) network for connectivity and choose not to use Tunnel Provisioning Server (TPS) and tunnel provisioning and directly register to IoT FND, in which case, Tunnel High Availability and TPS High Availability might not apply. Also if FND Easy mode is used, then Public Key Infrastructure (PKI) High Availability will not apply.

IoT FND solution components can be classified as below.

FND Solution Component	Description
FND Server HA	IoT FND HA is achieved by connecting multiple IoT FND servers to a load balancer. Traffic originating at MEs, FARs, and ASRs goes to the load balancer, and based on the load balancing algorithm, the load balancer distributes the load among the IoT FND cluster servers.
Load Balancer	There can be HA for load balancer as well, which can be discussed with the Load Balancer product vendor, however, this document discusses load balancer in the context of providing HA for FND servers.
Database High Availability	HA is achieved using Oracle Data Guard, which provides automatic failover between primary and secondary DB servers.
Tunnel High Availability, on page 117	HA is achieved by using alternate links and/or more than one Head End Router (HER) or FAR.
HER Redundancy	HER is not managed by IoT FND, but used for tunnel termination. HA is achieved by having multiple HERs.
FAR Redundancy	FAR redundancy using Hot Standby Router Protocol (HSRP) is supported for the LAN, where one device can act as primary and another can act as secondary router for the LAN and hence for Wireless Personal Area Network (WPAN) components, if the router itself fails. **Note** FAR HA is available only for CGR devices.
Tunnel Provisioning Server High Availability	HA is achieved using load balancer which is similar to FND server.
Public Key Infrastructure	HA can be achieved by having multiple servers that are clustered or load balanced. There can be manual failover as well. It depends on the PKI vendor and product capabilities. **Note** Please consult your PKI vendor for HA capabilities and implementation.

FND Solution Component	Description
Software Security Module	Only manual HA is achieved for SSM at the time of release of 4.9.0 FND. SSM is an optional component that is required only if MEs are managed by IoT FND.
Hardware Security Module	HSM HA can vary. There can be two different HSM servers with one partition on each HSM server. HSM client has to be installed on same RHEL server in which FND server is installed and HSM client also has to be configured appropriately.

Network Requirements for FND HA Setup

- Ensure that the primary server, the secondary server, and the FND have the network connectivity.
- The ports 1522 and 1622 must be open in the primary and secondary servers.

FND Server HA

IoT FND Server HA can be achieved by having two or more IoT FND application servers that are balanced using load balancers. Load balancers can take in incoming connections, monitor the load on each IoT FND and serve traffic accordingly.

Note If there are cgmesh keys, enter these parameters in `/opt/cgms/server/cgms/conf/cgms.properties` file to fetch mesh keys from the primary CGR in a HA setup.

- cgr-ha-fetch-mesh-key-attempts = 3 <-- you can modify the number of attempts to fetch the mesh keys
- cgr-ha-fetch-mesh-key-delay-mins = 1 <-- number of minutes (interval) between mesh-key-attempts

The following configuration is required for IoT FND application server HA.

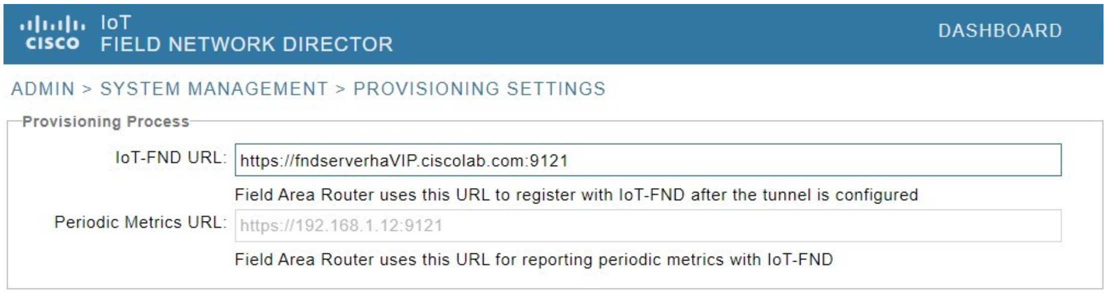
- Configure at the `/opt/cgms/bin/cgms.conf` file that specifies the `CLUSTER_BIND_ADDR` and `UDP_MULTICAST_ADDR`

  CLUSTER_BIND_ADDR= a.b.c.d

  UDP_MULTICAST_ADDR= w.x.y.z

  where `CLUSTER_BIND_ADDR` is the IP address of the server itself and `UDP_MULTICAST_ADDR` must be the same on all instances. It can be either an IPv4 multicast address or an IPv6 address which is not used in the network.

- See Certificate Requirements for IoT FND Server HA Deployment for more information on generating certificate for IoT FND server HA deployment.
- In FND UI, the provisioning settings must point to the cluster VIP IP of the FND servers. For example, if there are fndserver1 and fndserver2, and they are served by fndserverhaVIP.ciscolab.com, then provide it in **Admin** > **System Management** > **Provisioning Settings**.



IoT-FND URL: https://fndserverhaVIP.ciscolab.com:9121

Field Area Router uses this URL to register with IoT-FND after the tunnel is configured

Periodic Metrics URL: https://192.168.1.12:9121

Field Area Router uses this URL for reporting periodic metrics with IoT-FND

Load Balancer

The load balancer plays a critical role in IoT FND HA, as it performs these tasks:

- Load balances traffic destined for IoT FND.
- Maintains heartbeats with servers in the cluster and detects any failure. If a IoT FND server fails, the load balancer directs traffic to other cluster members. The load balancer maintains heartbeats with each IoT FND server in the cluster.

In the health monitoring mechanism of a load balancer, heartbeats will be sent to each FND server that is load balanced. If response is received within a certain interval and if the response is good, then it is marked as active. However, if response is not received or response received is not the expected response from the FND server, then this FND server is marked down. The load balancer again retries after a specific interval.

The default values for checking the heart beat, the time it takes to retry to mark the server up or down etc, depends on vendor specific implementation for load balancer. Heart beats can be implemented as regular http GET messages to IoT FND server on port 80. IoT FND expects an HTTP 200 OK response from an active IoT FND server. If response is not received within certain period of time, then it is marked down. Some load balancers support custom health monitors to be configured as a user-defined script. In such cases, load balancer can get the response from FND server using the following command.

RHEL Version	Command
8.x	`systemctl status cgms`
7.x	`service cgms status`

Example Output:

```
[root@fndtest ~]# service cgms status
IoT-FND Version 4.8.1-72
09-22-2022 16:45:59 IST: INFO: IoT-FND database server: 1.1.1.1
09-22-2022 16:45:59 IST: INFO: IoT-FND database connection verified.
09-22-2022 16:46:00 IST: INFO: IoT-FND application server is up and running.
09-22-2022 16:46:01 IST: INFO: IoT-FND is up and running.
[root@fndtest ~]#
```

The user defined script defined in custom health monitor can check if the last 2 lines show as "up and running", that indicates IoT FND application server is up and running.

Load-Balancing Policies

The table describes the load-balancing policy for each type of traffic the LB supports:

Traffic	Load Balancing Policy
HTTPS traffic to and from browsers and IoT FND API clients (IPv4; ports 80 and 443)	The LB uses Layer 7 load balancing for all traffic from Web browsers and IoT FND API clients. The LB uses stickiness for general HTTPS traffic.
For FAR IPv4 traffic going to ports 9121 and 9120: • Tunnel Provisioning on port 9120 over HTTPS • Regular registration and periodic on 9121 over HTTPS	The LB uses Layer 3 load balancing for all FAR traffic. This is the traffic from the FAR to IoT FND.
For IPv6 CSMP traffic to and from mesh endpoints (MEs): • UDP traffic over port 61624 • Registration • Periodic transmission of metrics • Firmware push • Configuration push • UDP traffic over port 61625 For outage notifications sent by MEs.	The LB uses Layer 3 load balancing for all ME traffic to port 61624, and outage messages to port 61625.

Database High Availability

IoT FND Database HA works in IoT FND single-server and cluster deployments. IoT FND HA uses Oracle Active Dataguard to deploy Oracle HA. To configure HA for the IoT FND Database, use the Oracle Recovery Manager (RMAN) and Dataguard Management CLI (DGMGRL). Table 8: IoT FND Database Helper Scripts are provided by Cisco to achieve this deployment of primary and secondary DB, if the DB is set up ONLY for the purpose of IoT FND application.

If Oracle DB is used by other applications as well, then RMAN and DGMGRL can be used to set up the primary and secondary DB, but in this case, the other helper scripts like backup on FND and the restore of FND scripts might not work because the Oracle environment variables will differ.

Oracle DB HA can be achieved whether we have redundancy at the FND server or not. One or more FND servers can connect to the same Database Data Guard cluster.

The IoT FND Database HA configuration process involves:

- Configuring the primary and secondary databases on separate physical servers or VM.

