Installing Cisco IoT FND

This chapter provides an overview of the steps required to install Cisco IoT Field Network Director (Cisco IoT FND) in your network.

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 Guide, Release 4.2.x.

How to install IoT FND and related software:

- Before You Install IoT FND
- 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
- Deploying IoT FND/Oracle/TPS Virtual Machines on ESX 5.x

Before You Install IoT FND

Use the procedures in the following sections to prepare for your IoT FND installation:

- IoT FND Map View Requirements
- System Requirements
- Obtaining IoT FND and CNR Licenses
- Installing the Linux Packages Required for Installing Oracle
- Obtaining IoT FND RPM Packages
- Configuring NTP Service
- IoT FND Installation Overview

IoT FND Map View Requirements

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 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 tile servers. Only IoT FND operator browsers are allowed access to the GIS map tile 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 ask-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 tiles. No utility operational data or asset information is ever used (that is, sent over Internet) to retrieve Google map tiles. Map tiles are retrieved only using geographic location information.

**FQDN INFORMATION EXAMPLE**

For example, your non-cluster installation has a domain named UtilityA.com, and cgms1 as the hostname with an FQDN of cgms1.UtilityA.com. You would email ask-fnd-pm-external@cisco.com and include the FQDN, cgms1.UtilityA.com.

In a cluster deployment with one or more IoT FND servers and a load balancer with the FQDN of loadbalancer-vip, which directs traffic to the cgms-main or cnms-dr cluster (DR installations), you would email ask-fnd-pm-external@cisco.com and include the FQDN, loadbalancer-vip.UtilityA.com.

**System Requirements**

Refer to the [IoT FND Release Notes](#) for the latest details on hardware and software requirements, as well as requirements for large scale deployments.

**Obtaining IoT FND and CNR Licenses**

- Contact your Cisco partner to obtain the necessary licenses to use IoT FND and CNR.
- Obtain a license to use SafeNet as your HSM for mesh endpoint security.

**Installing the Linux Packages Required for Installing Oracle**

Install these packages in this order 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:

<table>
<thead>
<tr>
<th>RPM Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgms-version_number.x86_64.rpm</td>
<td>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.</td>
</tr>
<tr>
<td>cgms-oracle-version_number.x86_64.rpm</td>
<td>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.</td>
</tr>
<tr>
<td>cgms-tools-version_number.x86_64.rpm</td>
<td>Contains a few optional command-line tools. If needed, install this package on the system running the IoT FND application server.</td>
</tr>
<tr>
<td>cgms-ssm-version_number.x86_64.rpm</td>
<td>Contains the Software Security Module (SSM). Install this package on the system running the IoT FND application server.</td>
</tr>
<tr>
<td>cgms-tpsproxy-version_number.x86_64.rpm</td>
<td>Contains the TPS proxy application. Install this package on the IoT FND TPS proxy system.</td>
</tr>
</tbody>
</table>

Configuring NTP Service

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.

To configure NTP on your RHEL servers:

1. **Configure the */etc/ntp.conf* file.**
   
   For 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
   ...
   ```

2. **Restart the NTP daemon and ensure that it is set to run at boot time.**
   ```
   service ntpd restart
   chkconfig ntpd on
   ```

3. **Check the configuration changes by checking the status of the NTP daemon.**
   ```
   # ntpq -p
   +-----------------+-----------------+-----------------+---------------+---------------+----------------+-----------------+
   | remote          | refid           | st  t when poll | delay         | offset        | jitter         |
   +-----------------+-----------------+-----------------+---------------+---------------+----------------+-----------------+
   | 192.0.2.1        | 198.51.100.1    | 3 u 309 1024    | 377           | 0.694         | 0.899          | 0.435           |
   | LOCAL(0)         | LOCAL          | 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 Installation Overview

Complete the following procedures to install IoT FND:

1. Installing and Setting Up the IoT FND Database.
2. Installing and Setting Up IoT FND.
3. Installing and Configuring the IoT FND TPS Proxy.

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 Overview

The following topics provide an overview of IoT FND deployment:

- Single-Server Deployment
- High Availability Deployment

Single-Server Deployment

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

1. Log in to the database server.
2. Downloading and Unpacking Oracle Database.
3. Running the Oracle Database Installer.
4. Setting Up the IoT FND Database.

High Availability Deployment

To install and set up IoT FND database for HA:

1. Log in to the primary IoT FND database server.
2. Downloading and Unpacking Oracle Database.
3. Running the Oracle Database Installer.
4. Log in to the standby database server.
5. Downloading and Unpacking Oracle Database.
6. Running the Oracle Database Installer.

7. Setting Up IoT FND Database for HA.

Downloading and Unpacking Oracle Database

To download the Oracle database:

1. Log in to your server as root.

2. Download Oracle 11g Enterprise Edition (11.2.0.3 64-bit) or Oracle12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production (with Patch 20830993).

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

   `xhost + local:oracle`

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

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

5. Unpack the Oracle Database zip archives.

   `p10404530_112030_Linux-x86-64_1of7.zip`
   `p10404530_112030_Linux-x86-64_2of7.zip`
   `p10404530_112030_Linux-x86-64_3of7.zip`
   `p10404530_112030_Linux-x86-64_4of7.zip`
   `p10404530_112030_Linux-x86-64_5of7.zip`
   `p10404530_112030_Linux-x86-64_6of7.zip`
   `p10404530_112030_Linux-x86-64_7of7.zip`

Running the Oracle Database Installer

**Note:** Before running the Oracle installer, disable the firewall.

To install the Oracle database:

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

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

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

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

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

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

6. Click **Enterprise Edition (4.29GB (Oracle 11g) or 6.4GB (Oracle12c)**, and then click **Next**.

7. Select the following two default installation values, Oracle Base and Software Location (**11.2.0** or **12.1.0**), and then click **Next**.

   - Oracle Base—/home/oracle/app/oracle
Installing Cisco IoT FND

- Software Location—/home/oracle/app/oracle/product/11.2.0/dbhome_1
- 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.

8. On the Create Inventory page, keep the default values, and then click Next.
   - Inventory Directory—/home/oracle/app/oracle/product/orainventory
   - orainventory_Group Name—dba

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 (12c only)
   - Data Guard administrative (OSDGDBA) group—dba (12c only)
   - Encryption Key Management administrative (OSKMDBA) group—dba (12c only)

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.

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.

12. Click Next.

13. On the Summary page, verify the database settings, and then click Finish (11g) or Install (12c) to start the installation process.

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.

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

    Note: If performing a new installation of Oracle 12c or upgrading from Oracle 11g, you must install the Oracle 12c Patch 20830993. Go to (Mandatory) Installing 12c Patch.

(Mandatory) Installing 12c Patch

For all new Oracle 12c database installations and all Oracle 11g upgrades, you must install the 12c patch.
Installing Cisco IoT FND

To install the patch:

1. Stop IoT FND application if running.

2. Stop Oracle service if running.

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.*

```bash
/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/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 12c 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 Ball 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
Installing Cisco IoT FND

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 12c 12.1.0.2.0
Oracle Database 12c 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 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
Installing Cisco IoT FND

There are 135 products installed in this Oracle Home.

There are no Interim patches installed in this Oracle Home.

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
Installing Cisco IoT FND

Log file location: /home/oracle/app/oracle/product/12.1.0/dbhome_1/cfgtoollogs/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/logs/opatch2016-02-25_10-48-48AM_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/opatch2016-02-25_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/lisinventory2016-02-25_11-05-19AM.txt
Installing Cisco IoT FND

---

Installed Top-level Products (1):
Oracle Database 12c 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 Bai 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 12c 12.1.0.2.0
Oracle Database 12c 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
<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle JDBC Server Support Package</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle JDBC/OCI Instant Client</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle JDBC/TTHIN Interfaces</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle JFC Extended Windowing Toolkit</td>
<td>11.1.1.6.0</td>
</tr>
<tr>
<td>Oracle JVM</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle JVM For Core</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle LDAP administration</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Locale Builder</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Message Gateway Common Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia Client Option</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia Java Advanced Imaging</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia Locator</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia Locator Java Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Multimedia Locator RDBMS Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Net</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Net Java Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Net Listener</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Net Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Net Services</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Netca Client</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Notification Service</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Notification Service (eONS)</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Notification Service for Instant Client</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle ODBCServer</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle ODBC Driver</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle ODBC Driver for Instant Client</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle OLAP</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle OLAP API</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle OLAP RDBMS Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Partitioning</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Programmer</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Quality of Service Management (Client)</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle R Enterprise Server Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle RAC Deconfiguration</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle RAC Required Support Files-HAS</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Real Application Testing</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Recovery Manager</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Security Developer Tools</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Spatial and Graph</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle SQL Developer</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Starter Database</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Text</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Text ATG Language Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Text for Core</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Text Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Universal Connection Pool</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Universal Installer</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle USM Deconfiguration</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle Wallet Manager</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle XML Development Kit</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Oracle XML Query</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>oracle.swd.oui.core.min</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Parser Generator Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Perl Interpreter</td>
<td>5.14.1.0.0</td>
</tr>
<tr>
<td>Perl Modules</td>
<td>5.14.1.0.0</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>PL/SQL Embedded Gateway</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Platform Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Precompiler Common Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Precompiler Common Files for Core</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Precompiler Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>Precompilers</td>
<td>12.1.0.2.0</td>
</tr>
<tr>
<td>RDBMS Required Support Files</td>
<td>12.1.0.2.0</td>
</tr>
</tbody>
</table>
There are 135 products installed in this Oracle Home.

Interim patches (1):
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

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

### Table 1  Oracle Database Environment Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_BASE</td>
<td>Defines the path to the Oracle root directory on your system. For example:</td>
</tr>
<tr>
<td></td>
<td>$ export ORACLE_BASE=/home/oracle/app/oracle</td>
</tr>
<tr>
<td></td>
<td>If this variable is not set, the IoT FND setup script displays an error.</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>Defines the path to the Oracle home of the IoT FND database. For example:</td>
</tr>
<tr>
<td></td>
<td>$ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not have any trailing backslashes in the ORACLE_HOME environment variable.</td>
</tr>
<tr>
<td>PATH</td>
<td>Defines the path to the Oracle binaries. For example:</td>
</tr>
<tr>
<td></td>
<td>$ export PATH=$PATH:$ORACLE_HOME/bin</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH</td>
<td>Defines the path to the libraries. For example:</td>
</tr>
<tr>
<td></td>
<td>$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>ORACLE_SID</td>
<td>Defines the Oracle System ID (SID).</td>
</tr>
<tr>
<td></td>
<td>If you are only using one database server or installing an HA deployment,</td>
</tr>
<tr>
<td></td>
<td>set this variable on the primary database server to <strong>cgms</strong>:</td>
</tr>
<tr>
<td></td>
<td>$ export ORACLE_SID=cgms</td>
</tr>
<tr>
<td></td>
<td>If deploying a standby database server, set this variable on the standby</td>
</tr>
<tr>
<td></td>
<td>database server to <strong>cgms_s</strong>:</td>
</tr>
<tr>
<td></td>
<td>$ export ORACLE_SID=cgms_s</td>
</tr>
<tr>
<td></td>
<td>If this variable is not set, the IoT FND setup script displays an error.</td>
</tr>
</tbody>
</table>

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

**On a Single or Primary Database Server**

```
$ su - oracle
$ export ORACLE_BASE=/home/oracle/app/oracle
$ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1
$ export PATH=$PATH:$ORACLE_HOME/bin
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
$ export ORACLE_SID=cgms
```

**On a Standby Database Server**

