

Troubleshoot Guide for Cisco IoT Field Network Director

First Published: 2020-08-20

Last Modified: 2026-06-09

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

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

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

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

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

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

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

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Troubleshooting Guide for Cisco Field Network Director** 1

Introduction 1

CHAPTER 2 **Troubleshooting Common IoT FND Issues** 3

Access Docker Containers 3

Common Errors 4

Troubleshoot Influx event migration to Postgres 20

FND Debugging — How to Enable 22

FND Debugging — Enable from FND Boot 24

Java Debugging 25

Log Files 25

Troubleshoot L+G device metrics refresh 26

Troubleshoot Metric Interval Errors 27

Runtime Exception Error 27

File Upload Error 29

Missing Configuration Error 30

Retrieve Inventory Error 30

SOAP Error 30

Timeout Error 31

Network Connectivity Errors 32

Webservice Error 33

SSL Debugging 34

Troubleshoot High CPU Issues 35

Zero Touch Deployment — Tunnel Provisioning 38

ZTD Easy Mode for PNP 38

Zero Touch Deployment Steps — Log Entries for Plug and Play 39

ZTD Step by Step — Entries for IXM Registration 39
ZTD Step by Step — Log Entries for IXM Tunnel 40
ZTD Step by Step — Log Entries for Registration 40
Upgrading the Virtual Machine 40

CHAPTER 3 **Troubleshooting for Specific IoT FND Components** 43
Troubleshoot PNP 44
Debug IGMA websocket sessions for L+G border routers 45
Troubleshooting Steps to Upload ODM File 47
Retrieve Inventory Error 47
Upload the ODM File from FND UI 56
Troubleshoot TCL Scripts 59
Troubleshoot Certificate Enrollment 59
Certificate Enrollment — Test Manual 60
Certificate Enrollment — Example Output 60
Troubleshoot WSMA 61
Troubleshoot Tunnel Provisioning 62
Troubleshoot Netconf: FND—HER Communications 62
Troubleshoot Configuration Deployment 64
Troubleshoot HSM Connectivity 64
Issues Faced During HSM Client Upgrade 65

CHAPTER 4 **Troubleshoot Tunnel Provisioning Issues** 67
Troubleshoot Archive Directory Creation During ZTD 68
Troubleshoot Zero Touch Deployment 69
Troubleshoot Tunnel Provisioning DHCP Configuration Issue 71
Troubleshoot Incorrect CGNA Profile URL 72
Troubleshoot HER FlexVPN 72
Troubleshoot Profile Environment Variable 72
Debug FAR Devices Using Commands 73
Verify TPS and DNS Connectivity 73
Verify Tunnel Configuration Information 74

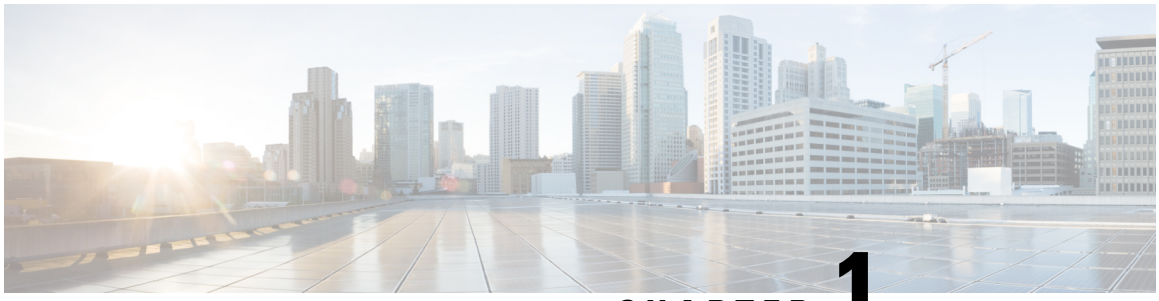
CHAPTER 5 **FAR Management Issues** 75

Certificate Exception 75
FAR Keeps Reloading and Does Not Switch to the Up State 75
Incorrect FAR State in IoT FND 76