Note The secondary database server is also referred to as the standby database.

There is a possibility of losing some data during a database failover.

Note If the primary database fails, the associated standby database becomes the primary database. This is transparent to the IoT FND servers. All IoT FND servers in the cluster connect to the new primary database.

- Configuring data replication to be performed over SSL using an Oracle wallet. The wallet contains a self-signed certificate to facilitate quick deployment.

Note The Oracle wallet bundled with the IoT FND RPMs uses self-signed certificates. You can configure custom certificates and wallet to facilitate replication.

There is no performance impact when performing data replication over SSL.

- Using the sys user for replication and not cgms_dev.
- Configuring replication as asynchronous to prevent performance bottlenecks.

By default, IoT FND connects to the database using TCP over port 1522. Replication uses TCPS (TCP over SSL) on port 1622.

The scripts for configuring IoT FND Database HA are included in the IoT FND Oracle Database RPM package (cgms-oracle-version_number.x86_64.rpm). When you install the IoT FND Database, the HA scripts are located in $ORACLE_HOME/cgms/scripts/ha.

Setting Up IoT FND Database for HA

To set up the IoT FND Database HA:

Step 1 Set up the standby database (see Setting Up the Standby Database).

Note Always configure the standby database first.

• The default SID for the standby server is **cgms_s** and *not* cgms.
• Before setting up the standby server for HA, ensure that the environment variable $ORACLE_SID on the standby server is set to **cgms_s**.
• The port is always 1522.

Step 2 Set up the primary database (see Setting Up the Primary Database).

The default SID for the primary server is **cgms**.
Before setting up the primary server for HA, ensure that the environment variable $ORACLE_SID on the primary server is set to **cgms**.

Step 3 Set up IoT FND for database HA (see Setting Up IoT FND for Database HA).

Step 4 Set up the database Observer (see Setting Up the Observer).

Setting Up the Standby Database

To set up the standby database server for HA, run the setupStandbyDb.sh script. This script prompts for configuration information needed for the standby database, including the IP address of the primary database.

```
$ cd $ORACLE_BASE/cgms/scripts/ha
$ ./setupStandbyDb.sh
$ Are you sure you want to setup a standby database ? (y/n)? y
09-20-2012 13:59:18 PDT: INFO: User response: y
09-20-2012 13:59:18 PDT: INFO: CGMS_S database does not exist.
Enter the SYS DBA password. NOTE: This password should be same as the one set on the primary
 server:
Re-enter password for SYS DBA:
09-20-2012 13:59:58 PDT: INFO: User entered SYS DBA password.
Enter new password for CG-NMS database:
Re-enter new password CG-NMS database:
09-20-2012 14:00:09 PDT: INFO: User entered CG-NMS DB password.
Enter primary database server IP address: 192.168.1.12
09-20-2012 14:00:27 PDT: INFO: Cleaning up instance - cgms_s
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
…
Total System Global Area 329895936 bytes
Fixed Size 2228024 bytes
Variable Size 255852744 bytes
Database Buffers 67108864 bytes
Redo Buffers 4706304 bytes
...
09-20-2012 14:00:29 PDT: INFO: ========== CGMS_S Database Setup Completed Successfully
==========
```

Setting Up the Primary Database

To set up the primary database server for HA, run the setupHaForPrimary.sh script. This script prompts for configuration information needed for the primary database, including the IP address of the standby database.

```
$ cd $ORACLE_BASE/cgms/scripts/ha
$ ./setupHaForPrimary.sh
[oracle@pdb ha]$ ./setupHaForPrimary.sh
09-20-2012 13:58:39 PDT: INFO: ORACLE_BASE: /home/oracle/app/oracle
09-20-2012 13:58:39 PDT: INFO: ORACLE_HOME: /home/oracle/app/oracle/product/11.2.0/dbhome_1
09-20-2012 13:58:39 PDT: INFO: ORACLE_SID : cgms
09-20-2012 13:58:39 PDT: INFO: Make sure the above environment variables are what you expect
Are you sure you wish to configure high availability for this database server ? (y/n)? y
09-20-2012 13:58:45 PDT: INFO: User response: y
Enter standby database server IP address: 192.168.1.10
09-20-2012 13:58:56 PDT: INFO: Secondary listener reachable. Moving on with configuration
mkdir: cannot create directory `/home/oracle/app/oracle/oradata/cgms': File exists
09-20-2012 13:58:58 PDT: INFO: Reloading the listener to pick the new settings
LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 13:58:58
…
DGMGRL> 09-20-2012 14:14:54 PDT: INFO: Please start the 'Observer' on appropriate server
for ha
monitoring
Total time taken to perform the operation: 975 seconds
09-20-2012 14:14:54 PDT: INFO: ========== Completed Successfully ==========
```

Setting Up the Observer

The Observer should run on a separate server, but can be set up on the server hosting the standby database.

Note The password required for running Observer is the same as the SYS DBA password. See Creating IoT FND Oracle Database topic in Cisco IoT Field Network Director Installation Guide - Oracle Deployment, Releases 4.3.x and Later for more information.

To set up the Observer:

Step 1 On a separate server, run the observer script.

```
$ cd $ORACLE_BASE/cgms/scripts/ha
$ ./manageObserver.sh start cgms_s password
$ DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
...
Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Observer started
```

Step 2 Run the getHaStatus.sh script to verify that the database is set up for HA.

```
$ cd $ORACLE_BASE/cgms/scripts/ha
$ ./getHaStatus.sh
...
Configuration - cgms_dgconfig

Protection Mode: MaxPerformance
Databases:
cgms - Primary database
cgms_s - (*) Physical standby database

Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS

DGMGRL>
Database - cgms

Role: PRIMARY
Intended State: TRANSPORT-ON
Instance(s):
cgms

Database Status:
SUCCESS

DGMGRL>
Database - cgms_s

Role: PHYSICAL STANDBY
Intended State: APPLY-ON
Transport Lag: 0 seconds
Apply Lag: 0 seconds
Real Time Query: OFF
Instance(s):
cgms_s
```

```
Database Status:
SUCCESS
```

Validating the FND Oracle Database HA

This section provides commands to validate the following post FND Oracle database HA setup.

- Database roles
- Flashback status
- Open mode
- Log into DB and data guard broker
- Check the status of DB and the data guard broker
- DB Sync

Step 1 To check the roles, flashback status, and open mode of the database, run the following commands on the primary and the secondary DB.

a) Primary DB:

```
[oracle@fndPrimaryDB ~]$ sqlplus / as sysdba

SQL> select name, open_mode, flashback_on, DATABASE_ROLE from v$database;

NAME      OPEN_MODE         FLASHBACK_ON       DATABASE_ROLE
--------- ------------     ---------------     --------------
CGMS      READ WRITE           YES                PRIMARY
```

b) Secondary DB:

```
[oracle@fndSecondaryDB ~]$ sqlplus / as sysdba

SQL> select name , open_mode , flashback_on , DATABASE_ROLE from v$database;

NAME      OPEN_MODE            FLASHBACK_ON       DATABASE_ROLE
--------- -------------          -------------      -------------
CGMS      MOUNTED                 YES               PHYSICAL STANDBY
```

Note Ensure that the FLASHBACK_ON is `YES` and the roles and the open mode of the database are displayed in output.

Step 2 Run the following commands in the data guard broker to ensure that the databases are ready for the switchover. The expected output post validation of the servers for switchover is shown in steps a and b.

a) To log into DB and Oracle data guard broker:

```
[oracle@fndSecondaryDB ~]$ dgmgrl sys/<cgmsdba password>
DGMGRL> validate database "cgms";
```

Output:

```
Database Role:    Primary database

Ready for Switchover:  Yes
```

Note You can run the command either on the primary or the secondary DB.

b) To check the status of DB and the data guard broker:

```
DGMGRL> validate database "cgms_s";
```

Output:

```
Database Role:     Physical standby database
Primary Database:  cgms_s

Ready for Switchover:  Yes
Ready for Failover:    Yes (Primary Running)
```

Step 3 Run the following commands to check if the primary and the secondary DB are in sync.

a) Primary:

```
SQL> archive log list;
```

Output:

```

Database log mode                Archive Mode
Automatic archival               Enabled
Archive destination              USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence       6
Next log sequence to archive   8
Current log sequence             8
```

b) Secondary:

```
SQL> select process, status, sequence# from v$managed_standby;
```

Output:

```
PROCESS   STATUS      SEQUENCE#
--------- ------------ ----------
ARCH     CONNECTED             0
ARCH     CLOSING               6
ARCH     CONNECTED             0
ARCH     CONNECTED             0
ARCH     CLOSING               7
RFS      RECEIVING             8
RFS      IDLE                  0
RFS      IDLE                  0
MRP0  APPLYING_LOG           8
```

Note The primary and the secondary DB are in sync only if the output of `MRP0 APPLYING_LOG` in the secondary matches with the `Next log sequence to archive` of the primary as shown in the above output.