```
$ su - oracle
$ export ORACLE_BASE=/home/oracle/app/oracle
$ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1
$ export PATH=$PATH:$ORACLE_HOME/bin
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
$ export ORACLE_SID=cgms
```

You can also use a .bashrc file to define these variables.

### 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:

1. Log in as the root user.

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

3. Create the cgms directory and download the scripts and templates to it:

   $ cd $ORACLE_BASE/app/oracle
   $ mkdir cgms
   $ cd cgms
   $ cp -R /opt/cgms-oracle/scripts .
   $ cp -R /opt/cgms-oracle/templates .
   $ cp -R /opt/cgms-oracle/tools .
   $ cd ..
   $ chown -R oracle:dba cgms

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: 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

09-13-2012 10:38:07 PDT: INFO: ======== CGMS Database Setup Started =========
09-13-2012 10:38:07 PDT: INFO: Are you sure you want to setup CG-NMS database (y/n)? y
Enter new password for SYS DBA:
Re-enter new password for SYS DBA:
Enter new password for CG-NMS database:
Re-enter new password for CG-NMS database:
09-13-2012 10:38:18 PDT: INFO: Stopping listener ...
09-13-2012 10:38:18 PDT: INFO: Deleting database files ...
09-13-2012 10:38:18 PDT: INFO: Creating listener ...
09-13-2012 10:38:19 PDT: INFO: Configuring listener ...
Installing Cisco IoT FND

To start the IoT FND Oracle database:

1. Run the script:

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

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
- 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
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:

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

Changing the SYS DBA and IoT FND Database Passwords

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

1. On the IoT FND server, run the setupCgms.sh script and change the password for the cgms_dba user.

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

   ```bash
   # cd /opt/cgms/bin
   # ./setupCgms.sh
   ...
   Do you want to change the database password (y/n)? y
   Enter database password:
   Re-enter database password:
   09-13-2012 17:15:31 PDT: INFO: Configuring database password. This may take a while. Please wait ...
   ...
   For information about running the setupCgms.sh script, see Setting Up IoT FND.
   ```

2. On the Oracle server, run the change_password.sh script and change the password for the cgms_dba user:

   ```bash
   $ ./change_password.sh
   Are you sure you want to change CG-NMS database password (y/n)? y
   Enter current password for SYS DBA:
   Re-enter current password for SYS DBA:
   Enter new password for SYS DBA:
   Re-enter new password for SYS DBA:
   Enter new password for CG-NMS database:
   Re-enter new password CG-NMS database:
   User altered.
   ...
   **Note:** As root, you can also use this script to change the password for the sys user (SYS DBA).
3. On the IoT FND server, run the cgms_status.sh script to verify the connection between IoT FND and the IoT FND database:

```
# service cgms status
```

## IoT FND Database Helper Scripts

Table 2 describes helper IoT FND database scripts available in the $ORACLE_BASE/cgms/scripts/ directory:

<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>change_password.sh</td>
<td>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.</td>
</tr>
<tr>
<td>backup_archive_log.sh</td>
<td>Use this script to back up the archive logs.</td>
</tr>
<tr>
<td>backupCgmsDb.sh</td>
<td>Use this script to back up the IoT FND database. This script supports full and incremental backups.</td>
</tr>
<tr>
<td>restoreCgmsDb.sh</td>
<td>Use this script to restore the IoT FND database from a backup.</td>
</tr>
<tr>
<td>startOracle.sh</td>
<td>Use this script to start the IoT FND database.</td>
</tr>
<tr>
<td>stopOracle.sh</td>
<td>Use this script to stop the IoT FND database.</td>
</tr>
<tr>
<td>setupStandbyDb.sh</td>
<td>(IoT FND database HA installations only) Use this script to set up the standby database server.</td>
</tr>
<tr>
<td>setupHaForPrimary.sh</td>
<td>(IoT FND database HA installations only) Use this script to set up the primary database server.</td>
</tr>
<tr>
<td>getHaStatus.sh</td>
<td>Run this script to verify that the database is set up for HA.</td>
</tr>
</tbody>
</table>

## Installing and Setting Up the SSM

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 CiscoJ 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 Setting Up an HSM Client for information on the Hardware Security Module (HSM).

**BEFORE YOU BEGIN**

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:

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%]

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

   - ssm_csmp_keystore password: ciscossmsm
   - csmp alias name: ssm_csmp
   - key password: ciscossmsm
   - 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:

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

      Enter current ssm_csmp_keystore password : ciscossmsm
      Enter alias name : ssm_csmp
      Enter key password : ciscossmsm

      security-module=ssm
      ssm-host=<Replace with IPv4 address of SSM server>
      ssm-port=8445
      ssm-keystore-alias=ssm_csmp
      ssm-keystore-password=NQ1/zokip4gtUe8yQnJuNw==
      ssm-key-password=NQ1/zokip4gtUe8yQnJuNw==

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

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.

6. Start the SSM server:

```
[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.

To uninstall the SSM server:

1. Stop the SSM server:

   `service ssm stop`

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

3. Uninstall the cgms-ssm rpm:

   `rpm -e cgms-ssm`

   Fresh installations only

4. Install a new SSM server.

5. Copy and overwrite the `/opt/cgms-ssm/conf` directory with the contents moved in 2.

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:

1. Stop IoT FND.

   `service cgms stop`

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

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

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==

5. To set up the HSM, specify the following properties in the cgms.properties file (see also, Setting Up an HSM Client):
   
   security-module=ssm/hsm (required; hsm : Hardware Security Module default.)
   hsm-keystore-name=testGroup1 (optional; hsm partition name; testGroup1 default)
   hsm-key-store-password=TestPart1 (optional; encrypted hsm partition password; TestPart1 default)

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

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

BEFORE YOU BEGIN

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:

1. Log in to the RHEL server that will host IoT FND.
2. Installing IoT FND.
3. Setting Up IoT FND.

4. Running the IoT FND Database Migration Script.

5. Checking IoT FND Status.

6. 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.

Installing IoT FND

To install the IoT FND application:

1. Run the IoT FND installation RPM:

   $ rpm –ivh cgms-version.x86_64.rpm

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 Installing Certificates).

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. 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 4.1.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
09-13-2012 17:10:00 PDT: INFO: ========= CG-NMS Setup Started - 2012-09-13-17-10-00 =========
09-13-2012 17:10:00 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

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

Enter database server IP address [example.com]: 128.107.154.246
Enter database server port [1522]:
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522

Enter database SID [cgms]:
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms

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

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

Enter database password:
Re-enter database password:
09-13-2012 17:15:31 PDT: INFO: Configuring database password. This may take a while. Please wait ...

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

Do you want to change the web application 'root' user password (y/n)? n
09-13-2012 17:16:34 PDT: INFO: User response: n

Do you want to change the FTP settings (y/n)? n
09-13-2012 17:16:45 PDT: INFO: User response: n

09-13-2012 17:16:45 PDT: INFO: ========= CG-NMS 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
Installing Cisco IoT FND

Enter database server IP address [example.com]: 128.107.154.246

Enter database server port [1522]:
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522

Enter database SID [cgms]:
09-13-2012 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
- Database System ID (SID), which is cgms for the primary database server

Enter 1522.

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
Enter database server IP address [-]: 128.107.154.20
Enter database server port [-]: 1522
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522
Enter database SID [-]: cgms_s
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms_s
09-13-2012 17:11:18 PDT: INFO: Configuring database settings. This may take a while. Please wait ...

For information about setting up database HA, see Setting Up IoT FND Database for HA.

Configuring the IoT FND Database Password

When prompted to change the IoT FND database password, enter the password of the cgms_dba user 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
Enter database password:
Re-enter database password:
09-13-2012 17:15:31 PDT: INFO: Configuring database password. This may take a while. Please wait ...

Configuring the Keystore Password

To configure the keystore password:

Do you want to change the keystore password (y/n)? y
Enter keystore password: keystore_password
Re-enter keystore password: keystore_password
Installing Cisco IoT FND

09-13-2012 10:21:59 PDT: INFO: Configuring keystore password. This may take a while. Please wait ...

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
09-13-2012 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

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

09-13-2012 17:16:49 PDT: INFO: Configuring FTP settings. This may take a while. Please wait ...
09-13-2012 17:16:57 PDT: INFO: FTP settings configuration completed successfully

Checking IoT FND Status

Before you can start IoT FND, check its connection to the IoT FND database by running this command:

# service cgms status

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.

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:

```
https://nms_machine_IP_address/
```

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

#### 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:

1. From the username drop-down menu (top right), choose Time Zone.
2. Select a time zone.
3. Click Update Time Zone.
4. Click OK.
Installing Cisco IoT FND

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.

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.

Click Hide Filters to close the search 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.

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
- Upgrading IoT FND
- Uninstalling IoT FND

Starting IoT FND

To start IoT FND, run this command:

```
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

To check IoT FND status, run this command:

```
service cgms status
```

Stopping IoT FND

To stop IoT FND, run this command:

```
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 this command:

```
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.
Installing Cisco IoT FND

IoT FND Helper Scripts

Table 3 describes the helper IoT FND scripts in the /opt/cgms/bin/ directory.

<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deinstall_cgms_watchdog.sh</td>
<td>Uninstalls the watchdog script.</td>
</tr>
<tr>
<td>install_cgms_watchdog.sh</td>
<td>Installs the watchdog script.</td>
</tr>
<tr>
<td>mcast_test.sh</td>
<td>Tests the communication between cluster members.</td>
</tr>
<tr>
<td>password_admin.sh</td>
<td>Changes or resets the user password used to access IoT FND.</td>
</tr>
<tr>
<td>print_cluster_view.sh</td>
<td>Prints cluster members.</td>
</tr>
</tbody>
</table>

Upgrading IoT FND

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 before performing this upgrade.

Caution: Run the following steps sequentially.

To upgrade the IoT FND application:

1. Obtain the new IoT FND RPM.
2. Stop IoT FND:
   
   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.
3. Make sure the cgms service has stopped:
   
   service cgms status

4. Upgrade the IoT FND RPM:
   
   rpm -Uvh new_cgms_rpm_filename

   Note: These files overwrite the files in /opt/cgms.
5. Run the database migrations to upgrade the database from the /opt/cgms directory:
   
   cd /opt/cgms/bin
   ./db-migrate

   Note: You must run the db-migrate script after each upgrade.
6. When prompted, enter the database password. The default password is cgms123.
7. Start IoT FND:
   
   # service cgms start

You can also use the RHEL (Red Hat Enterprise Linux) GUI to start the IoT FND service (ADMIN > System Management > Server Settings > Services). For information, see the RHEL documentation.
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.

Installing and Configuring the IoT FND TPS Proxy

The first use of the optional TPS proxy is typically when a CGR 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 (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).

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.

Setting Up the TPS Proxy

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 `cgms-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`
To configure the proxy-server settings:

1. Configure a RHEL server to use as the TPS proxy.

2. Connect this RHEL server so that it can be reached while outside the firewall.

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 cgms.properties and tpsproxy.properties files after running the encryption_util.sh script during IoT FND TPS Proxy Enrollment.

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 cgms.properties and tpsproxy.properties-template files for HTTPS outbound tunnel provisioning requests so that IoT FND recognizes them as requests from the TPS proxy.

### 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 cgms.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 *Generating and Exporting Certificates* in the Generating and Installing Certificates chapter of this guide.

To enroll the terminal TPS proxy:

1. Create a `cgms_keystore` file.
2. Add your certifications to this file.
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:

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.

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:
   ```
   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:

```
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`
Installing Cisco IoT FND

The following entry in the TPS proxy tpsproxy.log file defines inbound requests for a CGR:


This message entry in the TPS proxy tpsproxy.log file indicates that the TPS successfully forwarded the message to IoT FND:


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:JSJ1520047) 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:

Back 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 Full IoT FND Backup

Before You Begin

Before backing up your IoT FND database:

1. Download and install the latest cgms-oracle-version_number.x86_64.rpm package.
2. Copy the scripts, templates, and tools folders from the /opt/cgms-oracle folder to the $ORACLE_BASE/cgms folder.
3. 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.
Installing Cisco IoT FND

**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:

1. On the IoT FND database server, open a CLI window.
2. Switch to the user oracle:
   ```
   su - oracle
   ```
3. Change directory to the location of the IoT FND backup script (backupCgmsDb.sh):
   ```
   cd /home/oracle/app/oracle/cgms/scripts
   ```
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/bkp
   ```
   Are you sure you want to backup CG-NMS database (y/n)? y
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.

1. On the IoT FND database server, open a CLI window.
2. Switch to the user oracle:
   ```
   su - oracle
   ```
3. Change directory to the location of the IoT FND backup script (backupCgmsDb.sh):
   ```
   cd /home/oracle/app/oracle/cgms/scripts
   ```
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
   ```

**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 an 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:

1. On the IoT FND database server, open a CLI window.
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
   ```
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
   ```

### 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 11.2.0.1 can only be restored on v11.2.0.1 if using the supplied scripts. Backups do not work across different versions of Oracle, for example, a backup taken on 11.2.0.1 cannot be restored on 11.2.0.3 using the supplied scripts. If a database upgrade from 11.2.0.1 to 11.2.0.3 is required, follow the Oracle upgrade procedure. Refer to the Oracle upgrade document and Web site.