CHAPTER 6 Mesh Endpoint Management Issues 77
Mesh Endpoints Not Registering with IoT FND 77
Mesh Endpoint Registration Reason Codes 77
Reasons for Mesh Endpoint WPAN Changes 78

CHAPTER 7 Operational Issues You May Encounter 81
Tunnel Provisioning DHCP Configuration Issues 81
Recovering an Expired Database Password 82
Unlocking the IoT FND Database Password 82
IoT FND Service Will Not Start 82
Exception in the server.log File on the IoT FND Server 83
Resetting the root Password 84
Second IoT FND Server Not Forming a Cluster 84
IoT FND Service Restarts Automatically 84
Fallback URL When SSO Fails 85
Router Registration with IoT FND Fails over Cellular Network after Successful Tunnel Provisioning 85
DB Migration fails due to Incorrect Incremental Size 86
Missing Endpoint Marker Line to CGR 88
Troubleshoot remote authentication failures when NTLMv2-only policy is enforced 88

CHAPTER 8 Licensing Issues 91
Device Import Failure 91
License File Upload Failure 91



CHAPTER 1

Troubleshooting Guide for Cisco Field Network Director

This guide provides steps to troubleshoot both common IoT FND issues and specific IoT FND components.

- Introduction, on page 1

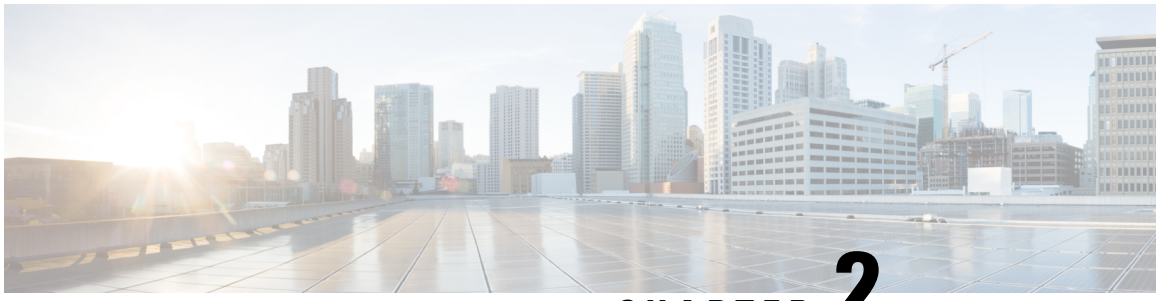
Introduction

This guide applies to the following deployments in IoT FND:

- IoT FND Installation using Docker, PostgreSQL and InfluxDB
- IoT FND Installation using Oracle

Note Not all troubleshooting options noted in this document apply to both deployment types noted above.

- Troubleshooting Common IoT FND Issues
- Troubleshooting for Specific IoT FND Components
- Operational Issues You May Encounter
- FAR Management Issues
- Mesh Endpoint Management Issues
- Licensing Issues, on page 91



CHAPTER 2

Troubleshooting Common IoT FND Issues

This chapter explains some common IoT FND issues and the workaround for them.

- Access Docker Containers, on page 3
- Common Errors, on page 4
- Troubleshoot Influx event migration to Postgres, on page 20
- FND Debugging — How to Enable, on page 22
- FND Debugging — Enable from FND Boot, on page 24
- Java Debugging, on page 25
- Log Files, on page 25
- Troubleshoot L+G device metrics refresh, on page 26
- Troubleshoot Metric Interval Errors, on page 27
- SSL Debugging, on page 34
- Troubleshoot High CPU Issues, on page 35
- Zero Touch Deployment — Tunnel Provisioning, on page 38
- ZTD Easy Mode for PNP, on page 38
- Zero Touch Deployment Steps — Log Entries for Plug and Play, on page 39
- ZTD Step by Step — Entries for IXM Registration, on page 39
- ZTD Step by Step — Log Entries for IXM Tunnel, on page 40
- ZTD Step by Step — Log Entries for Registration, on page 40
- Upgrading the Virtual Machine, on page 40