Setting Up IoT FND for Database HA

To set up IoT FND for database HA:

Step 1 Stop IoT FND.

Step 2 Run the setupCgms.sh script.

The script prompts you to change the database settings. Enter **y**. Then, the script prompts you to enter the primary database server information (IP address, port, and database SID). After that, the script prompts you to add another database server.

Enter **y**. Then, the script prompts you to enter the standby database server information (IP address, port, and database SID), as follows:

Note IoT FND always uses port 1522 to communicate with the database. Port 1622 is only used by the database for replication.

```
# cd /opt/cgms/bin
# ./setupCgms.sh
05-18-2023 16:32:17 EDT: INFO: ========== IoT-FND Setup Started - 2023-05-18-16-32-17 ==========
05-18-2023 16:32:17 EDT: INFO: Log file: /opt/cgms/bin/../server/cgms/log/cgms_setup.log

Are you sure you want to setup IoT-FND (y/n)? y

05-18-2023 16:33:05 EDT: INFO: User response: y

Do you want to change the database settings (y/n)? y

05-18-2023 16:33:07 EDT: INFO: User response: y

Do you want to use custom database connection string (y/n)? n

05-18-2023 16:33:10 EDT: INFO: User response: n

Enter database server hostname or IP [10.106.13.231]:
05-18-2023 16:33:12 EDT: INFO: Database server: 10.106.13.231

Enter database server port [1522]:
05-18-2023 16:33:14 EDT: INFO: Database server port: 1522

Enter database SID or Service Name [cgms]:
05-18-2023 16:33:15 EDT: INFO: Database SID: cgms

Do you wish to configure another database server for this IoT-FND (y/n)? y

05-18-2023 16:33:18 EDT: INFO: User response: y

Do you want to use custom database connection string (y/n)? n

05-18-2023 16:33:21 EDT: INFO: User response: n

Enter database server hostname or IP []: 10.106.13.232
05-18-2023 16:33:31 EDT: INFO: Database server: 10.106.13.232
Enter database server port []: 1522
05-18-2023 16:33:34 EDT: INFO: Database server port: 1522
Enter database SID or Service Name []: cgms_s
05-18-2023 16:33:41 EDT: INFO: Database SID: cgms_s
05-18-2023 16:33:41 EDT: INFO: Configuring database settings. This may take a while. Please wait ...
05-18-2023 16:33:42 EDT: INFO: Database settings configured.

Do you want to change the database password (y/n)? y

05-18-2023 16:33:47 EDT: INFO: User response: y

Enter database password:
Re-enter database password:

05-18-2023 16:33:52 EDT: INFO: Configuring database password. This may take a while. Please wait ...
05-18-2023 16:33:57 EDT: INFO: Database password configured.

Do you want to change the keystore password (y/n)? n

05-18-2023 16:34:00 EDT: INFO: User response: n
```

```
Do you want to change the web application 'root' user password (y/n)? n

05-18-2023 16:34:02 EDT: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

05-18-2023 16:34:03 EDT: INFO: User response: n

Do you want to change router CGDM protocol settings (y/n)? n

05-18-2023 16:34:05 EDT: INFO: User response: n

Do you want to change router management mode [Demo, Bandwidth Optimized, Default] (y/n)? n

05-18-2023 16:34:05 EDT: INFO: User response: n

Do you want to configure timeseries database (y/n)? n

05-18-2023 16:34:07 EDT: INFO: User response: n
05-18-2023 16:34:07 EDT: INFO: Configuring timeseries flag  none in system properties. This may take
 a while. Please wait...
05-18-2023 16:34:07 EDT: INFO: timeseries flag none

Do you want to change log file settings)? (y/n)? n

05-18-2023 16:34:08 EDT: INFO: User response: n
05-18-2023 16:34:08 EDT: INFO: ========== IoT-FND Setup Completed Successfully ==========
```

Disabling IoT FND Database HA

To disable IoT FND Database HA:

Step 1 On the server running the Observer program, stop the Observer:

```
$ ./manageObserver.sh stop cgms_s password
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Done.
$ Observer stopped
```

Step 2 On the standby IoT FND Database server, delete the standby database:

```
$ ./deleteStandbyDb.sh

Are you sure you want to delete the standby database ? All replicated data will be lost (y/n)? y

09-20-2012 14:27:02 PDT: INFO: User response: y
09-20-2012 14:27:02 PDT: INFO: Cleaning up instance - cgms_s
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Done.
DGMGRL> DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production
```

```
Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Disabled.
DGMGRL> 09-20-2012 14:27:06 PDT: INFO: Removing dataguard configuration
DGMGRL for Linux: Version 11.2.0.3.0 - 64bit Production

Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> Connected.
DGMGRL> Removed configuration
DGMGRL> 09-20-2012 14:27:07 PDT: INFO: Stopping the database

SQL*Plus: Release 11.2.0.3.0 Production on Thu Sep 20 14:27:07 2012

Copyright (c) 1982, 2011, Oracle. All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:27:19

Copyright (c) 1991, 2011, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm)(PORT=1522))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=cgms_s)))
The command completed successfully
Cleaning up instance - cgms_s
09-20-2012 14:27:29 PDT: INFO: ========== Completed Successfully ==========
```

Step 3 On the primary IoT FND Database server, delete the HA configuration:

```
$ ./deletePrimaryDbHa.sh
Are you sure you want to delete the high availability configuration ? All replicated data will be
lost (y/n)? y

09-20-2012 14:25:25 PDT: INFO: User response: y
09-20-2012 14:25:25 PDT: INFO: Removing secondary configuration from primary

SQL*Plus: Release 11.2.0.3.0 Production on Thu Sep 20 14:25:25 2012

Copyright (c) 1982, 2011, Oracle. All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
System altered.
```

```
...
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
09-20-2012 14:25:28 PDT: INFO: Removing data guard config files
09-20-2012 14:25:28 PDT: INFO: Removing standby redo logs
09-20-2012 14:25:29 PDT: INFO: Creating listener file
09-20-2012 14:25:29 PDT: INFO: Listener successfully configured.
09-20-2012 14:25:29 PDT: INFO: Recreating tnsnames ora file
09-20-2012 14:25:29 PDT: INFO: reloading the listener

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:25:29

Copyright (c) 1991, 2011, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm-db2)(PORT=1522)))
The command completed successfully

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 20-SEP-2012 14:25:30

Copyright (c) 1991, 2011, Oracle. All rights reserved.

Starting /home/oracle/app/oracle/product/11.2.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.3.0 - Production
System parameter file is /home/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to /home/oracle/app/oracle/diag/tnslsnr/test-scale-15krpm-db2/cgmstns/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=test-scale-15krpm-db2)(PORT=1522)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=test-scale-15krpm-db2)(PORT=1522)))
STATUS of the LISTENER
------------------------
Alias cgmstns
Version TNSLSNR for Linux: Version 11.2.0.3.0 - Production
Start Date 20-SEP-2012 14:25:30
Uptime 0 days 0 hr. 0 min. 0 sec
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File /home/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Listener Log File /home/oracle/app/oracle/diag/tnslsnr/test-scale-15krpm-db2/cgmstns/alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=test-scale-15krpm-db2)(PORT=1522)))
Services Summary...
Service "cgms" has 1 instance(s).
Instance "cgms", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
09-20-2012 14:25:30 PDT: INFO: ========== Completed Successfully ==========
```

Tunnel High Availability

Tunnels are managed by IoT FND whereas HER is not managed by IoT FND. IoT FND only reads the configuration and maintains a copy of the configuration made on HER. Tunnel redundancy is dependent on:

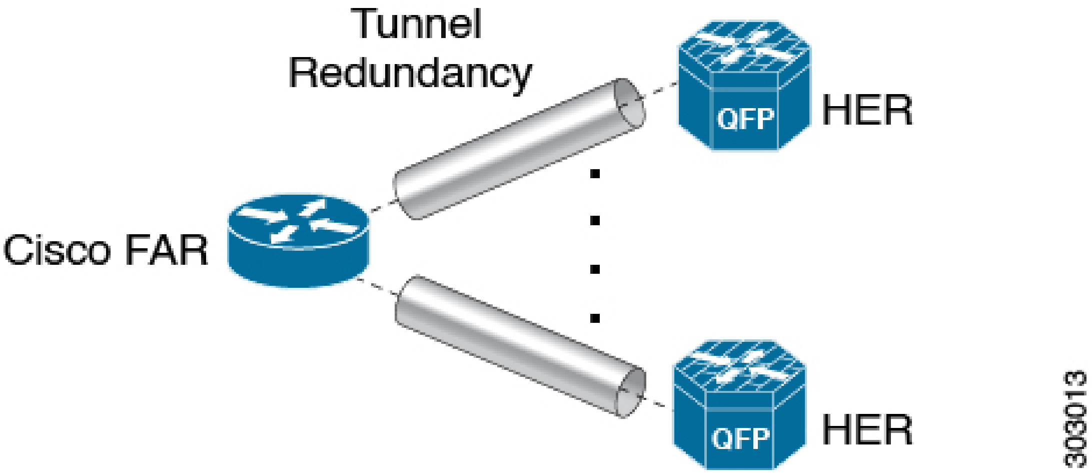
- Link redundancy
- FAR redundancy
- HER redundancy

The redundancy of the tunnel source (that is, CGR), the redundancy of tunnel destination (that is, HER), and the Link between HER and FAR over which the tunnel is formed are critical for tunnel redundancy.