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 or a higher 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:

1. Stop IoT FND.
   ```
   service cgms stop
   ```
2. Switch to the user oracle, change directories to the script location, and stop Oracle:
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:

   06-08-2012 13:12:56 PDT: INFO: Import completed successfully
   06-08-2012 13:12:56 PDT: ERROR: Insufficient shared memory file system. Increase your shared memory file system before restoring this database.
   06-08-2012 13:12:56 PDT: ERROR: ========== CGMS Database Restore Failed ==========
   06-08-2012 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

4. Start Oracle:

   ./startOracle.sh

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.

6. Start IoT FND:

   service cgms start

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
Deploying IoT FND/Oracle/TPS Virtual Machines on ESX 5.x

You use the VMware vSphere client to import OVA files into ESXi 5.x.

BEFORE YOU BEGIN

- Install the VMware vSphere Client for the ESXi 5.x server.
- Locate the VMware ESXi 5.x credentials to create virtual machines in ESXi 5.x.
- Ensure that you meet the VMware server machine requirements.

Listed below are the VM CPU and memory requirements for a small scale deployment:

**NMS OVA**
- 16 GB memory
- 1 core and 4 virtual sockets
- 150 GB of virtual storage

**Oracle OVA**
- 24 GB of memory
- 2 virtual sockets with 2 cores per socket
- 300 GB of virtual storage

**TPS OVA**
- 4 GB of memory
- 1 virtual socket with 1 core
- 50 GB of virtual storage

DETAILED STEPS

To import the IoT FND, Oracle, and TPS virtual appliances into ESXi 5.x or ESXi 6.x using VMware vSphere Client version 5.0.0 or 6.0.0, respectively:

1. Select the **Network Adapter > NAT** setting for the server.

2. At the VMware vSphere Client window enter the IP address, username and password of the server where VMware resides. Click **Login**.
3. Select **File > Deploy OVF Template...**

4. Browse to the FND-OVA\oracle-415.ovf file.

5. Ensure that the correct OVA file displays in the Source location window, and then click **Next**. *(Note: The Next button, not shown, is found in the bottom, right-hand portion of the window.)*

6. In the OVF Template Details window, enter the Name and Inventory Location of the deployed template (for example, FND-Oracle-12c-VM) Click **Next**.

7. In the next window that appears, verify the OVA file name matches what you entered in the previous window. If no issues, click **Next**. If the OVA file name does **not** match, click **Back** and reenter the information.
8. Select a Storage destination for the VM deployed template.

9. In the Disk Format window (not shown), select the Thin Provision option, and then click Next.

   Note: Thin Provision allows the VM disk to grow as needed.

10. In the Ready to Complete window, confirm your deployment settings, and then click Finish. Deployment of the FND-Oracle-12C-VM templates begins.

11. After completion of the FND-Oracle-12c-VM install, a window displays.
    - In the left panel, select the FND-Oracle-12c-VM server.
    - In the right panel, Under Basic Task, select Edit virtual machine settings to confirm CPU and MEM settings.
    - Click OK to close the window.
12. Click **Power on the virtual machines** and select **Console** tab in the window that opens. The console opens showing possible RHEL server versions to select.

13. After you select a RHEL option, you will be prompted for your username and password.

14. Click **Sign In** and select the **GNOME Classic** option.
15. Click Sign In.

16. At the Welcome screen, select the default language (such as English–United States) for the interface, Click Next.

17. Continue through the install script until you reach the final screen that indicates that “Your computer is ready to use.” Click Start using Red Hat Enterprise Linus Server button.

18. Close Getting Started window, right-click to open.

19. At the Application Places window, right-click in the window to display a menu panel. Select Open in Terminal.
Generating and Installing Certificates

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

- Information About Certificates
- Generating and Exporting Certificates
- Installing the Certificates
- Configuring IoT FND to Access the Keystore
- Configuring the TPS Proxy to Access the Keystore
- Setting Up an HSM Client
- Configuring the HSM Group Name and Password

Information About Certificates

The following topics provide information on certificates:

- Role of Certificates
- Keystore

Role of Certificates

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

For details on generating certificates for CGRs, refer to the 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.

Keystore

The Keystore provides details for a specific system (such as IoT FND or the TPS proxy) and includes the following items:

- The certificate for that system (such as the IoT FND certificate or TPS proxy certificate)
- The private key for the system
- The certificate chain (path to the CA or subCA)

The IoT FND key and certificates are stored in the cgms_keystore file on the IoT FND server in the /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
- Enabling a Certificate Template
- Generating Certificates for IoT FND and the IoT FND TPS Proxy
- Command Authorization Support
- Configuring a Custom CA for HSM
- Configuring a Custom CA for SSM
- Exporting the CA Certificate

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:

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.

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

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

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


6. Click OK.

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.

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

   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.

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.

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.

11. Click the NMS Properties > Cryptography tab, and retain the following default settings:
    - Algorithm name: RSA
    - Minimum key size: 2048
    - Cryptographic provider: Requests can use any provider available on the subject computer
    - Request hash: SHA256

12. Click OK.

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

14. Click OK.

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

Enabling a Certificate Template

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

To enable the certificate template:

1. Configure a certificate template (see Configuring a Certificate Template for IoT FND and the IoT FND TPS Proxy).
2. Open the Certificate Authority application on the Windows Server.
3. Expand the menu to view the Certificate Templates folder.
4. Right-click Certificate Templates and choose New > Certificate Template to Issue from the context menu.
5. In the Enable Certificate Templates window, highlight the new NMS template.
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.

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

Tip: In 9.b 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:

1. Configure a certificate template (see Configuring a Certificate Template for IoT FND and the IoT FND TPS Proxy).
2. Enable the certificate template (see Enabling a Certificate Template).
3. From a server running Windows Server 2008, choose Start > Run and enter mmc to open the MMC console.
4. In the Console 1 window, expand the Certificates > Personal folders.
5. Right-click Certificates and choose All Tasks > Request New Certificate from the context menu.
6. In the Before You Begin window, click Next.
8. In the Request Certificates window, do the following:
   a. Check the NMS check box.
   b. Click the More information... link.
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.
10. Click **Apply**. Click **OK**.

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

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

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

14. Choose **Personal > Certificates**.

15. In the Issued To pane, right-click the new certificate and choose **All Tasks > Export** from the context menu. The Export Wizard window appears.
16. Initiate the Export Wizard.

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

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**.

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.

20. Click **Next**.

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

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.

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.

---

**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.
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.

This section discusses the following topics:

- Enabling Command Authorization Using NMS/TPS Certificates
- Adding an OID Value to the CA Certificate
- Renewing Certificates

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.

1. Generate new NMS/TPS certificates (see Generating Certificates for IoT FND and the IoT FND TPS Proxy) or renew the existing NMS/TPS certificate (see Renewing Certificates).
2. Add an OID value to the CA certificate (see Adding an OID Value to the CA Certificate).
3. Generate a new .pfx file for the NMS/TPS certificate (see Generating Certificates for IoT FND and the IoT FND TPS Proxy).
4. Stop IoT FND (see Stopping IoT FND).
5. Rename the existing cgms_keystore file (for example, cgms_keystore_no_oid).
6. Export the .pfx file to IoT FND and create a new cgms_keystore file (see Using Keytool to Create the cgms_keystore File).
7. Install the new certificates (see Installing the Certificates).
8. Add the new cgms_keystore file to IoT FND (see Copying the cgms_keystore File to IoT FND).
9. Start IoT FND (see Starting IoT FND).
10. Register the routers with IoT FND.

Adding an OID Value to the CA Certificate

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:

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
   ```

2. Restart the CA.

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

4. In the Request Certificates window, do the following:
   a. Check the NMS check box.
   b. Click the More information... link.
5. In the Certificate Properties window, click the **Subject** tab and complete the fields.

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

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

   1.3.6.1.4.1.9.21.3.3.1

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

   01

9. Click **Add**.
Generating and Installing Certificates

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

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

Renewing Certificates

To renew certificates and add the OID value:

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`

2. Restart the CA server.

3. Open the certificate console in the MMC.

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

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.

6. In the Certificate Enrollment window, click **Next**.
7. Click **Details**.

8. Click **Properties**.

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

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

11. At Request Certificates panel, click **Enroll**.
12. Click **Finish**.

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.

**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).

To configure a custom CA for generating HSM certificates:

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

2. Generate a keypair on the HSM and export a CSR for that keypair (see Keystore).

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

   ```bash
   [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 curve type [1] NISTP 192
   [2] NISTP 224
   [3] NISTP 256
   [5] NISTP 521
   Enter curve type [1] NISTP 192
   [2] NISTP 224
   ```
Generating and Installing Certificates

[3] NISTP 256  <-- Choose option 3
[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
CPNhbiBKb3NlMRowGAYDVQQKExFDaXNjbyBTeXN0ZW1zIEluYzEPMA0GA1UECxMG
SW9UU1NRMjQwZzQ5MjB3TmVyc29uZGVyb2Rhbi1TVVNLPSNMMNkZwYXHkoZIzJ0D
AQcQDkQ8dFsd1zrcjVzN3Nyej9trr115qd0w55du8Vj2a17JAP/vVPoU0Yiw/uXwD6+
bb8vq3W2iA67mqRbj+FU6g3Bm/tvCqAAMc6BYygGSM49BARFAAN1ADBPA1EArIoYO
q3dHA2LjIZGwUO1VYys642Nkb484qyBoUZIGsCIFs017UyGQreM1BaSDEPHARZ
RvPlrKo/Zi3c8O4zgFZW
-----END NEW CERTIFICATE REQUEST-----

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

   Note: Ensure that the certificate is signed for 30 years. Mesh nodes reject any certificate signed for less than 30
   years. You can use the root CA that is used for 802.1x authentication for node admission.

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
Generating and Installing Certificates

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

5. Configure IoT FND to use this new certificate.

```
[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=2bVWZaq+vsq94YxuAKdaag== <--- encrypted password for the partition
```

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

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

```
Certificate for CSMP
```

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.

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).

To configure a custom CA for generating SSM certificates:

1. Stop the ssm service.

   [root@nms-rhel-6-6 ~]# stop ssm

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

   1. Generate a new keyalias with self signed certificate for CSMP
   2. Generate a new keypair & certificate signing request for CSMP  <--- Choose option 2
   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: 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
   Successfully 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]#
3. Save the generated CSR to your CA and sign the certificate.

   **Note:** Ensure that the certificate is signed for 30 years. Mesh nodes reject any certificate signed for less than 30 years. You can use the root CA that is used for 802.1x authentication for node admission.

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

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

   ```bash
   [root@nms-rhel-6-6 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  <-- Choose option 3
   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 : 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
   Successfully imported certificate into alias root
   
   6. Use the `ssm_setup.sh` script to import the signed certificate for the alias:

   ```bash
   [root@nms-rhel-6-6 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  <-- Choose option 3
   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 : 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
Successfully imported certificate into alias ssmcustomca

7. Update cgms.properties file with 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==

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

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:

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

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

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

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

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

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

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

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

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

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

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

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.
Generating and Installing Certificates

- **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:

**BEFORE YOU BEGIN**
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**
- **Copying the cgms_keystore File to IoT FND**
- **Importing the CA Certificate**
- **Installing Custom Browser Certificates**

### Using Keytool to Create the cgms_keystore File

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

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

   ```bash
   [root@tps_server ~]# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
   
   Note: Viewing the .pfx provides the Alias Name required during the import.
   
2. Enter the keystore password when prompted.
   
   This is the same password entered when creating the .pfx file.