Access Docker Containers

Procedure

Step 1 To access FND or FD container shell (see Figure 5):

```
[root@iot-fnd ~]# docker exec -it fnd-container bash
[root@fnd-server /]#
```

Step 2 To copy files to and from containers (containers are not persistent):

```
[root@iot-fnd ~] # docker cp fnd-container:/opt/cgms/version.txt
[root@iot-fnd ~]# cat version.txt
JBoss Enterprise Application Platform - Version 6.2.0 GA
```

Figure 1: Access Docker Container



Common Errors

Listed below are some common errors that you may see during various stages of using IoT FND with suggested ways to resolve the problems.

If the OS version is RHEL 8.x or greater, then use **systemctl** command instead of the **service** command as given in the table.

Table 1: For CGMS

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> cgms`
7.x	`service cgms <status/start/restart/stop>`

Similarly, use the systemctl command for TPS Proxy and SSM as well.

Table 2: For TPSPROXY

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> tpsproxy`
7.x	`service tpsproxy <status/start/restart/stop>`

Table 3: For SSM

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> ssm`
7.x	`service ssm <status/start/restart/stop>`

Table 4: For FND RA

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> fnd-ra`
7.x	`service fnd-ra <status/start/restart/stop>`



Note To check the OS version, run the following command:

```
cat /etc/os-release
```

Table 5: Common Errors

Common Errors	Items to Check and/or Resolve Errors
Checkpoint Failed.	Check the archive.
CiscoIosFileUploadException: Full error: Error occurred while verifying file upload operation for net element CGR1120/K9+FOC21255MYX	Check provisioning URL (HTTP, HTTPS) Check WSMA with test script: user and port
org.apache.cxf.interceptor.Fault: Connection refused (Connection refused)	Check port used for HTTPS communication (varies by platform). For example: • FAR: ip http secure-port 8443 • IR1101: ip http secure-port 443

Common Errors	Items to Check and/or Resolve Errors
PnP Service Error 3341 Full error: Error while creating FND trustpoint on the device. errorCode: PnP Service Error 3341, errorMessage: SSL Server ID check failed after cert-install	Check SAN field in the FND certificate: • Certificate which FND offers for PNP: https://10.48.43.229:9120/pnp/HELLO • Trustpoint which FND offers for PNP: Click to view the truspoint. For additional information, click to view the document: Enter the keystore command to list SAN fields on the certificate in the keystore used for PNP. This verifies the accuracy of the SAN field(s). keytool -list -v -keystore cgms_keystore | grep SubjectAlt -A3 Enter keystore password: keystore SubjectAlternativeName [IPAddress: 10.48.43.229]

Common Errors	Items to Check and/or Resolve Errors
PnP Service Error 1702 Full error: Error while deploying odm/config file on the device. errorCode: PnP Service Error 1702, errorMessage: I/O error	If error is seen, enable debug in FND for bootstrapping, Ensure that FAR is able to reach TPS or FND using its hostname. For example, in the below debug logs for FND bootstrapping, FAR should be able to resolve and reach iot-tps.example.cisco.com on 9120 and viceversa. [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.4/16]: <fileTransfer> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.5/16]: <copy> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.6/16]: <source> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.7/16]: <location>https://iot-tps.example.cisco.com:9120/pnp/odm/IR829GW</location> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.8/16]: </source> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.9/16]: <destination> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.10/16]: <location>flash:/managed/odm/cg-nms.odm</location> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.11/16]: </destination>
java.lang.reflect. InnvocationTargetException. Full error description: PnP request for element ID [IR1101-K9+FCW223700AV] failed [java.lang.reflect.InvocationTargetException].	Check bootstrap configuration. If error is seen immediately after updating ODM: • Check provisioning settings in the user interface. • Check debug log for empty value for proxy-bootstrap-ip property field. • Must provide a valid IP address or hostname.
Could not generate DH keypair. Full error description: java.security.Invalid.AlgorithmParameterException: DH key size must be multiple of 64 and must be in the range of 512 to 2048 (inclusive). The specific key size 4096 is not supported.	Check: ip http secure-ciphersuite