Note If a tunnel fails, traffic flows through another tunnel.

For HA, the below options are possible:

- Option 1: : There can be one FAR, one HER, but more than one link and hence more than one tunnel.
- Option 2: There can be one FAR, but more than one link and one HER.



This involves steps to configure tunnel redundancy at IoT FND.

- HER Redundancy
- Configuring IoT FND for Tunnel Redundancy, on page 119
  - Configuring Tunnel Provisioning Policies - This step involves adding HER to the tunnel provisioning group, and defining policies that determine the mapping between FAR and HER (ASR) interfaces.
  - Modifying the Tunnel Provisioning Templates - The sample FAR tunnel addition template and HER tunnel addition template are provided.

HER Redundancy

Configuration for HER redundancy involves both:

- configuration at the FND. For more information, see Example - one FAR and multiple HER (ASRs) in Tunnel Redundancy.
- configuration at HER (ASR). Flex VPN configuration at HER has to be done once manually when setting up the IoT FND solution. After this, for subsequent addition of FARs, modification is not required at HER. This is because virtual tunnel template Flex VPN config at HER takes care of dynamic Flex VPN between HER (hub) and FAR (spoke). For more information, see https://www.cisco.com/c/en/us/support/security/flexvpn/products-configuration-examples-list.html.

Configuring IoT FND for Tunnel Redundancy

Use tunnel policies to configure multiple tunnels for a FAR. Each tunnel is associated with an interface on a FAR and an HER. If a tunnel provisioning group has one or more HERs, IoT FND displays a policy in the Tunnel Provisioning Policies tab. Use this policy to configure FAR-to-HER interface mapping.

Configuring Tunnel Provisioning Policies

To map FAR-to-HER interfaces in IoT FND:

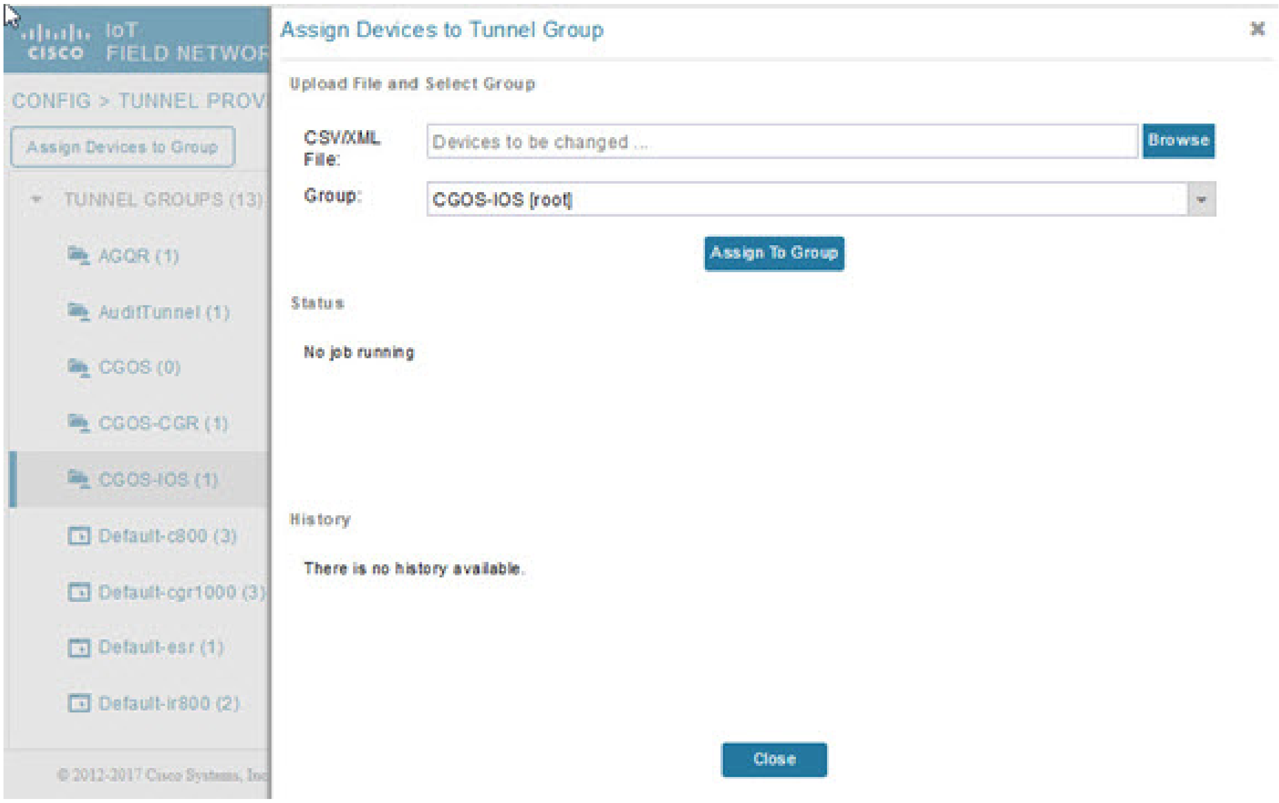
Step 1 Choose **Config** > **Tunnel Provisioning**.

Step 2 In the TUNNEL GROUPS pane, select a group to configure with tunnel redundancy.

Step 3 Create a CSV or XML file that lists the HERs to add to the group in the format *EID*, *device type*, as follows:

```
eid,deviceType
asr-0, asr1000
asr-1, asr1000
asr-2, asr1000
```

Step 4 Click **Assign Devices to Tunnel Group** to import the file and add HERs to the group.



CSV/XML File: Devices to be changed ...

Group: CGOS-IOS [root]

Status

No job running

History

There is no history available.

Note A HER can be a member of multiple tunnel provisioning groups.

Step 5 With the tunnel provisioning group selected, click the **Policies** tab.

By default, IoT FND displays the default-interface-mapping-policy-tunnel-group name for the selected tunnel group within the Policy Name panel.

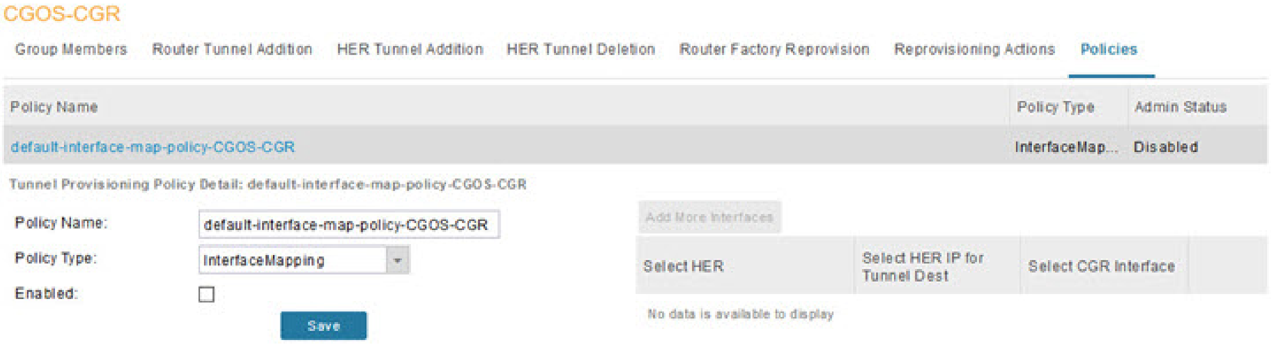
Note Interface-mapping is the only policy type currently supported in IoT FND.

IoT FND displays one interface mapping entry for every HER in the group. You can add or remove interface mapping entries as needed.

Define Policies that Determine Mapping between Interfaces on FAR and HER

To define policies:

Step 1 Click the Policy Name link within the Policy Name Panel to open an entry panel. In the Policy Name field, enter the name of the policy.



Policy Name	Policy Type	Admin Status
default-interface-map-policy-CGOS-CGR	InterfaceMap...	Disabled

Tunnel Provisioning Policy Detail: default-interface-map-policy-CGOS-CGR

Policy Name: default-interface-map-policy-CGOS-CGR
Policy Type: InterfaceMapping
Enabled: □

Select HER	Select HER IP for Tunnel Dest	Select CGR Interface

No data is available to display

To add an interface-mapping entry to the policy, click **Add More Interfaces** (button found above Select HER listing right-side of page). To delete an entry, click **Delete** (X) for that entry.