3. The information that displays (see the following Example) includes the alias_name needed for 3.

   ```bash
   keytool -importkeystore -v -srckeystore filename.pfx -srcstoretype pkcs12
   -destkeystore cgms_keystore -deststoretype jks -srcalias alias_name
   -destalias cgms -destkeypass keystore_password
   
   4. At the prompt, enter the destination keystore password.

   5. Re-enter the keystore password when prompted.

   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.
Generating and Installing Certificates

# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
Enter keystore password: keystore
Keystore type: PKCS12
Keystore provider: SunJSSE
Your keystore contains 1 entry
Alias name: le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b
Creation date: March 29, 2012
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
...Storing cgms_keystore

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_password
...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:

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

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:

1. On the IoT FND application server, log in as root.
2. Change directory to /opt/cgms/server/cgms/conf, where you have placed the cgms_keystore file:

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

3. Import the CA certificate:

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

   A script displays on the screen.

4. Enter the keystore password when prompted.
5. Re-enter the password.
6. Enter yes when prompted to trust the certificate.
Generating and Installing Certificates

The certificate is added to the Keystore.

**Example**

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

```bash
# 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
Certificate fingerprints:
  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
  KeyCertSign
  Crl_Sign
  #3: ObjectId: 2.5.29.14 Criticality=false
  SubjectKeyIdentifier [KeyIdentifier [0000:C6 AB 3B 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 the IoT FND TPS Proxy Keystore

Follow the same steps as in Importing the CA Certificate to import the CA certificate into cgms_keystore on the IoT FND TPS proxy.

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
- Importing Custom Certificates with the North Bound API Client (Windows)
- Importing Custom Certificates with Windows IE
- Managing Custom Certificates
Generating and Installing Certificates

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** 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

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

2. Delete the existing jbossas.keystore and jbossas.keystore.password files from the /opt/cgms/server/cgms/conf/ directory.

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

   ```bash
   #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, 2016
   Entry type: PrivateKeyEntry
   Certificate chain length: 2
   Certificate[1]:
   ...```

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:

   ```bash
   # 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]
   
   5. (Optional) Define a **salt**.

   **Note:** If salt is unchanged, then you can skip this step.
Generating and Installing Certificates

The salt defines the strength of the encrypted password and must be at least 8 characters long. For example: A1aB2b2C3c3D4dE5e5F6f6G7g7H8h8I9i9J10j10K11k11L12l12M13m14n14O15

a. Copy the file /opt/cgms/server/cgms/deploy/security-service.xml to a safe location.


NOTE: Select either Step 6 or Step 7 below, based on the NMS release you are running.

6. CG-NMS Releases earlier than 2.1.0 store the keystore password in the following file: /opt/cgms/server/cgms/conf/jbossas.keystore.password

This step encrypts the password that will be stored in the jbossas.keystore.password file.

The password is used to open the jbossas.keystore that has the new custom certificate imported in Step 4.

a. Run /opt/cgms/bin/encrypt-password.sh script with the following parameters:

- Specify the new salt defined in step 5, or use the existing one in the /opt/cgms/server/cgms/deploy/security-service.xml file.
- Set count to 1024.
- Set the password file to jbossas.keystore.password.
- Set your_keystore_password.

#./encrypt-password.sh A1aB2b2C3c3D4dE5e5F6f6G7g7H8h8I9i9J10j10K11k11L12l12M13m14n14O15 1024 jbossas.keystore.password your_keystore_password

b. Move or copy the jbossas.keystore.password to the /opt/cgms/server/cgms/conf directory.

c. Go to Step 8.

7. CG-NMS releases later than 2.1.0 or IoT FND 3.0 release or later, store the keystore password in the /opt/cgms/server/cgms/conf/VAULT.dat file

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

a. Backup the VAULT.dat and vault.keystore files in /opt/cgms/server/cgms/conf to a safe location.

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

#./opt/cgms/bin/vault.sh -k /opt/cgms/server/cgms/conf/vault.keystore -p cgms123 -e /opt/cgms/server/cgms/conf -i 50 -s 12345678 -v vault -b your_keystore_password -a password -x your_keystore_password

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. The default password to open vault.keystore is cgms123.

8. Restart IoT FND:

# service cgms restart

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

10. Accept and add the new certificates.

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

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.
2. In the Security Alert window, click OK.
3. In the security certificate warning window, click the Continue to this Website (Not Recommended) link.
4. In the Security Alert window, click OK.
5. Click the Certificate error section of the address bar.

   ![Certificate error section](image)

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

   ![Certificate Invalid window](image)

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

9. Click **Install Certificate**.

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

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

13. Click **Next**.

14. Click **Finish**.

15. Click **OK**.

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

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

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**.

19. Click **Next**.

20. Click **Finish**.
21. Click OK.

22. In the Certificate window, click OK.

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).

24. Close and restart the browser.

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

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

   - In the /opt/cgms/server/cgms/conf/ directory:
     - jbossas.keystore.password
     - jbossas.keystore
   - 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.
   - In the /opt/cgms/server/cgms/conf directory:
     - VAULT.dat
     - vault.keystore

2. Perform the IoT FND upgrade or new installation (see Upgrading IoT FND).

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 will send events. IoT FND accepts SSL certificates and handshakes published by the NB API client.

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:

1. Stop IoT FND.
2. Run the setupCgms.sh script:

```bash
cwd
/opt/cgms/bin
./setupCgms.sh
```

```
Are you sure you want to setup CG-NMS (y/n)? y
... 
Do you want to change the keystore password (y/n)? y
Enter keystore password: keystore_password
Re-enter keystore password: keystore_password
06-12-2012 10:21:59 PDT: INFO: Configuring keystore password. This may take a while.
Please wait ...
06-12-2012 10:22:00 PDT: INFO: Keystore password configured.
...
This script saves the password set in the cgms.properties file.
```

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:

1. Change to the tpsproxy bin directory:

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

2. Convert your chosen password into encrypted form:

   ```bash
./encryptionUtil.sh {your chosen password for cgms_keystore}
```

   ```
7jlXPniVpMvat+TrDWqh1w==
   ```

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

   a. Open the file for editing.

      ```bash
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==”.

4. Restart TPS proxy:

   ```bash
service tpsproxy restart
   ```

### Setting Up an HSM Client

Complete the following procedures to set up the HSM client:
Installing an 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 an 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 an 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:

1. Get the HSM client package, unpack it, and run the installation script:
   
   sh install.sh

2. Change to the /usr/lunasa/bin directory:
   
   cd /usr/lunasa/bin/

3. Create the client certificate:
   
   ./vtl createCert -n ip_address_of_hsm_client

4. Download the HSM certificate from the HSM server:
   
   scp admin@ip_address_of_hsm_server:server.pem .

5. Upload the client certificate to the HSM server:
   
   scp ../cert/client/ip_address_of_hsm_client.pem admin@ip_address_of_hsm_server: .

6. Load the HSM certificate:
   
   vtl addServer -n ip_address_of_hsm_server -c server.pem .

7. Ensure that the HSM server is added:
   
   vtl listServer

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 2012 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:>

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
b. List the clients 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.

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
```

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 2012 at 20:57:53)

*** CHRYSTOKI DEMO - SIMULATION LAB ***

Status: Doing great, no errors (CKR_OK)
```

### TOKEN FUNCTIONS

- Open Session
- Close Session
- Login
- Logout
- Change PIN
- Init Token
- Get Info
- Slot Info
- Token Info
- Session Info
- Get Slot List
- Wait for Slot Event
- Factory Reset
- CloneMofN

### OBJECT MANAGEMENT FUNCTIONS

- Create object
- Copy object
- Destroy object
- Object size
- Get attribute
- Set attribute
- Find object
- Display Object

### SECURITY FUNCTIONS

- Encrypt file
- Decrypt file
- Sign
- Verify
- Hash file
- Simple Generate Key
- Digest Key

### HIGH AVAILABILITY RECOVERY FUNCTIONS

- HA Init
- HA Login

### KEY FUNCTIONS

- Wrap key
- Unwrap key
- Generate random number
- Derive Key
- PBE Key Gen
- Create known keys
- Seed RNG
- EC User Defined Curves

### CA FUNCTIONS

- Set Domain
- Clone Key
- Set MofN
- Generate MofN
- Activate MofN
- Generate Token Keys
- Get Token Cert
- Sign Token Cert
- Generate CertCo Cert
Generating and Installing Certificates

(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
Generating and Installing Certificates

(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) SIMKxtract           (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

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
Generating and Installing Certificates

(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
Generating and Installing Certificates

OFFBOARD KEY STORAGE:
(97) Set App ID     (98) Options      (100) LKM Commands
(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 an 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 an HSM HA client:

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

2. Change to the /usr/safenet/lunaclient/bin/ directory:

   /usr/safenet/lunaclient/bin/

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 (10.), 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
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.)