Common Errors	Items to Check and/or Resolve Errors
Error: PKIX path building failed: sun.security.provider.certpath. SunCertPathBuilderException: unable to find valid certification path to requested target. Cause: Wrong certificate is offered through HTTPS-server on FAR.	Check the certificate for Web communication with IoT FND on the router (FAR): 1. Check the configuration of the secure-transport: • Router# sh run | i secure-trustpoint • ip http secure-trustpoint LDevID • ip http client secure-trustpoint LDevID 2. If the secure-transport configuration is correct, then restart https server on FAR: • router(config)# no ip http secure-server • router(config)# ip http secure-server

Common Errors	Items to Check and/or Resolve Errors
Error: PKIX path validation failed: java.security.cert.CertPathValidatorException: validity check failed. Cause: Wrong certificate is offered through HTTPS-server on FAR.	If this error is seen, then there is an issue with the certificate used for https communication between IoT FND and FAR. In certain situations, for example, if reload-during-bootstrap=true property is used in the cgms.properties file, then this error might be seen once, after which the tunnel formation is successful. This is because of the delay in obtaining the LDevID certificate after the router boots up. But the first tunnel formation request has already been sent before LDevID is obtained. So the first time failure of tunnel formation, this error message is seen. However, when the second tunnel formation request in sent, the LDevID has already been obtained by this time for the https communication and hence the tunnel formation is successful. Workaround: From IoT FND 4.6.x onwards, remove reload-during-bootstrap=true from the cgms.properties file, as this property was introduced as a workaround for CSCvk66991. **Note** CSCvk66991 is fixed now, hence this property is not mandatory from IoT FND 4.6.x onwards.

Common Errors	Items to Check and/or Resolve Errors
Error: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath. SunCertPathBuilderException: unable to find valid certification path to requested target Cause: Issuing CA certificate is missing in keystore.	Install Issuing CA cert.
Error in running file check command Full error: Error in running file check command: dir flash:/managed/odm/cg-nms.odm., Reason: javax.xml.ws.soap.SOAPFaultException: Serve D-H key verification failed	Add the following command to the file check: • ip http secure-client-auth • Check username and password or http conf.
Error during registration process: javax.xml.ws.WebServiceException: Could not send Message	Check WSMA. On the router (FAR), run debug: Router# debug ip http all
HTTP response ‘502: Bad Gateway’ Full error: org.apache.cxf.transport.http.HTTPException: HTTP response ‘502:Bad Gateway’ when communicating with https://10.48.43.249.443/wsma/config Error is typically seen with NGINX on IR1101. **Note** NGINX is a software-based web server. **Note** In most cases, the ‘502: Bad Gateway’ error is related to http max-connections set in the command below. tunnel(config)# ip http max-connections 20 **Note** Should the value that you enter in the command (noted above) return an error, you can increase the value until the error goes away.	On the IR1101, check NGINX log by entering one of the commands: IR1101# show platform software trace message nginx RP active -or- You can find the latest nginx file in the directory: IR1101# dir bootflash/tracelogs/nginx* To copy the latest nginx file, use one of the following: Cisco IOS file operations such as SCP or TFTP.