Step 2 To configure an interface-mapping entry, click the Policy Name link, and complete the following as necessary:

a) To select a different HER, click the currently selected HER and choose a different one from the **Select a HER** drop-down menu.

b) To select the HER IP for the tunnel destination on the HER, click the selected interface and choose a different one from the **Select HER IP** drop-down menu.

c) To select the FAR interface that maps to the selected HER interface, choose an interface from the **Select CGR Interface** drop-down menu.

d) Click **Update**.

Step 3 To enable the policy, check the **Enabled** check box.

Step 4 Click **Save**.

Modifying the Tunnel Provisioning Templates for Tunnel Redundancy

After defining the tunnel provisioning policy for a tunnel provisioning group, modify the Field Area Router Tunnel Addition and the Head-End Router Tunnel Addition templates to include commands to establish the multiple tunnels defined in the policy.

EXAMPLE: Field Area Router Tunnel Addition Template

In this example, bold text indicates the changes made to the default Field Area Router Tunnel Addition template to create multiple tunnels:

```
<#--
Configure a Loopback0 interface for the FAR. This is done first as features
look for this interface and use it as a source.
This is independent of policies
-->
interface Loopback0
<#--
Now obtain an IPv4 address that can be used to for this FAR's Loopback
interface. The template API provides methods for requesting a lease from
a DHCP server. The IPv4 address method requires a DHCP client ID and a link
address to send in the DHCP request. The 3rd parameter is optional and
defaults to "CG-NMS". This value is sent in the DHCP user class option.
The API also provides the method "dhcpClientId". This method takes a DHCPv6
Identity association identifier (IAID) and a DHCP Unique IDentifier (DUID)
and generates a DHCPv4 client identifier as specified in RFC 4361. This
provides some consistency in how network elements are identified by the
DHCP server.
-->
ip address ${far.ipv4Address(dhcpClientId(far.enDuid, 0), far.dhcpV4LoopbackLink).address}/32
<#--
Now obtain an IPv6 address that can be used to for this FAR's loopback
interface. The method is similar to the one used for IPv4, except clients
in DHCPv6 are directly identified by their DUID and IAID. IAIDs used for
IPv4 are separate from IAIDs used for IPv6, so we can use zero for both
requests.
-->
ipv6 address ${far.ipv6Address(far.enDuid, 0, far.dhcpV6LoopbackLink).address}/128
exit
<#-- Make certain the required features are enabled on the FAR. -->
feature crypto ike
feature ospf
feature ospfv3
feature tunnel
<#-- Features ike and tunnel must be enabled before ipsec. -->
feature crypto ipsec virtual-tunnel
<#--
Toggle on/off the c1222r feature to be certain it uses the Loopback0
interface as its source IP.
-->
no feature c1222r
feature c1222r
<#-- Configure Open Shortest Path First routing processes for IPv4 and IPv6. -->
router ospf 1
exit
router ospfv3 2
exit
<#--
Now that OSPF has been configured complete the configuration of Loopback0.
-->
interface Loopback0
ip router ospf 1 area ${far.ospfArea1!"1"}
ipv6 router ospfv3 2 area ${far.ospfV3Area1!"0"}
exit
<#-- Configure Internet Key Exchange for use by the IPsec tunnel(s). -->
crypto ike domain ipsec
identity hostname
policy 1
<#-- Use RSA signatures for the authentication method. -->
authentication rsa-sig
```

```
<#-- Use the 1536-bit modular exponential group. -->
group 5
exit
exit
crypto ipsec transform-set IPSecTransformSet esp-aes 128 esp-sha1-hmac
crypto ipsec profile IPSecProfile
set transform-set IPSecTransformSet
exit
<#--
Define template variables to keep track of the next available IAID (IPv4)
and the next available tunnel interface number. We used zero when leasing
addresses for Loopback0, so start the IAID at one.
-->
<#assign iaId = 1>
<#assign interfaceNumber = 0>
<#--
The same logic is needed for each of the IPsec tunnels, so a macro is used
to avoid duplicating configuration. The first parameter is the prefix to
use when looking for the WAN interface on the FAR to use for the source of
the tunnel. The second parameter is the OSPF cost to assign to the tunnel.
-->
<#macro configureTunnel interfaceNamePrefix destinationInterface her tunnelIndex ospfCost>
<#--
If an interface exists on the FAR whose name starts with the given prefix
and an IPv4 address as been assigned to that interface then the IPsec
tunnel can be configured, otherwise no tunnel will be configured. The
template API interfaces method will return all interfaces whose name
starts with the given prefix.
-->
<#assign wanInterface = far.interfaces(interfaceNamePrefix)>
<#-- Check if an interface was found and it has an IPv4 address. -->
<#if (wanInterface[0].v4.addresses[0].address)??>
<#--
Determine the HER destination address to use when configuring the tunnel.
If the optional property "ipsecTunnelDestAddr1" has been set on this FAR
then use the value of that property. Otherwise look for that same property
on the HER. If the property is not set on the FAR or the HER, then fallback
to using an address on the HER GigabitEthernet0/0/0 interface.
-->
<#assign destinationAddress = her.interfaces(destinationInterface)[0].v4.addresses[0].address>
<#if !(destinationAddress??)>
${provisioningFailed("Unable to determine the destination address for IPsec tunnels")}
</#if>
interface Tunnel${interfaceNumber}
<#assign interfaceNumber = interfaceNumber + 1>
description IPsec tunnel to ${her.eid}
<#--
For a tunnel interface two addresses in their own tiny subnet are
needed. The template API provides an ipv4Subnet method for leasing an
IPv4 from a DHCP server. The parameters match those of ipv4Address,
with a fourth optional parameter that can be used to specify the
prefix length of the subnet to request. If not specified the prefix
length requested will default to 31, which provides the two addresses
needed for a point to point link.
NOTE: If the DHCP server being used does not support leasing an IPv4
subnet, then this call will have to be changed to use the ipv4Address
method and the DHCP server will have to be configured to respond
appropriately to the request made here and the second request that
will have to be made when configuring the HER side of the tunnel.
That may require configuring the DHCP server with reserved addresses
for the client identifiers used in the calls.
-->
<#assign lease = far.ipv4Subnet(dhcpClientId(far.enDuid, tunnelIndex), far.dhcpV4TunnelLink)>
```

```
<#assign iaId = iaId + 1>
<#-- Use the second address in the subnet for this side of the tunnel. -->
ip address ${lease.secondAddress}/${lease.prefixLength}
ip ospf cost ${ospfCost}
ip ospf mtu-ignore
ip router ospf 1 area ${far.ospfArea1!"1"}
tunnel destination ${destinationAddress}
tunnel mode ipsec ipv4
tunnel protection ipsec profile IPSecProfile
tunnel source ${wanInterface[0].name}
no shutdown
exit
</#if>
</#macro>
<#--
Since we are doing policies for each tunnel here, the list of policies passed to this
template can be
iterated over to get the tunnel configuration viz interface mapping
tunnelObject.ipSecTunnelDestInterface is the "interface on CGR"
tunnelObject.ipSecTunnelSrcInterface is the "interface on HER"
tunnelObject.her is the HER of interest
-->
<#list far.tunnels("ipSec") as tunnelObject>
<@configureTunnel tunnelObject.ipSecTunnelDestInterface tunnelObject.ipSecTunnelSrcInterface
tunnelObject.her tunnelObject.tunnelIndex 100/> <----- Loop through policies (aka Tunnels)
</#list>
<#--
Make certain provisioning fails if we were unable to configure any IPsec
tunnels. For example this could happen if the interface properties are
set incorrectly.
-->
<#if iaId = 1>
${provisioningFailed("Did not find any WAN interfaces to use as the source for IPsec
tunnels")}
</#if>
<#--
Configure an IPv6-in-IPv4 GRE tunnel to allow IPv6 traffic to reach the data
center.
-->

<#macro configureGreTunnel destinationInterface her tunnelIndex>
<#assign destinationAddress = her.interfaces(destinationInterface)[0].v4.addresses[0].address>
<#if !(destinationAddress??)>
${provisioningFailed("Unable to determine the destination address for GRE tunnels")}
</#if>
interface Tunnel${interfaceNumber}
<#assign interfaceNumber = interfaceNumber + 1>
description GRE IPv6 tunnel to ${her.eid}
<#--
The ipv6Subnet method is similar to the ipv4Subnet method except instead
of obtaining an IPv4 subnet it uses DHCPv6 prefix delegation to obtain an
IPv6 prefix. The prefix length will default to 127, providing the two
addresses needed for the point to point link. For the IAID, zero was used
when requesting an IPv6 address for loopback0, so use one in this request.
-->
<#assign lease = far.ipv6Subnet(far.enDuid, tunnelIndex, far.dhcpV6TunnelLink)>
ipv6 address ${lease.secondAddress}/${lease.prefixLength}
ipv6 router ospfv3 2 area ${far.ospfV3Area1!"0"}
ospfv3 mtu-ignore
tunnel destination ${destinationAddress}
tunnel mode gre ip
tunnel source Loopback0
no shutdown
```

```
exit
</#macro>
<#-- Loop through the policies for GRE tunnels -->
<#list far.tunnels("gre") as greTunnelObj>
<@configureGreTunnel greTunnelObj.greDestInterface greTunnelObj.her greTunnelObj.tunnelIndex/>
</#list>
```