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
```

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.

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:

1. Edit the `cgms.properties` file to add the following properties:
Generating and Installing Certificates

- hsm-keystore-name <name>
- 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.

2. Save the cgms.properties file.

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

   ```
   service cgms start
   ```
Managing Tunnel Provisioning

This section describes how to configure IoT FND for tunnel provisioning and how to manage and monitor tunnels connecting FARs (CGRs and C800s) and HERs. This section includes the following topics:

- Overview
- Configuring Tunnel Provisioning
- Monitoring Tunnel Status
- Reprovisioning CGRs

Overview

IoT FND sends the commands generated from processing the tunnel provisioning templates to FARs and HERs to provision secure tunnels between them. The default IoT FND templates contain CLI commands to set up and configure GRE and IPsec tunnels. One HER can serve up to 500 FARs, which may include multiple tunnels with the same HER EID and name.

Figure 1  Tunnels Connect FARs and their Corresponding HERs

To provision tunnels between HERs and FARs, IoT FND executes CLI tunnel configuration commands on these devices. By default, IoT FND provides basic tunnel configuration templates containing the CLI tunnel configuration commands. You can also use your own templates. Although the tunnel provisioning process is automatic, you must first complete the configuration steps outlined in Tunnel Provisioning Configuration Process. After that, whenever a FAR comes online, IoT FND automatically provisions it with a tunnel. Before you configure IoT FND for tunnel provisioning, ensure that the IoT FND TPS Proxy is installed and running.

ZTD without IPsec

Beginning with IoT FND Release 3.1.x, you have the option to initiate ZTD with no IPsec configured by ensuring that the Tunnel Provisioning Template is empty of any CLI. This initial approach of bringing up your network without a factory configuration does not preclude subsequent use of IPsec in your network.
Tunnel Provisioning Configuration Process

You must generate the keystore files on the IoT FND and TPS Proxy before configuring tunnel provisioning. Then, you configure IoT FND and the TPS Proxy to talk to one another (Setting Up the TPS Proxy and Configuring IoT FND to Use the TPS Proxy).

To configure IoT FND for tunnel provisioning:

1. Configure the DHCP servers.
   - Configure DHCP servers to provide unique IP addresses to IoT FND. The default IoT FND tunnel provisioning templates configure a loopback interface and the IP addresses required to create the tunnels.
   - Cisco IOS CGRs use FlexVPN. Ensure that the template only contains addresses for the loopback interface.

2. Configure the tunnel settings.
   - Configure the NMS URL and the DHCP proxy client settings on the Provisioning Settings page in IoT FND (Admin > System Management > Provisioning Settings).

3. (CG-OS CGRs) Configure IoT FND to accept FAR registration requests on first contact (call home) to request tunnel provisioning.
   - Cisco IOS CGRs use the CGNA service.

4. Configure HER management.
   - Configure HERs to allow management by IoT FND using NETCONF over SSH.

5. Add HERs to IoT FND.

6. Review the IoT FND tunnel provisioning templates to ensure that they create the correct type of tunnel.

7. (Optional) If you plan to use your own templates for tunnel provisioning, create one or more tunnel provisioning groups and modify the default tunnel provisioning templates.

8. (CG-OS CGRs) Configure FARs to call home.
   - Configure FARs to contact IoT FND over HTTPS through the IoT FND TPS proxy.

9. Add FARs to IoT FND.
   - Import the FARs into IoT FND using the Notice-of-Shipment XML file.

10. Map FARs to their corresponding HER.

Notes
- Configuring the DHCP Server for Tunnel Provisioning.
- Configuring HERs Before Adding them to IoT FND.
- Adding HERs to IoT FND.
- Configuring Tunnel Provisioning Templates
- This step is typically performed at the factory where the FARs are configured to contact the TPS Proxy.

After completing the previous steps, deploy the FARs and power them on. Tunnel provisioning happens automatically.
This is the sequence of events after a FAR is turned on:

1. Upon joining the uplink network after being turned on, the FAR sends a request for certificate enrollment.

2. The FAR then requests tunnel provisioning to IoT FND through the IoT FND TPS Proxy.

3. IoT FND looks up the FAR record in the IoT FND database and determines which tunnel provisioning templates to use. IoT FND also looks up which HERs to which to establish a tunnel.

4. For Cisco IOS CGRs, the default templates configure the CGR to use FlexVPN. The FlexVPN client is configured on the CGR that will contact the HER and ask for a FlexVPN tunnel to be dynamically constructed. This is how the HER dynamically adds a new tunnel endpoint interface for the CGR.

5. Before processing FAR templates, IoT FND processes the HER Tunnel Deletion template and sends the resulting commands to the HERs. This is done for each HER to remove existing tunnel configuration that may be associated with the FAR.

6. IoT FND uses the FreeMarker template engine to process the FAR Tunnel Addition template. The engine converts the templates to text, which IoT FND assumes to be CLI configuration commands (CG-OS or Cisco IOS, per the CGR). IoT FND uses these commands to configure and bring up one end of the tunnel on the FAR.

7. IoT FND uses the FreeMarker template engine to process the HER Tunnel Addition template. The engine converts the templates to text, which IoT FND assumes to be commands for configuring the tunnel on the HERs.

8. This step is OS-specific:
   - For Cisco IOS CGRs, if no errors occurred applying the commands generated by the templates to the FAR and HERs, IoT FND configures a new active CGNA profile “cg-nms-register,” and deactivates the cg-nms-tunnel profile. That cg-nms-register profile uses the IoT FND URL.
   - For CG-OS CGRs, IoT FND re-configures the call home URL to the IoT FND URL specified in the Provisioning Settings page (ADMIN > System Management > Provisioning Settings).
Managing Tunnel Provisioning

The specified URL uses the IoT FND registration port (default 9121) instead of the tunnel provisioning port. The Fully Qualified Domain Name (FQDN) in that URL is different and resolves to an IP address that is only reachable through the tunnels.

Configuring Tunnel Provisioning

This section describes how to configure IoT FND for tunnel provisioning.

- Configuring the DHCP Server for Tunnel Provisioning
- Configuring DHCP for Tunnel Provisioning Using CNR

Configuring the DHCP Server for Tunnel Provisioning

For tunnel provisioning to succeed, configure the DHCP server used by IoT FND to supply addresses to create tunnels between the FARS and HERs. For example, configure the DHCP server to provide IP addresses for tunnel provisioning on a permanent-lease basis.
Managing Tunnel Provisioning

IoT FND makes the DHCP requests based on the settings defined in the tunnel provisioning templates. During tunnel provisioning, the IoT FND templates can make two kinds of DHCP requests:

- Request an IP address, and then make it available to the template.
- Request a subnet with two IP addresses, and then make both addresses available to the template.

IoT FND can make these requests for IPv4 addresses and IPv6 addresses.

The ability to request DHCP addresses from the template gives you maximum flexibility when defining tunnel configurations because you allocate the exact address needed for each FAR and corresponding interface on the HER. The default tunnel provisioning templates provided address the most common use case: one IPsec tunnel between the FAR and its corresponding HER. Each end of this IPsec tunnel gets a dynamically allocated IPv4 address:

- If your DHCP server supports subnet allocation, use it to obtain two addresses that belong to the same subnet.
- If your DHCP server only supports address allocation, configure it so that the two DHCP address requests return addresses that can be used as ends of an IPsec tunnel.
- If your routing plan calls for allocating unique IPv4 addresses for each FAR and assigning it to a loopback interface above the IPsec tunnel, allocate this address using the IoT FND template.

If you choose to build IPv6 GRE tunnels, allocate the IPv6 addresses for each end of the tunnel using DHCP prefix delegation or individual address requests.

This section describes example DHCP settings for tunnel provisioning. How you configure these settings depends on your installation. This section provides general guidelines for configuring the DHCP server for tunnel provisioning using the Cisco Network Registrar (CNR).

Configuring DHCP for Tunnel Provisioning Using CNR

The CNR CLI script in the following example configures the CNR DHCP server to service requests made by the default tunnel provisioning templates in IoT FND. When using this script, ensure that the subnets are appropriate for your DHCP server environment.

**Example CNR DHCP Server Tunnel Provisioning Script**

```bash
# These commented out commands support re-applying the configuration by first
# removing any previously applied configuration, in reverse order. This should
# not be done in a production environment, but may be useful when initially
# developing and testing a configuration.

# scope v4address-perm delete
# dhcp-address-block v4subnet-perm delete
# prefix v6subnet-perm delete
# prefix v6address-perm delete
# policy permanent delete

# Configure the server to automatically map any IPv4 or IPv6 user class
# option values to selection tags. By default CG-NMS includes a value of
# "CG-NMS" for the user class in its requests. The tag is used to insure
# prefixes and scopes configured to satisfy requests from CG-NMS are only
# used for that purpose.

dhcp set map-user-class-id=append-to-tags

# Since CG-NMS uses the leased addresses and subnets in router
# configuration the addresses and subnets must be permanently allocated
# for that purpose. Create a policy that instructs the DHCP server to
# offer a permanent lease.
```

83
policy permanent create
policy permanent set permanent-leases=enabled

# Configure DHCPv6.

# The default CG-NMS tunnel template will request IPv6 addresses for
# use with CGR loopback interfaces.
prefix v6address-perm create 2001:DB8:0:0:1::/80 dhcp-type=dhcp
prefix v6address-perm set description="Pool for leasing addresses for loopback interfaces."
prefix v6address-perm set policy=permanent
prefix v6address-perm set selection-tags=CG-NMS

# The default CG-NMS tunnel template will request IPv6 prefixes for
# use with GRE tunnels. Force use of a /127 prefix.
prefix v6subnet-perm create 2001:DB8:0:0:2::/80 dhcp-type=prefix-delegation
prefix v6subnet-perm set description="Pool for leasing prefixes for GRE tunnels."
prefix v6subnet-perm set policy=permanent
prefix v6subnet-perm set selection-tags=CG-NMS
prefix-policy v6subnet-perm set default-prefix-length=127
prefix-policy v6subnet-perm set shortest-prefix-length=127

# Configure DHCPv4.

# The default CG-NMS tunnel template will request IPv4 subnets for
# use with IPsec tunnels. Note that currently address pools for
# IPv4 subnet allocation can only be configured using the CLI as the
# CNR Web UI does not currently support them.

# If CNR allowed you to set a description on DHCP address blocks it would be:
# "Pool for leasing subnets for IPsec tunnels."
dhcp-address-block v4subnet-perm create 192.0.2.0/24
dhcp-address-block v4subnet-perm set default-subnet-size=31
dhcp-address-block v4subnet-perm set policy=permanent
dhcp-address-block v4subnet-perm set selection-tags=CG-NMS

# The default CG-NMS tunnel template will request IPv4 addresses for
# use with loopback interfaces.
scope v4address-perm create 198.51.100.0 255.255.255.0
scope v4address-perm set description="Pool for leasing addresses for loopback interfaces."
scope v4address-perm set policy=permanent
scope v4address-perm addRange 198.51.100.2 198.51.100.254
scope v4address-perm set selection-tag-list=CG-NMS

# Configure detailed logging of incoming and outgoing packets. This is useful when
# debugging issues involving DHCP, however this level of logging will lower the
# performance of the DHCP server. If this is a production server under heavy load
# it may be necessary to forgo detailed packet logging.

dhcp set
log-settings=missing-options,incoming-packet-detail,outgoing-packet-detail,unknown-criteria,client-deta
il,client-criteria-processing,dropped-waiting-packets,v6-lease-detail

# Save the changes and reload the server to have them take effect.
save
dhcp reload

# List the current configuration.
policy list
prefix list
dhcp-address-block list
Managing Tunnel Provisioning

scope list
dhcp show

Configuring Tunnel Group Settings

You use groups in IoT FND to bulk configure tunnel provisioning. By default, all FARs are added to the appropriate default group (default-cgr, default-c800). Default groups contain the templates used for tunnel provisioning.

Topics in this section include the following:

- Creating Tunnel Groups
- Deleting Tunnel Groups
- Viewing Tunnel Groups
- Moving FARs to Another Group
- Renaming a Tunnel Group

Creating Tunnel Groups

If you plan to use one set of templates for all FARs, whether using the default templates, modified default templates or custom templates, do not create additional groups. To define multiple sets of templates, create groups and customize the templates for these groups.

Note: CGRs and C800s can be in the same tunnel provisioning group if your custom templates are applicable to both router types.

To create a tunnel group:

1. Choose CONFIG > Tunnel Provisioning.

2. Click + icon in left pane to add a group.

3. Enter a name of the new group, and then click OK.

   The group appears in the Tunnel Groups pane.

After creating a tunnel group, the next step is to move FARs from other groups to it, as described in Moving FARs to Another Group.

Deleting Tunnel Groups

Only empty groups can be deleted. Before you can delete a tunnel group, you must move the devices it contains to another group.

To delete an empty tunnel group:

1. Choose CONFIG > Tunnel Provisioning.

2. In the TUNNEL GROUPS left pane, select the tunnel group to delete.

3. Click ( seaside) to delete the group.

4. Click Yes to confirm deletion.
Managing Tunnel Provisioning

Viewing Tunnel Groups

The Tunnel Provisioning page lists information about existing tunnel groups.

Follow these steps to view the tunnel groups defined in IoT FND:

1. Choose CONFIG > Tunnel Provisioning.
2. Click Group Members tab.
3. In the TUNNEL GROUPS pane (left), select a group.

IoT FND displays the following Tunnel Group information for each router in the group. Not all routers support all fields. (See Table 1).

Table 1  Tunnel Group Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Router EID (device identifier).</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the router:</td>
</tr>
<tr>
<td></td>
<td>- Unheard—The router has not contacted IoT FND yet.</td>
</tr>
<tr>
<td></td>
<td>- Unsupported—The router is not supported by IoT FND.</td>
</tr>
<tr>
<td></td>
<td>- Up—The router is in operation.</td>
</tr>
<tr>
<td></td>
<td>- Down—The router is turned off.</td>
</tr>
<tr>
<td>Last Heard</td>
<td>Last time the router contacted or sent metrics to IoT FND. If the router</td>
</tr>
<tr>
<td></td>
<td>never contacted IoT FND, never appears in this field. Otherwise, IoT FND</td>
</tr>
<tr>
<td></td>
<td>displays the date and time of the last contact, for example, 4/10 19:06.</td>
</tr>
<tr>
<td>Tunnel Source Interface 1</td>
<td>Router interface used by the tunnel.</td>
</tr>
<tr>
<td>Tunnel Source Interface 2</td>
<td>Router interface used by the tunnel.</td>
</tr>
<tr>
<td>OSPF Area 1</td>
<td>Open shortest path first (OSPF) areas 1 and 2.</td>
</tr>
<tr>
<td>OSPF Area 2</td>
<td></td>
</tr>
<tr>
<td>OSPFv3 Area 1</td>
<td>OSPFv3 area 1.</td>
</tr>
<tr>
<td>OSPFv3 Area 2</td>
<td>OSPFv3 area 2.</td>
</tr>
<tr>
<td>IPsec Dest Addr 1</td>
<td>IPv4 destination address of the tunnel.</td>
</tr>
<tr>
<td>IPsec Dest Addr 2</td>
<td></td>
</tr>
<tr>
<td>GRE Tunnel Dest Addr 1</td>
<td>IPv6 destination address of the tunnel.</td>
</tr>
<tr>
<td>GRE Tunnel Dest Addr 2</td>
<td></td>
</tr>
<tr>
<td>Certificate Issuer</td>
<td>Name of the CA that issued the certificate.</td>
</tr>
<tr>
<td>Common Name</td>
<td></td>
</tr>
</tbody>
</table>

Renaming a Tunnel Group

You can rename a tunnel group at any time. Cisco recommends using short, meaningful names. Names cannot be more than 250 characters long.

To rename a tunnel group:

1. Choose CONFIG > Tunnel Provisioning.
2. In the TUNNEL GROUPS pane, mouse over the tunnel group to rename and click the Edit pencil icon ( ).
3. Enter the new Group Name and then click OK.
Note: When you enter an invalid character entry (such as, @, #, !, or +) in the entry field, the field is highlighted in red and disables the OK button.

Moving FARs to Another Group

You can move FARs to another group in two ways:

- Moving FARs to Another Group Manually
- Moving FARs to Another Group in Bulk

Moving FARs to Another Group Manually

To move FARs to another group manually:

1. Choose CONFIG > Tunnel Provisioning.

2. Click the Group Members tab.

3. In the TUNNEL GROUPS pane, select the tunnel group with the routers to move.

4. Choose the device type from the Select a device type drop-down menu.

5. Check the check boxes of the FARs to move.

   To select all FARs in a group, click the check box at the top of the column. When you select devices, a yellow bar displays that maintains a count of selected devices and has the Clear Selection and Select All commands. The maximum number of devices you can select is 1000.

6. Click the Change Tunnel Group button.

7. From the drop-down menu, choose the tunnel group to which you want to move the FARs.
Managing Tunnel Provisioning

8. Click **Change Tunnel Group**.

9. Click **OK** to close the dialog box.

Moving FARs to Another Group in Bulk

You can move FARs in bulk to another group by importing a CSV or XML file containing the names of the FARs to move. Ensure that the file contains entries in the format shown the following example:

```
eid
CGR1120/k9+JSM1
CGR1120/k9+JSM2
CGR1120/k9+JSM3
CGR1120/k9+JSM4
C819HGW-S-A-K9+FTX174685V0
```

The first line is the header, which tells IoT FND to expect FAR EIDs in the remaining lines (one FAR EID per line).

To move FARs to another group in bulk:

1. Create a CSV or XML file with the EIDs of the devices to move to a different group.

2. Choose **CONFIG > Tunnel Provisioning**.

3. Click **Assign Devices to Tunnel Group** to open an entry panel.

4. Click **Browse** and locate the file that contains the FARs that you want to move.

5. From the **Group** drop-down menu, choose the destination tunnel group.

6. Click **Assign To Group**.

7. Click **Close**.
Managing Tunnel Provisioning

Configuring Tunnel Provisioning Templates

IoT FND has three default tunnel provisioning templates:

- **Field Area Router Tunnel Addition**—IoT FND uses this template to generate the CLI configuration commands for creating one end of an IPsec tunnel on the FAR.
- **Head-End Router Tunnel Addition**—IoT FND uses this template to generate the CLI configuration commands for creating the other end of the IPsec tunnel on the HER.
- **Head-End Router Tunnel Deletion**—IoT FND uses this template to generate the CLI configuration commands for deleting any existing tunnel to the FAR at the other end of the tunnel.

Tunnel Provisioning Template Syntax

The IoT FND tunnel provisioning templates are expressed with the FreeMarker syntax. FreeMarker is an open-source Java-based engine for processing templates and is built into IoT FND. As shown in **Figure 2**, FreeMarker takes as input the tunnel provisioning template and data supplied by IoT FND, and generates CLI commands that IoT FND runs on the FARs and HERs in the “configure terminal” context.

**Figure 2  CLI Command Generation from Templates in IoT FND**

In IoT FND, the tunnel provisioning templates consist of router CLI commands and FreeMarker variables and directives. The use of FreeMarker syntax allows IoT FND to define one template to provision multiple routers.

This section describes the basic FreeMarker syntax in the tunnel provisioning templates. For information about FreeMarker visit [http://freemarker.sourceforge.net/](http://freemarker.sourceforge.net/).

- **Template Syntax**
- **Data Model**
### Template Syntax

**Table 2** describes the syntax in the default tunnel provisioning templates.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text</strong></td>
<td>Unmarked text is carried through as CG-OS CLI configuration commands for FARs and Cisco IOS CLI commands for HERs.</td>
</tr>
</tbody>
</table>
| **Interpolations** | FreeMarker replaces this construct with the value of a string variable that IoT FND supplies. In this example, IoT FND provides the EID of the FAR:  
  `description IPsec tunnel to ${far.eid}`                          |
| **Default Values** | FreeMarker replaces this construct with the value of a string variable. If the variable is not set, FreeMarker replaces this construct with Default.                                                                 |
| **Conditionals** | FreeMarker uses this construct to determine the text to use in the output. For example:  
  `<#if far.ipsecTunnelDestAddr1??>`  
  `<#assign destinationAddress=far.ipsecTunnelDestAddr1>`  
  `<#else>`  
  `<#assign destinationAddress=her.interfaces("GigabitEthernet0/0/0")[0].v4.addresses[0].address>`  
  `</#if>`                                                                 |
| **Iteration over lists** | FreeMarker uses this construct to iterate over a list.                                                                 |
Managing Tunnel Provisioning

Data Model

This section describes the data model in the tunnel provisioning templates. The far and her prefixes provide access to the properties of the FARs and HERs, respectively. These properties are stored in the IoT FND database. Table 3 describes referencing the information provided by the data model in tunnel provisioning templates.

### Table 3  Data Model

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>far.eid</td>
<td>Returns the EID of the FAR. For example: ${far.eid}</td>
</tr>
<tr>
<td>far.hostname</td>
<td>Returns the hostname of the FAR.</td>
</tr>
<tr>
<td>far.tunnelSrcInterface1</td>
<td>Returns the name of the FAR interface on which to establish the tunnel.</td>
</tr>
</tbody>
</table>
## Table 3  Data Model (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>far.ipsecTunnelDestAddr1</td>
<td>Returns the name of the tunnel destination IP address on the HER.</td>
</tr>
</tbody>
</table>
| far.ipv4Address(clientId, linkAddress, userClass) | Returns an IPv4 address. The IPv4 address method takes these parameters as input:  
  - clientId — DCHP Client Identifier for the DHCP request  
  - linkAddress — Link address for the DHCP request  
  - userClass — Value for the DHCP User Class option (defaults to "CG-NMS")  

To establish a loopback interface and assign it an address:

```bash
interface Loopback0  
ip address ${far.ipv4Address(dhcpClientId(far.enDuid, 0),  
  far.dhcpV4LoopbackLink).address}/32  
ipv6 address ${far.ipv6Address(far.enDuid, 0,  
  far.dhcpV6LoopbackLink).address}/128  
exit
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| far.ipv4Subnet()                 | Returns a DHCP IPv4 subnet lease. This call takes a clientId and linkAddress as arguments.  

Construct the clientId from the FAR EID and interface ID number using the dhcpClientId() method provided in the template API. This method takes as input a DHCPv6 Identity Association Identifier (IAID) and a DHCP Unique Identifier (DUID) and generates the DHCPv4 client identifier, as specified in RFC 4361. This method provides consistency for how network elements are identified by the DHCP server.  

For example:

```bash
<#assign lease=far.ipv4Subnet(dhcpClientId(far.enDuid,iaId),  
  far.dhcpV4TunnelLink)>
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>far.[any device property]</td>
<td>Returns the value of the specified property.</td>
</tr>
</tbody>
</table>

For example, far.tunnelSrcInterface1 returns the value of the FAR tunnelSrcInterface1 property.  

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| far.interfaces(interfaceNamePrefix) | Returns a list of device interfaces that match the requested prefix (not case sensitive).  

Use square brackets to index list members, for example, [0], [1], [2], and so on. Use the `<#list>` construct to iterate list members.  

For example:

```bash
<#assign wanInterface = far.interfaces(interfaceNamePrefix)>  
<#if (wanInterface[0].v4.addresses[0].address)??>
  ...
```
Managing Tunnel Provisioning

Addresses

Table 4 describes referencing addresses in the tunnel provisioning templates.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address.address</td>
<td>Returns the address of the interface.</td>
</tr>
<tr>
<td>address.prefixLength</td>
<td>Returns the prefix length of the address.</td>
</tr>
<tr>
<td>address.prefix</td>
<td>Returns the address prefix.</td>
</tr>
<tr>
<td>address.subnetMask</td>
<td>Returns the subnet mask for the address.</td>
</tr>
<tr>
<td>address.wildcardMask</td>
<td>Returns the wildcard mask for the subnet.</td>
</tr>
</tbody>
</table>

**Table 4 Address References**

Configuring the Field Area Router Tunnel Addition Template

To edit the FAR Tunnel Addition template to provide one end of an IPsec tunnel on FARs in the group:

1. Choose **CONFIG > Tunnel Provisioning**.
2. In the TUNNEL GROUPS pane, select the tunnel group with the template to edit.
3. Click the **Router Tunnel Addition** tab.
4. Modify the default template.

**Tip**: Use a text editor to modify templates and copy the text into the template field in IoT FND.
5. Click the **Disk** icon to save changes.
6. Click **OK** to confirm the changes.

See also, [Tunnel Provisioning Template Syntax](#).

### Configuring the Head-End Router Tunnel Addition Template

**Note:** To ensure that both endpoints are in a matching subnet, this template must use the same IAID as the FAR template.

To edit the HER Tunnel Addition template to create the other end of the IPsec tunnel on HERs in the group:

1. Choose **CONFIG > Tunnel Provisioning**.
2. In the TUNNEL GROUPS pane, select a tunnel group.
3. Click the **HER Tunnel Addition** tab.
4. Modify the default HER addition template.
5. Click the **Disk** icon to save changes.
6. Click **OK** to confirm the changes.

### Configuring the HER Tunnel Deletion Template

To edit the HER tunnel deletion template to delete existing tunnels to FARs at the other end of the tunnel:

1. Choose **CONFIG > Tunnel Provisioning**.
2. In the TUNNEL GROUPS pane, select the tunnel group whose template to edit.
3. Click the **HER Tunnel Deletion** tab.
4. Modify the default HER deletion template.
5. Click the **Disk** icon to save changes.
6. Click **OK** to confirm the changes.

### Monitoring Tunnel Status

To view tunnel status, choose **OPERATIONS > Tunnel Status**. The Tunnel Status page lists devices and their provisioned tunnels and displays relevant information about tunnels and their status. Tunnels are provisioned between HERs and FARs.

When you select **Show Filter** at the top of the page (when selected, replaced by **Hide Filter**), a number of search fields appear. You can filter by all the Field Names listed in Table 5. The value entered in one search field will determine the available selections in the other fields. Select **Hide Filter** to remove the search fields.

Table 5 describes the tunnel status fields. To change the sort order of tunnels in the list by name, click the HER Name column heading. A small arrow next to the heading indicates the sort order.

**Note:** It takes time for the status of the newly created tunnel to be reflected in IoT FND.
Managing Tunnel Provisioning

Reprovisioning CGRs

In IoT FND, CGR reprovisioning is a process for modifying the configuration files on CGRs.

- CGR Reprovisioning Basics
- Tunnel Reprovisioning
- Factory Reprovisioning

Note: C800s do not support reprovisioning.

Table 5  Tunnel Status Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HER Name</td>
<td>The EID of the HER at one end of the tunnel. To view the HER details, click its EID.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Because one HER can serve up to 500 FARs, there may be multiple tunnels in the list with the same HER EID.</td>
</tr>
<tr>
<td></td>
<td>The Network Interfaces area of the Device Info page displays a list of tunnels configured on the HER. The Config Properties and Running Config tabs also contain information about tunnels configured on this HER.</td>
</tr>
<tr>
<td>HER Interface</td>
<td>The name of the HER tunnel interface. These names are automatically generated when tunnels are created (Tunnel1, Tunnel2, Tunnel3, and so on) or Virtual-Interface1, Virtual-Interface 2 and so on.</td>
</tr>
<tr>
<td>Admin Status</td>
<td>The administrative status of the tunnel (up or down). This indicates if the administrator enabled or disabled the tunnel.</td>
</tr>
<tr>
<td>Oper. Status</td>
<td>The operational status of the tunnel (up or down). If the tunnel is down, traffic does not flow through the tunnel, which indicates a problem to troubleshoot. Ping the HER and FAR to determine if they are online, or log on to the routers over SSH to determine the cause of the problem.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The protocol used by the tunnel (IPSEC, PIM, or GRE).</td>
</tr>
<tr>
<td>HER Tunnel IP Address</td>
<td>The IP address of the tunnel at the HER side. Depending on the protocol used, the IP address appears in dotted decimal (IPv4) or hexadecimal (IPv6) slash notation.</td>
</tr>
<tr>
<td>HER IP Address</td>
<td>The destination IP address of the tunnel on the HER side.</td>
</tr>
<tr>
<td>FAR IP Address</td>
<td>The destination IP address of the tunnel on the FAR side.</td>
</tr>
<tr>
<td>FAR Interface</td>
<td>The name of the interface on the FAR used by the tunnel.</td>
</tr>
<tr>
<td>FAR Tunnel IP Address</td>
<td>The IP address of the tunnel on the FAR side.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The IP addresses on both sides of the tunnel are on the same subnet.</td>
</tr>
<tr>
<td>FAR Name</td>
<td>The EID of the FAR. To view the FAR details, click its EID.</td>
</tr>
<tr>
<td></td>
<td>The Network Interfaces area of the Device Info page displays a list of tunnels configured on the FAR. The Config Properties and Running Config tabs also contain information about tunnels configured on this FAR.</td>
</tr>
</tbody>
</table>
Managing Tunnel Provisioning

CGR Reprovisioning Basics

- CGR Reprovisioning Actions
- CGR Reprovisioning Sequence

CGR Reprovisioning Actions

In IoT FND, you can perform the following two CGR reprovisioning actions at the Reprovisioning Actions pane of the Tunnel Provisioning page (CONFIG > Tunnel Provisioning > Reprovisioning Actions). You can also activate mesh firmware.

**Tip:** You can also type in the interface instead of selecting the preloaded interface values.

<table>
<thead>
<tr>
<th>Reprovisioning Actions</th>
<th>Description</th>
</tr>
</thead>
</table>
| Factory Reprovisioning                  | Drop-down menu allows you to change the express-setup-config file loaded on the CGR during factory configuration.  
                                           | This file contains a minimal set of information and is loaded on the CGR at the factory. This file provides the CGR with information to contact IoT FND (call home) through the TPS Proxy after the CGR is deployed and powered on. |
| Tunnel Reprovisioning                   | Drop-down menu allows you to change the golden-config file on a CGR. This file has the tunnel configuration defined on the CGR. |
| Mesh Firmware Activation                | Drop-down menu allows you to select the Interface (such as cellular, Ethernet, etc.) and Interface Type (IPv6 or IPv4). |

Table 6 describes the fields on the Reprovisioning Actions pane.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Action</td>
<td>The current reprovisioning action being performed and the associated interface.</td>
</tr>
<tr>
<td>Reprovisioning Status</td>
<td>The status of the reprovisioning action.</td>
</tr>
<tr>
<td>Completed devices / All Scheduled Devices</td>
<td>The number of CGRs that were processed relative to the number of all CGRs scheduled to be processed.</td>
</tr>
<tr>
<td>Error devices/ All Scheduled Devices</td>
<td>The number of CGRs that reported an error relative to the number of all CGRs scheduled to be processed.</td>
</tr>
<tr>
<td>Name</td>
<td>The EID of the CGR.</td>
</tr>
<tr>
<td>Reprovisioning Status</td>
<td>The status of the reprovisioning action for this CGR.</td>
</tr>
<tr>
<td>Last Updated</td>
<td>The last time the status of the reprovisioning action for this CGR was updated.</td>
</tr>
</tbody>
</table>
Managing Tunnel Provisioning

CGR Reprovisioning Sequence

When you start tunnel or factory reprovisioning on a tunnel provisioning group, the reprovisioning algorithm sequentially goes through 12 CGRs at a time and reprovisions them.

After IoT FND reprovisions a router successfully or if an error is reported, IoT FND starts the reprovisioning process for the next router in the group. IoT FND repeats the process until all CGRs are reprovisioned.

There is a timeout of 4 hours when reprovisioning each CGR in the group. If the CGR does not report successful reprovisioning or an error within the timeout period, then IoT FND changes the Reprovisioning Status of the CGR to Error and displays a timeout error and any further information displays in the Error Details field.

Tunnel Reprovisioning

If you make changes to the Field Area Router Tunnel Addition template and want all CGRs already connected to IoT FND reprovisioned with new tunnels based on the modified template, use the tunnel reprovisioning feature of IoT FND.

Tunnel reprovisioning places the CGR in a state where no tunnels are configured, and then initiates a new tunnel provisioning request. To reprovision tunnels, IoT FND sequentially goes through the FARs (12 at a time) in a tunnel provisioning group. For every CGR, IoT FND rolls back the configuration of the CGR to that defined in the ps-start-config template file.

After a rollback to ps-start-config, the CGR contacts IoT FND to request tunnel provisioning. IoT FND processes the Field Area Router Tunnel Addition template and sends the resultant configuration commands for creating new tunnels to the CGR. As shown in Figure 3, the tunnel provisioning process includes updating the golden-config file to include the new configuration information.

### Table 6  Reprovisioning Actions Pane Fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Version</td>
<td>The version of the Field Area Router Factory Reprovision template being applied.</td>
</tr>
<tr>
<td>Error Message</td>
<td>The error message reported by the CGR, if any.</td>
</tr>
<tr>
<td>Error Details</td>
<td>The error details.</td>
</tr>
</tbody>
</table>

97
Managing Tunnel Provisioning

Figure 3  Tunnel Reprovisioning Process (CG-OS Routers)

Note: For CG-OS CGRs, FND initiates a config rollback and if the rollback fails, the router will go through a reload. Also, when IoT FND rolls back a CGR, IoT FND removes the corresponding tunnel information from the HERs to which the CGR was connected.

For Cisco IOS routers, the checkpoint files are before-tunnel-config, before-registration-config, and Express-setup-config. You perform a configuration replace for Cisco IOS based CGRs.

Note: The Field Area Router Factory Reprovision template is not used when performing tunnel reprovisioning.

To configure and trigger tunnel reprovisioning:

1. Choose CONFIG > Tunnel Provisioning.
2. In the TUNNEL GROUPS pane, select the tunnel group whose template to provision.
3. Click the Reprovisioning Actions tab.
4. From the Action drop-down menu, choose Tunnel Reprovisioning.
5. Click Start.

IoT FND changes the Reprovisioning Status field to Initialized, and then to Running.

Note: If you click Stop while tunnel reprovisioning is running, IoT FND stops the reprovisioning process only for the FARs in the queue that were not selected. However, for those CGRs in the queue that were selected for reprovisioning, the process completes (success or error) and cannot be stopped.

The reprovisioning process completes after IoT FND finishes attempting to reprovision each CGR in the tunnel provisioning group. If a CGR cannot be reprovisioned, IoT FND displays the error message reported by the CGR.

Factory Reprovisioning

Use the Factory Reprovisioning feature in IoT FND to change the factory configuration of CGRs (express-setup-config).
Managing Tunnel Provisioning

Factory Reprovisioning involves these steps:

1. Sending the roll back command to the CGR.
2. Reloading the CGR.
3. Processing the Field Area Router Factory Reprovision template, and pushing the resultant commands to the CGR.
4. Saving the configuration in the express-setup-config file.

After these steps complete successfully, IoT FND processes the Field Area Router Tunnel Addition, Head-End Router Tunnel Addition, and Head-End Router Tunnel Deletion templates and pushes the resultant commands to the CGR (see Tunnel Provisioning Configuration Process).

To configure and trigger factory reprovisioning:

1. Choose CONFIG > Tunnel Provisioning.
2. In the TUNNEL GROUPS pane, select the tunnel group whose template you want to edit.
3. Click the Router Factory Reprovision tab and enter the template that contains the configuration commands to apply.
   
   **Note:** The Router Factory Reprovision template is processed twice during factory reprovisioning; once when pushing the configuration and again before saving the configuration in express-setup-config. Because of this, when making your own template, use the specific if/else condition model defined in the default template.

4. Click Disk icon to Save.
5. If needed, make the necessary modifications to the Field Area Router Tunnel Addition, Head-End Router Tunnel Addition, and Head-End Router Tunnel Deletion templates.
6. Click Reprovisioning Actions tab.
7. Select Factory Reprovisioning.

8. From the Interface drop-down menu, choose the CGR interface for IoT FND to use to contact the FARs for reprovisioning.
9. From the Interface Type drop-down menu, choose IPv4 or IPv6.
10. Click the Start button.

   IoT FND changes the Reprovisioning Status field to Initialized, and then to Running.

   **Note:** If you click Stop while factory reprovisioning is running, IoT FND stops the reprovisioning process only for the FARs in the queue that were not selected. However, for those CGRs in the queue that were selected for reprovisioning, the process completes and cannot be stopped.

The reprovisioning process completes after IoT FND has finished attempting to reprovision each CGR in the tunnel provisioning group. If a CGR cannot be reprovisioned, IoT FND displays the error message reported by the CGR.
Sample Field Area Router Factory Reprovision Template

This sample template changes the WiFi SSID and passphrase in the factory configuration.

<#--IMPORTANT: This template is processed twice during factory reprovisioning. The if/else condition described below is needed to determine which part of the template is applied. In this example, if no schedule name wimaxMigrationRebootTimer is found in runningConfig, then the if part of the if/else section is applied. During the second pass, this template runs the commands in the else section and the no scheduler command is applied. If modifying this template, do not remove the if/else condition or else the template fails. -->

<#if !far.runningConfig.text?contains("scheduler schedule name wimaxMigrationRebootTimer")>

<#--Comment: This is a sample of generating wifi ssid and passphrase randomly-->

wifi ssid ${far.randomSSID("PREFIX_")}
authentication key-management wpa2
wpa2-psk ascii ${far.randomPassword(10)}
exit

feature scheduler
scheduler job name wimaxMigration
reload
exit

scheduler schedule name wimaxMigrationRebootTimer
time start +02:00
job name wimaxMigration
exit

<#else>

no scheduler job name wimaxMigration
no scheduler schedule name wimaxMigrationRebootTimer

</#if>
Managing High Availability Installations

This section describes how to set up IoT FND for high availability, and includes the following sections:

- Overview of IoT FND High Availability
- HA Guidelines and Limitations
- Configuring IoT FND Installations for HA

Overview of IoT FND High Availability

This section provides an overview of IoT FND high availability installations, including the following sections:

- Load Balancer
- Server Heartbeats
- Database High Availability
- Tunnel Redundancy

IoT FND is a critical application for monitoring and managing a connected grid. IoT FND High Availability (IoT FND HA) solutions address the overall availability of IoT FND during software, network, or hardware failures.

IoT FND provides two main levels of HA, as shown in Figure 1:

- IoT FND Server HA—This is achieved by connecting multiple IoT FND servers to a Cisco ACE 4710 load balancer. Traffic originating at MEs, FARs, and ASRs goes to the load balancer, which uses a round-robin protocol 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.
In addition to IoT FND Server and Database HA, IoT FND improves reliability by adding tunnel redundancy. This is achieved by defining multiple tunnels between one FAR and multiple ASRs. If one tunnel fails, the FAR routes traffic through another tunnel.

IoT FND HA addresses these failure scenarios:

<table>
<thead>
<tr>
<th>Failure Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT FND server failure</td>
<td>If a server within a IoT FND server cluster fails, the load balancer routes traffic to the other servers in the cluster.</td>
</tr>
<tr>
<td>IoT FND database failures</td>
<td>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.</td>
</tr>
<tr>
<td>Tunnel failure</td>
<td>If a tunnel fails, traffic flows through another tunnel.</td>
</tr>
</tbody>
</table>

Load Balancer

The Load Balancer (LB) 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 LB directs traffic to other cluster members.
Managing High Availability Installations


Server Heartbeats

The LB maintains heartbeats with each IoT FND server in the cluster. In the health monitoring mechanism adopted by the IoT FND solution (there are alternate solutions), the heartbeats are regular GET messages to IoT FND on port 80. IoT FND expects an HTTP 200 OK response from an active IoT FND server.

You can configure these heartbeat parameters on the LB:

- **Periodicity of probes**—This is the number of seconds between heartbeats. The default value on the Cisco ACE is 15 seconds.
- **Number of retries**—This is the number of times the LB tries to send a heartbeat to a non-responding IoT FND server before declaring it down. The default number of retries is 3.
- **Regular checks after failure detection**—The LB checks whether the server is back online at this time interval. The default failure detection check value is 60 seconds.

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).

The IoT FND Database HA configuration process involves:

- Configuring the primary and secondary databases the same on separate physical servers.
  
  **Note:** The secondary database server is also referred to as the standby database.

  **Note:** There is a possibility of losing some data during a database failover.

- 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.

  **Note:** 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.
Managing High Availability Installations

Tunnel Redundancy

To add another layer of redundancy to your IoT FND deployment, configure multiple tunnels to connect every FAR in a FAR tunnel provisioning group to multiple ASRs. For example, you could configure IoT FND to provision two tunnels for every FAR. One tunnel is active over the Cellular interface, while the redundant tunnel is configured to communicate with a second ASR over the WiMAX interfaces.

To configure tunnel redundancy, you need to:

1. Add ASRs to a tunnel provisioning group.
2. Modify the tunnel provisioning templates to include commands to create additional tunnels.
3. Define policies that determine the mapping between interfaces on the FAR and ASR interfaces:
   - Configuring Tunnel Provisioning Policies
   - Modifying the Tunnel Provisioning Templates for Tunnel Redundancy

HA Guidelines and Limitations

Note the following about IoT FND HA configurations:

- IoT FND HA does not include HA support for other network components like FARs, ASRs, and the load balancer.
- 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.

HA Installation for FND

Be sure to enter the following information in the /opt/cgms/bin/cgms.conf file:

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.

Configuring IoT FND Installations for HA

This section describes the various configuration settings for IoT FND HA installations, including the following sections:

- Setting Up IoT FND Database for HA
- Disabling IoT FND Database HA
- Load-Balancing Policies
Setting Up IoT FND Database for HA

To set up the IoT FND Database HA:

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.

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`.

3. Set up IoT FND for database HA (see Setting Up IoT FND for Database HA).

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.