Head-End Router Tunnel Addition Template

In this example, bold text indicates the changes made to the default Head-End Router Tunnel Addition template to create multiple tunnels:

```
<#--
Define template variables to keep track of the IAID (IPv4) that was used by
the FAR template when configuring the other end of the tunnel. This template
must use the same IAID in order to locate the same subnet that was leased by
the FAR template so both endpoints are in the matching subnet.
-->
<#assign iaId = 1>
<#--
The same logic is needed for each of the IPsec tunnels, so a macro is used.
-->

<#macro configureTunnel ipSecTunnelSrcInterface ipSecTunnelDestInterface her tunnelIndex
ospfCost>
<#--
Only configure the HER tunnel end point if the FAR tunnel end point was
configured. This must match the corresponding logic in the FAR tunnel
template. The tunnel will not have been configured if the WAN interface
does not exist on the FAR or does not have an address assigned to it.
-->
<#assign wanInterface = far.interfaces(ipSecTunnelDestInterface)>
<#if (wanInterface[0].v4.addresses[0].address)??>
<#-- Obtain the full interface name based on the prefix. -->
<#assign interfaceName = wanInterface[0].name>
<#--
Locate a tunnel interface on the HER that is not in use. The template
API provides an unusedInterfaceNumber method for this purpose. All of
the parameters are optional. The first parameter is a name prefix
identifying the type of interfaces, it defaults to "tunnel". The second
parameter is a lower bound on the range the unused interface number must
be in, it defaults to zero. The third parameter is the upper bound on
the range, it defaults to max integer (signed). The method remembers
the unused interface numbers it has returned while the template is
being processed and excludes previously returned numbers. If no unused
interface number meets the constraints an exception will be thrown.
-->
interface Tunnel${her.unusedInterfaceNumber()}
description IPsec tunnel to ${far.eid}
<#assign lease = far.ipv4Subnet(dhcpClientId(far.enDuid, tunnelIndex), far.dhcpV4TunnelLink)>
<#assign iaId = iaId + 1>
ip address ${lease.firstAddress} ${lease.subnetMask}
ip ospf cost ${ospfCost}
ip ospf mtu-ignore
tunnel destination ${wanInterface[0].v4.addresses[0].address}
tunnel mode ipsec ipv4
tunnel protection ipsec profile IPSecProfile
tunnel source ${ipSecTunnelSrcInterface}
no shutdown
exit
router ospf 1
network ${lease.prefix} ${lease.wildcardMask} area ${far.ospfArea1!"1"}
```

```
exit
</#if>
</#macro>
<#list far.tunnels("ipSec") as tunnelObject>
<@configureTunnel tunnelObject.ipSecTunnelSrcInterface tunnelObject.ipSecTunnelDestInterface
tunnelObject.her tunnelObject.tunnelIndex 100/>
</#list>
<#--
Configure an IPv6-in-IPv4 GRE tunnel to allow IPv6 traffic to reach the data
center.
-->
<#macro configureGreTunnel greSrcInterface her tunnelIndex>
interface Tunnel${her.unusedInterfaceNumber()}
description GRE IPv6 tunnel to ${far.eid}
<#assign lease = far.ipv6Subnet(far.enDuid, tunnelIndex, far.dhcpV6TunnelLink)>
ipv6 address ${lease.firstAddress}/${lease.prefixLength}
ipv6 enable
ipv6 ospf 2 area ${far.ospfV3Area1!"0"}
ipv6 ospf mtu-ignore
tunnel destination ${far.interfaces("Loopback0")[0].v4.addresses[0].address}
tunnel mode gre ip
tunnel source ${greSrcInterface}
exit
</#macro>
<#-- Loop through the policies for GRE tunnels -->
<#list far.tunnels("gre") as greTunnelObj>
<@configureGreTunnel greTunnelObj.greSrcInterface greTunnelObj.her greTunnelObj.tunnelIndex/>
</#list>
```

FAR Redundancy

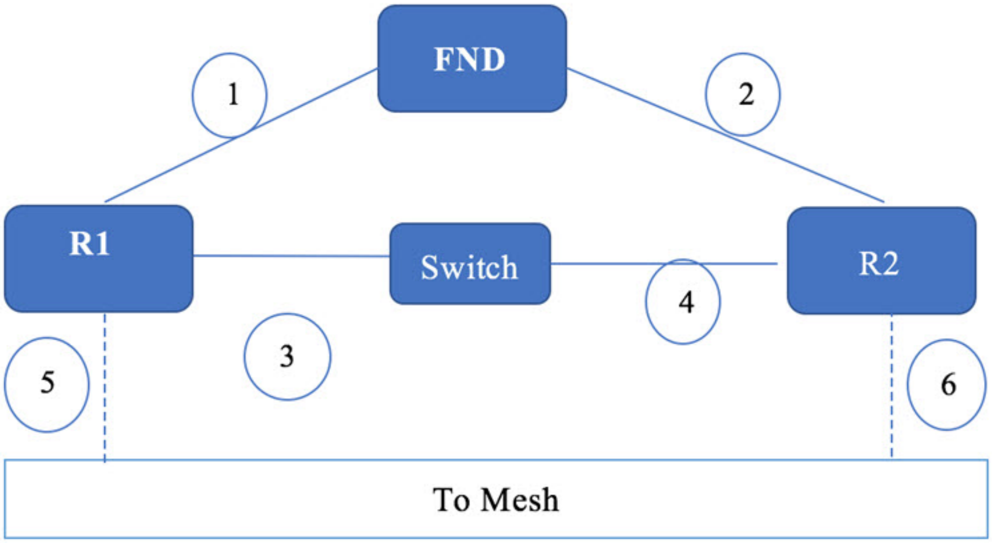
Note The high availibility feature is only supported on CGR1240s and CGM-WPAN-OFDM modules.

To ensure connectivity to the mesh network, you can deploy the following CGR1240 and CGM-WPAN High Availability (HA) network. This network involves two CGR 1240s (R1 and R2), each installed with a CGM-WPAN-OFDM module, which are attached to a mesh network. A switch in the network provides access to the two CGM-WPAN-OFDM modules and manages their connection to the mesh network.

Key Facts:

- CGR1240 HA application is supported on NEW installs only. If you have an existing CGR1240 and CGM-WPAN-OFDM installation that you want to reconfigure for an HA deployment, you must first disable the components and then re-install them.
- You must have a minimum version of Cisco IOS 158-3.M installed on the two CGR1240s.
- Only one CGR1240 can be active at a time.
- Each CGR1240 has its own IP address.
- Each CGR1240 sends its HA state to FND. FND then updates its database.
- CGM-WPAN-OFDM modules must both be running the same software.

Figure 2: CGR1240 High Availability Deployment with CGM-WPAN-OFDM Modules



New Properties:

There are 3 new properties associated with the HA feature. All of the items listed below are populated in FND via CSV import after your two CGRs are installed and configured to support the HA deployment. See CGR Configuration to Support a HA Deployment.

- peerDevice (CGR1240): EIDs for the CGR1240 HA pairs represented as R1 and R2
- haTunnelip: IP address used by HSRP process
- ipsecTunnelDestAddr2: IP addresses of the HA destination tunnel

Table 9: Values You Must Assign To Support a Tunnel (Example Values Only)

peerDevice	haTunnelIp	ipsecTunnelDestAddr2
CGR1240/K9+FTX2150G04T	11.0.0.2	10.255.255.2
CGR1240/K9+FTX2150G04T	11.0.0.1	10.255.255.1

Listed below are sample configurations you would configure on the two CGRs in a HA deployment:

CGR Router 1 (R1)-Active

```
track 1 interface fa2/4 line-protocol
track 2 interface wpan 4/1 line-protocol
Conf t
Int fa2/4
ip address 11.0.0.2 255.255.255.128
standby version 2
standby 12 ip 11.0.0.11
standby 12 preempt
standby 12 track 1 decrement 10
standby 12 track 2 decrement 10
standby 12 track 3 decrement 10
```

```
duplex auto
speed auto
interface Wpan4/1
bandwidth inherit
no ip address
ip broadcast-address 0.0.0.0
standby 12 track 1 decrement 10
standby 12 track 2 decrement 10
no ip route-cache
ieee154 beacon-async min-interval 10 max-interval 10 suppression-coefficient 1
ieee154 dwell window 12400 max-dwell 400
ieee154 panid 440
ieee154 ssid hatest
outage-server 2002:1111:2222::250
peer-to-peer
rpl dag-lifetime 60
rpl dio-dbl 4
rpl dio-min 14
ha enable
control-interface fa2/4 standby 12
peer-ip 11.0.0.1
authentication host-mode multi-auth
authentication port-control auto
ipv6 address 2002:DB8:1111:2222::1/64
ipv6 dhcp server iok-dhcpd6 rapid-commit
dot1x pae authenticator
```

CGR Router 2 (R2)-Standby

```
track 1 interface fa2/4 line-protocol
track 2 interface wpan 4/1 line-protocol
interface fa2/4
ip address 11.0.0.1 255.255.255.0
standby version 2
standby 12 ip 11.0.0.11
standby 12 preempt
standby 12 track 1 decrement 10
standby 12 track 2 decrement 10
standby 12 track 3 decrement 10
duplex auto
speed auto
interface Wpan4/1
bandwidth inherit
no ip address
ip broadcast-address 0.0.0.0
standby 12 track 1 decrement 10
standby 12 track 2 decrement 10
no ip route-cache
shutdown
ieee154 beacon-async min-interval 10 max-interval 10 suppression-coefficient 1
ieee154 dwell window 12400 max-dwell 400
ieee154 panid 440
ieee154 ssid hatest
outage-server 2002:1111:2222::250
peer-to-peer
frame-counter
rpl dag-lifetime 60
rpl dio-dbl 4
rpl dio-min 14
ha enable
control-interface fa2/4 standby 12
peer-ip 11.0.0.2
authentication host-mode multi-auth
authentication port-control auto
ipv6 address 2002:DB8:1111:2222::1/64
```

```
ipv6 dhcp server iok-dhcpd6 rapid-commit
dot1x pae authenticator
```

Tunnel Provisioning Server High Availability

This can be achieved using load balancers very similar to how FND server HA is achieved.

Public Key Infrastructure

Consult the respective PKI vendors on HA capabillities and implementation for Certificate Authority (CA) server and also the Registration Authority (RA) server. For example, if you are using Microsoft PKI solution, then Active Directory Certificate Servers support clustering from Microsoft server 2019 onwards.

Note Clustering is supported only for the Certificate Services role.

For information on certificate SAN field requirements for HA implementation for FND, seeCertificate Requirements for IoT FND Server HA Deployment, on page 20 .

For design of PKI, see Microsoft PKI Architecture.

Clustering is not supported for other role services such as Web Enrollment, Net Device Enrollment, and Online Responder. To make these other role services like web enrollment and others highly available, configure them on separate servers behind a load balancer. For more information, see Active Directory Certificate Services (AD CS) Public Key Infrastructure (PKI) Design Guide.

Software Security Module

SSM or HSM is required only if MEs or meters are managed by IoT FND. Manual HA is available by saving a snapshot of the server and then recovering it if the main server is down.

Hardware Security Module

HSM is available as cloud service or Peripheral Component Interconnect (PCI) cards or HSM appliance. It stores the keys pair and certificates used by mesh endpoint communications. HSM appliance from Thales group is supported by IoT FND solution.

HSM HA involves:

- setup of the HSM appliance. For more information, see Installing and Configuring New HSM. HA deployment options with HSM and IoT FND are:
  - two different partitions on the same HSM server or
  - two HSM appliances with one partition on each HSM appliance.

  In both the cases, one HSM partition can act as primary, and another partition can act as secondary. Each HSM client on FND server will have primary partition of HSM and secondary partition of HSM configured.
- HA configuration at the HSM client. This HSM client is provided by Thales group and has to be downloaded from the Thales group website. This client has to be installed on the same linux server where

FND application server is installed. If there are multiple FND servers, then HSM client has to be installed on all FND servers.

For more information on HSM client version, see Hardware Security Module (HSM) Upgrade Table.

For more information, see Configuring the HSM HA Client

User Interface

IoT FND 4.3 has a new tab, WPAN HA, that appears on CGR1000 pages that displays details on the two CGRs (active and standby) and the HA status of each router.

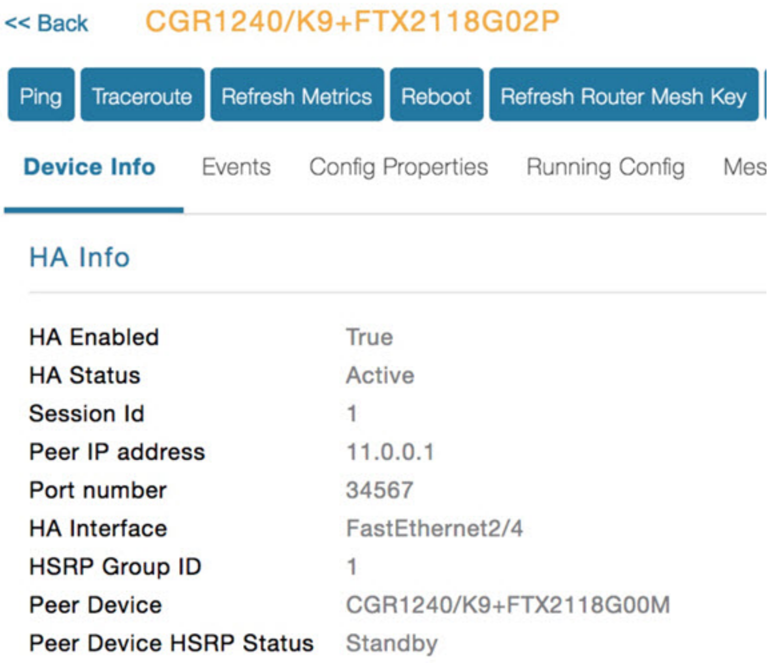
The following items are tracked for each CGR1240 pair:

- Last Heard
- HA Status (Active or Standby)
- Peer Device
- Peer Status
- Peer HA Status
- Group ID
- Overall HA Status

You can also view additional information for CGR HA pairs at the DEVICE > FIELD DEVICES page for the CGR1000:

- Mesh Link Keys (Key Refresh Time and Key Expiration Time)
- HA Info on Device Info tab (Figure 2): Enabled state, HA Status, Session ID, Peer IP address, Port Number, HA Interface, HSRP Group ID, Peer Device, Peer Device HSRP Status

Figure 3: HA Info



HA Info

HA Enabled	True
HA Status	Active
Session Id	1
Peer IP address	11.0.0.1
Port number	34567
HA Interface	FastEthernet2/4
HSRP Group ID	1
Peer Device	CGR1240/K9+FTX2118G00M
Peer Device HSRP Status	Standby

Troubleshooting Steps to Resolve Failover or Switchover Operation Failure

Use the following steps to validate the HA servers when there is a failure of failover or switchover operation.

- Validate the fal parameters mismatch
- Validate the redo log and standby log
- Validate the flashback_on (primary and secondary DB)
- Start the recovery manager and enable the fast_start failover

Note The steps recommended here are some of the common issues that are faced when configuring HA. Therefore, it is recommended to take help from a DBA when configuring the HA to find the root cause and fix any setup-specific issues.

Step 1 **Validate the fal parameters mismatch**: Run the query below on both primary and secondary DB to ensure that the fal parameters are matching. The expected output of the matching fal parameters of the primary and secondary DB is shown below.

If the fal parameters are not matching, then use the alter command to fix as shown in steps a and b.

Note The fal_client of primary DB is the fal_server of secondary and the fal_server of primary DB is the fal_client of secondary.

Expected output of the matching fal parameters:

Primary DB:

```
SQL> show parameter fal
NAME          TYPE        VALUE
------------  ---------- --------
fal_client    string      cgms_p
fal_server    string      cgms_s
```

Secondary DB:

```
SQL> show parameter fal
NAME          TYPE        VALUE
------------  ---------- --------
fal_client    string      cgms_s
fal_server    string      cgms_p
```

a) If the fal parameters of the databases are not matching, then run the following commands to fix the fal parameter mismatch.

```
alter system set  fal_client=cgms_p;
alter system set fal_server=cgms_s;
```

b) Validate the tnsping on both servers after fixing the fal parameters:

```
su - oracle
tnsping cgms_p
tnsping cgms_s
```

Ensure that the tnsping works for both `fal_client` and `fal_server`. If the tnsping fails, then check the tnsnames.ora file that is located in the `$ORACLE_HOME/network/admin` folder. Make sure that there is an entry for `cgms_p` and `cgms_s` in the `tnsnames.ora` file. A sample output of the ora file is shown. Any missing entry in the ora file is added manually.