```
$ ./setupStandbyDb.sh
$ Are you sure you want to setup a standby database ? (y/n)? y

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:
```

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
...
Managing High Availability Installations

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.

```
$ ./setupHaForPrimary.sh
[oracle@pdb ha]$ ./setupHaForPrimary.sh
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: 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

Enter standby database server IP address: 192.168.1.10
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 the IoT FND Oracle Database

To set up the Observer:

1. On a separate server, run the observer script.

   ```
   $ ./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
   ```

2. Run the getHaStatus.sh script to verify that the database is set up for 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
   ```
To set up IoT FND for database HA:

1. Stop IoT FND.
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.

```bash
# cd /opt/cgms/bin
# ./setupCgms.sh
09-13-2012 17:10:00 PDT: INFO: ========= CG-NMS Setup Started - 2012-09-13-17-10-00 =========
09-13-2012 17:10:00 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

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

Enter database server IP address [128.107.154.246]: 128.107.154.246

Enter database server port [1522]:
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522
```
Enter database SID [cgms]:
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms

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

Enter database server IP address []: 128.107.154.20

Enter database server port []: 1522
09-13-2012 17:11:07 PDT: INFO: Database server port: 1522

Enter database SID []: cgms_s
09-13-2012 17:11:12 PDT: INFO: Database SID: cgms_s


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

Enter database password:
Re-enter database password:

... Do you want to change the keystore password (y/n)? n


Do you want to change the web application 'root' user password (y/n)? n

09-13-2012 17:16:34 PDT: INFO: User response: n

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

09-13-2012 17:16:45 PDT: INFO: User response: n

09-13-2012 17:16:45 PDT: INFO: =========== CG-NMS Setup Completed Successfully ==========

Disabling IoT FND Database HA

To disable IoT FND Database HA:

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

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: Cleaning up instance - cgms_s
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: 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: 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 "cgmstns" has 1 instance(s).
  Instance "cgmstns", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
09-20-2012 14:25:30 PDT: INFO: ========== Completed Successfully ==========
Load-Balancing Policies

Table 1 describes the load-balancing policy for each type of traffic the LB supports:

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Load Balancing Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS traffic to and from browsers and IoT FND API clients (IPv4; ports 80 and 443)</td>
<td>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.</td>
</tr>
<tr>
<td>For FAR IPv4 traffic going to ports 9121 and 9120:</td>
<td>The LB uses Layer 3 load balancing for all FAR traffic. This is the traffic from the FAR to IoT FND.</td>
</tr>
<tr>
<td>▪ Tunnel Provisioning on port 9120 over HTTPS</td>
<td></td>
</tr>
<tr>
<td>▪ Regular registration and periodic on 9121 over HTTPS</td>
<td></td>
</tr>
<tr>
<td>For IPv6 CSMP traffic to and from mesh endpoints (MEs):</td>
<td>The LB uses Layer 3 load balancing for all ME traffic to port 61624, and outage messages to port 61625.</td>
</tr>
<tr>
<td>▪ UDP traffic over port 61624</td>
<td></td>
</tr>
<tr>
<td>▪ Registration</td>
<td></td>
</tr>
<tr>
<td>▪ Periodic transmission of metrics</td>
<td></td>
</tr>
<tr>
<td>▪ Firmware push</td>
<td></td>
</tr>
<tr>
<td>▪ Configuration push</td>
<td></td>
</tr>
<tr>
<td>▪ UDP traffic over port 61625</td>
<td></td>
</tr>
<tr>
<td>For outage notifications sent by MEs.</td>
<td></td>
</tr>
</tbody>
</table>

Running LB Configuration Example

The following is an example of the running configuration of a properly configured IoT FND LB:

```
# show running-config
Generating configuration....

ssh maxsessions 10

boot system image: c4710ace-t1k9-mz.A5_1_1.bin

hostname cgmsalb2
interface gigabitEthernet 1/1
  switchport access vlan 10
  no shutdown
interface gigabitEthernet 1/2
description server-side
  switchport access vlan 11
  no shutdown
interface gigabitEthernet 1/3
description client-side
  switchport access vlan 8
  no shutdown
interface gigabitEthernet 1/4
  switchport access vlan 55
  no shutdown
```
access-list ALL line 8 extended permit ip any any
access-list everyone line 8 extended permit ip any any
access-list everyone line 16 extended permit icmp any any
access-list ipv6_acl line 8 extended permit ip anyv6 anyv6
access-list ipv6_acl2 line 8 extended permit icmpv6 anyv6 anyv6

ip domain-lookup
ip domain-name cisco.com
ip name-server 171.68.226.120
ip name-server 171.70.168.183

probe http probe_cgnms-http
   port 80
   interval 15
   passdetect interval 60
   expect status 200 200
   open 1

rserver host 12-12-1-31
   ip address 12.12.1.31
   conn-limit max 4000000 min 4000000
   probe probe_cgnms-http
   inservice

rserver host 12-12-1-32
   ip address 12.12.1.32
   conn-limit max 4000000 min 4000000
   probe probe_cgnms-http
   inservice

rserver host 2002-cafe-server-202
   description realserver 2002:cafe:server::202
   ip address 2002::202
   conn-limit max 4000000 min 4000000
   probe probe_cgnms-http
   inservice

rserver host 2002-cafe-server-211
   ip address 2002:cafe:server::211
   conn-limit max 4000000 min 4000000
   probe probe_cgnms-http
   inservice

serverfarm host cgnms_2
   description cgnms-serverfarm
   probe probe_cgnms-http
   rserver 2002-cafe-server-202 61624
      conn-limit max 4000000 min 4000000
      inservice
   rserver 2002-cafe-server-211 61624
      conn-limit max 4000000 min 4000000
      inservice

serverfarm host cgnms_2_ipv4
   probe probe_cgnms-http
   rserver 12-12-1-31
      conn-limit max 4000000 min 4000000
      inservice
   rserver 12-12-1-32
      conn-limit max 4000000 min 4000000
      inservice

sticky ip-netmask 255.255.255.255 address source CGNMS_SRC_STICKY
serverfarm cgnms_2_ipv4
Managing High Availability Installations

class-map type management match-any remote_access
  2 match protocol xml-https any
  3 match protocol icmp any
  4 match protocol telnet any
  5 match protocol ssh any
  6 match protocol http any
  7 match protocol https any
  8 match protocol snmp any
class-map type management match-all ssh_allow_access
  2 match protocol ssh any
class-map match-any virtual-server-cgnms
  2 match virtual-address 2002:server:cafe::210 udp eq 61624
class-map match-any vs_cgnms_ipv4
  3 match virtual-address 12.12.1.101 tcp eq https
  4 match virtual-address 12.12.1.101 tcp eq 9120
  5 match virtual-address 12.12.1.101 tcp eq 9121
  6 match virtual-address 12.12.1.101 tcp eq 8443
  7 match virtual-address 12.12.1.101 tcp any
policy-map type management first-match remote_mgmt_allow_policy
  class remote_access
    permit
policy-map type loadbalance first-match virtual_cgnms_l7
  class class-default
    serverfarm cgnms_2
policy-map type loadbalance first-match vs_cgnms_l7_v4
  class class-default
    sticky-serverfarm CGNMS_SRC_STICKY
policy-map multi-match cgnms_policy_ipv6
  class virtual-server-cgnms
    loadbalance vip inservice
    loadbalance policy virtual_cgnms_l7
    loadbalance vip icmp-reply active
policy-map multi-match int1000
  class vs_cgnms_ipv4
    loadbalance vip inservice
    loadbalance policy vs_cgnms_l7_v4
    loadbalance vip icmp-reply active

interface vlan 8
  bridge-group 1
  access-group input everyone
  access-group input ipv6_acl
  no shutdown
interface vlan 10
  bridge-group 2
  access-group input everyone
  access-group input ipv6_acl
  service-policy input int1000
  no shutdown
interface vlan 11
  bridge-group 2
  access-group input everyone
  access-group input ipv6_acl
  no shutdown
interface vlan 55
  bridge-group 1
  access-group input everyone
  access-group input ipv6_acl
  service-policy input cgnms_policy_ipv6
no shutdown

interface bvi 1
ipv6 enable
ip address 2002:server:cafe::206/64
no shutdown
interface bvi 2
ip address 12.12.1.100 255.255.255.0
no shutdown
domain cisco.com

ip route 2011::/16 2002:server:cafe::101
ip route 2001:server:cafe::/64 2002:cafe::101
ip route 11.1.0.0 255.255.0.0 12.12.1.33
ip route 15.1.0.0 255.255.0.0 12.12.1.33
ip route 13.211.0.0 255.255.0.0 12.12.1.33

context VC_Setup1
allocate-interface vlan 40
allocate-interface vlan 50
allocate-interface vlan 1000

username admin password 5 $1$CB34uAB9$BW8a3ijxvBGttuGtTcST/ role Admin domain
default-domain
username www password 5 $1$q/YDKdp4g9PkZl1SBMQW7yZ7E.sOZA/ role Admin domain
default-domain

ssh key rsa 1024 force

Configuring Tunnel Provisioning Policies

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 (Config > Tunnel Provisioning). Use this policy to configure FAR-to-HER interface mapping.

To map FAR-to-HER interfaces in IoT FND:

1. Choose CONFIG > Tunnel Provisioning.

2. In the TUNNEL GROUPS pane, select a group to configure with tunnel redundancy.

3. Create a CSV or XML file that lists the HERs to add to the group in the format EID, device type, as follows:

<table>
<thead>
<tr>
<th>eid, deviceType</th>
</tr>
</thead>
<tbody>
<tr>
<td>asr-0, asr1000</td>
</tr>
<tr>
<td>asr-1, asr1000</td>
</tr>
<tr>
<td>asr-2, asr1000</td>
</tr>
</tbody>
</table>

4. Click Assign Devices to Group to import the file and add HERs to the group.
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.

6. 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.

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.
7. 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.

8. To enable the policy, check the Enabled check box.

9. 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.

Field Area Router Tunnel Addition Template Example

In this example, bold text indicates the changes made to the default Field Area Router Tunnel Addition template to create multiple tunnels:

```xml
<#--
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
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)>

117
Managing High Availability Installations

<!-- 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>
</#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. -->
    <#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 -->
<#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.greTunnelObj.greTunnelObj.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.

```puppet
<#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}
    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)>
  ip 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>

<#list far.tunnels("ipSec") as tunnelObject>
  <@configureTunnel tunnelObject.ipSecTunnelSrcInterface tunnelObject.ipSecTunnelDestInterface tunnelObject.her tunnelObject.tunnelIndex 100/>
</#list>
```
</#macro>

<#-- Loop through the policies for GRE tunnels -->
<#list far.tunnels("gre") as greTunnelObj>
  <@configureGreTunnel greTunnelObj.greSrcInterface greTunnelObj.her greTunnelObj.tunnelIndex/>
</#list>