Sample ora file output

```
cgms =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = fndPrimaryDB)(PORT = 1522))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cgms)
  )
 )
cgms_p =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = fndPrimaryDB)(PORT = 1622))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cgms)
  )
 )
cgms_ss =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = 10.104.198.103)(PORT = 1622))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cgms_s)
  )
 )
cgms_s =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = 10.104.198.103)(PORT = 1622))
  (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cgms_s)
  )
 )
```

Step 2 **Validate the redo log and standby log**: Ensure that the redo log outputs are same in both primary and secondary DB as shown in the sample output below:

```
SQL> select THREAD#, count(*) from v$log group by THREAD#;

THREAD#   COUNT(*)
-------- ----------
    1          3

SQL> select THREAD#, count(*) from v$STANDBY_LOG group by THREAD#;

THREAD#   COUNT(*)
-------- ----------
     1          4
```

If the redo logs are not matching as shown in the above output, then follow the steps below to fix the redo log mismatch:

a) Run the following scripts:

```
SQL>  select THREAD#, count(*) from v$log group by THREAD#;

THREAD#   COUNT(*)
-------- ----------
    1          3


SQL> select THREAD#, count(*) from v$STANDBY_LOG group by THREAD#;

THREAD#   COUNT(*)
-------- ----------
    0          2
    1          2
```

b) Run the following script in the standby database.

```
dgmgrl sys/cgmsDba123
disable fast_start failover;
exit

sqlplus / as sysdba
recover managed standby database cancel;


create a sql file like "a.sql" with below content.
vi a.sql[copy the below content and paste in the file]

set linesize 200
set pagesize 3000
set echo on
set feedback on
spool standby_redo.log
set time on
ALTER DATABASE CLEAR LOGFILE GROUP 4;
ALTER DATABASE CLEAR LOGFILE GROUP 5;
ALTER DATABASE CLEAR LOGFILE GROUP 6;
ALTER DATABASE CLEAR LOGFILE GROUP 7;
ALTER DATABASE DROP LOGFILE GROUP 4;
ALTER DATABASE DROP LOGFILE GROUP 5;
ALTER DATABASE DROP LOGFILE GROUP 6;
ALTER DATABASE DROP LOGFILE GROUP 7;

alter database add standby logfile thread 1 group 4
'/home/oracle/app/oracle/oradata/cgms_s/srl01.log' size 2147483648 reuse;
alter database add standby logfile thread 1 group 5
'/home/oracle/app/oracle/oradata/cgms_s/srl02.log' size 2147483648 reuse;
alter database add standby logfile thread 1 group 6
'/home/oracle/app/oracle/oradata/cgms_s/srl03.log' size 2147483648 reuse;
alter database add standby logfile thread 1 group 7
'/home/oracle/app/oracle/oradata/cgms_s/srl04.log' size 2147483648 reuse;

spool off


execute the a.sql file in the sql prompt.

sqlplus / as sysdba
@a.sql
```

c) Run the following command in the primary database.

```
create a sql file like "a.sql" with below content.
vi a.sql[copy the below content and paste in the file]


set time on
set feedback on
set echo on
spool standby.log

ALTER DATABASE CLEAR LOGFILE GROUP 4;
ALTER DATABASE CLEAR LOGFILE GROUP 5;
ALTER DATABASE CLEAR LOGFILE GROUP 6;
ALTER DATABASE CLEAR LOGFILE GROUP 7;


ALTER DATABASE DROP LOGFILE GROUP 4;
ALTER DATABASE DROP LOGFILE GROUP 5;
ALTER DATABASE DROP LOGFILE GROUP 6;
ALTER DATABASE DROP LOGFILE GROUP 7;

alter database add standby logfile thread 1 group 4 '/home/oracle/app/oracle/oradata/cgms/srl01.log'
 size 2147483648 reuse;
alter database add standby logfile thread 1 group 5 '/home/oracle/app/oracle/oradata/cgms/srl02.log'
 size 2147483648 reuse;
alter database add standby logfile thread 1 group 6 '/home/oracle/app/oracle/oradata/cgms/srl03.log'
 size 2147483648 reuse;
alter database add standby logfile thread 1 group 7 '/home/oracle/app/oracle/oradata/cgms/srl04.log'
 size 2147483648 reuse;

spool off

execute the a.sql file in the sql prompt.

sqlplus / as sysdba
@a.sql
```

d) Verify the redo logs after fixing the mismatch: On running the above scripts (steps a, b, and c), redo logs should be the same on both primary and secondary DB. To confirm, run the following commands on both primary and secondary DB.

```
select THREAD#, count(*) from v$log group by THREAD#;
select THREAD#, count(*) from v$STANDBY_LOG group by THREAD#;
```

If the output of the primary and secondary DB are the same, then redo log issue is fixed.

Step 3 **Validate the '`flashback_on`' on both primary and secondary DB**:

a) Run the following commands on both primary and secondary DB:

```
su - oracle
sqlplus / as sysdba
set linesize 200
set pagesize 3000
col file_name for a80
col member for a80
select name , open_mode , flashback_on , DATABASE_ROLE from v$database;

SQL> select name , open_mode , flashback_on , DATABASE_ROLE from v$database;

NAME      OPEN_MODE          FLASHBACK_ON       DATABASE_ROLE
--------- ----------------- ---------------- --------------
CGMS      READ WRITE           YES                PRIMARY
```

b) Check the tablespace flashback.

```
select a.file#, a.name file_name, b.ts#, b.name ts_name, b.flashback_on from v$datafile a,
v$tablespace b where a.ts#=b.ts#;

SQL> select a.file#, a.name file_name, b.ts#, b.name ts_name, b.flashback_on from v$datafile a,
v$tablespace b where a.ts#=b.ts#;

FILE# FILE_NAME                                           TS# TS_NAME                      FLA
---- -------------------------------------------------- ---------- ----------------------- ---
1 /home/oracle/app/oracle/oradata/CGMS/system01.dbf       0 SYSTEM                         YES
3 /home/oracle/app/oracle/oradata/CGMS/sysaux01.dbf       1 SYSAUX                         YES
4 /home/oracle/app/oracle/oradata/CGMS/undotbs01.dbf      2 UNDOTBS1                       YES
7 /home/oracle/app/oracle/oradata/CGMS/users04.dbf        4 USERS                          YES
8 /home/oracle/app/oracle/oradata/CGMS/users05.dbf        4 USERS                          YES
9 /home/oracle/app/oracle/oradata/CGMS/users06.dbf        4 USERS                          YES
10 /home/oracle/app/oracle/oradata/CGMS/users07.dbf       4 USERS                          YES
11 /home/oracle/app/oracle/oradata/CGMS/users08.dbf       4 USERS                          YES
12 /home/oracle/app/oracle/oradata/CGMS/users09.dbf       4 USERS                          YES
13 /home/oracle/app/oracle/oradata/CGMS/users10.dbf       4 USERS                          YES
14 /home/oracle/app/oracle/oradata/CGMS/users11.dbf       4 USERS                          YES
15 /home/oracle/app/oracle/oradata/CGMS/users12.dbf       4 USERS                          YES
16 /home/oracle/app/oracle/oradata/CGMS/users13.dbf       4 USERS                          YES
17 /home/oracle/app/oracle/oradata/CGMS/users14.dbf       4 USERS                          YES
6 /home/oracle/app/oracle/oradata/CGMS/users01.dbf        4 USERS                          YES
5 /home/oracle/app/oracle/oradata/CGMS/users02.dbf        4 USERS                          YES
18 /home/oracle/app/oracle/oradata/CGMS/users15.dbf       4 USERS                          YES
2 /home/oracle/app/oracle/oradata/CGMS/users03.dbf        4 USERS                          YES
```

Note Ensure that the `flashback_on` is 'yes' for the database and also for the tablespace.

If the `flashback_on` is not enabled for the tablespace, then follow the steps given below to enable:

1. Run the following commands in the secondary DB:

```
sqlplus / as sysdba

alter tablespace SYSTEM flashback on;
alter tablespace SYSAUX flashback on;
alter tablespace UNDOTBS1 flashback on;
alter tablespace USERS flashback on;
```

2. Run the following commands in the primary DB.

   ```
   sqlplus / as sysdba
   shutdown immediate
   startup mount
   alter tablespace SYSTEM flashback on;
   alter tablespace SYSAUX flashback on;
   alter tablespace UNDOTBS1 flashback on;
   alter tablespace USERS flashback on;
   alter database open;
   ```

c) On executing the above commands, the `flashback_on` should be enabled on the DB and also on the tablespace. Verify using the following commands:

```
select name , open_mode , flashback_on , DATABASE_ROLE from v$database;
select a.file#, a.name file_name, b.ts#, b.name ts_name, b.flashback_on from v$datafile a,
v$tablespace b where a.ts#=b.ts#;
```

Step 4 **Start the recovery manager and enable the fast_start failover**:

a) To start the recovery manager in the standby DB:

```
sqlplus / as sysdba
recover managed standby database parallel 4 nodelay disconnect from session;
```

b) To enable fast_start failover:

```
dgmgrl sys/cgmsDba123
show configuration verbose;
enable fast_start failover;
```

The above steps (1 through 4) provide guidance and is likely to resolve the failover or switchover issues on the HA servers.