Common Errors	Items to Check and/or Resolve Errors
Failed to load function 'CA InitRolePIN'Issue with (outdated) HSM Java libraries Full error: Failed to load function 'CA_InitSlotRolePIN' Failed to load function 'CA_...Failed to load function 'CA_DescribeUtilizationCounterId' Failed to load function 'CA TestTrace'	Backup/copy new libs to cgms or cgms-tools libs folder: [root@FNDPRDAPP01 bin]# cp -r /opt/cgms-tools/jre/lib/ext/opt/cgms-tools/jre/lib/ext-bc/ root@FNDPRDAPP01 bin]# cp /usr/safenet/lunaclient/jsp/lib/*/opt/cgms-tools/jre/lib/ext/
Reverse DNS (1 of 2) Nothing in FND log when running CGNA on FAR tcpdump does not show incoming traffic to FND Debugging CGNA/HTTP on FAR shows: cgna_httpc_post: http_send_request rc= 0 tid=55 cgna_prf timer_start:cg-nms-register:timer started Thu Jul 18 14:10:55 2019 httpc_request:Do not have the credentials cgna_http_resp_data: Received for sid=5 tid=55 status= 7	Debugging CGNA/HTTP on FAR should be (rather than the display to the left): cgna_httpc_post: http_send_request rc= 0 tid=114 cgna_prf timer_start:cg-nms-periodic: timer started Thu Jul 18 16:37:38 2019 httpc_request: Dont have the credentials Jul 18 16:37:40.844 UTC: Thu, 18 Jul 2019 14:37:40 GMT 10.48.43.251 http:10.48.43.299/cgna/ios/metrics ok Protocol = HTTP/1.1 Jul 18 16:37:40.844 UTC: Date =Thu, 18 Jul 2019 14:40:27 GMT cgna_http_resp_data: Received for sid= 4 tid=114 status=8
Reverse DNS (2 of 2) Every time FAR tries (http client) to create a TLS connection with FND, Java does a reverse DNS lookup of the source IP of the device. This is by design in Java. Apparently, for preventing DDoS attacks.	Remove DNS server or set the following in the cgms.properties: enable-reverse-dns-lookup=false (Addressed in CSCvk59944)

Common Errors	Items to Check and/or Resolve Errors
FND will not start (1 of 2) Symptom: FND stops suddenly or is unable to start on an Oracle installation where the database is installed locally.	Check the hard disk space using the command 'df-h' on the linux shell. If the disk is showing as 'full', most likely the Oracle DB archive logs have filled up the disk space and needs cleaning. Another reason could be that the database password has expired. Run the command to confirm: /opt/cgms/server/cgms/log/cgms_db_connection_test.log To change the password, become the oracle user and use the script provided in the Oracle RPM: su - oracle $ORACLE_BASE/cgms/scripts/change_password.sh
FND will not start (2 of 2) Symptom: FND service is up but GUI will not load.	Issue is mostly likely due to Linux firewall getting enabled. Disable firewall using the Linux CLI command: systemctl firewalld stop

Common Errors	Items to Check and/or Resolve Errors
After FND is upgraded to FND 4.8, the HSM Client to FND Server communication does not work and displays the following error message: 'Could not get CsmpSignatureKeyStore instance. Please verify HSM connection. Exception: Object not found.' The error above is seen in FND Deployments with HSM that are running with or without High Availability (HA).	This is an HSM library issue. HSM client is not sending right slot ID to the FND server. Hence, the customer will have to follow up with HSM support. 'Could not get CsmpSignatureKeyStore instance. Please verify HSM connection. Exception: Object not found.' (CSCvz59702) Although, the HSM client resides on the same Linux server, where the FND Application Server is also installed. The HSM client is not provided by HSM and not by Cisco. Only HSM has the expertise and visibility to the HSM code and the HSM support team can help fix this issue. FND uses SSM or HSM to store encrypted information and keys. If there is an issue with SSM or HSM, then FND will not initialize. The IoT FND component remains in Down state even if the FND application server is in UP state. In this case, when the SSM is used, then you can contact Cisco Support. They have the expertise and visibility to the code to help you resolve this issue. However, if the HSM client to server connection has issues, then the Thales/HSM vendor has the visibility and expertise to help resolve the issue.

Common Errors	Items to Check and/or Resolve Errors
CSMP certificate not displayed in IoT FND GUI during fresh install.	

Common Errors	Items to Check and/or Resolve Errors
	For a fresh install of IoT FND and HSM integration, the CSMP certificate appears in the FND UI only when an endpoint/meter is added to FND, irrespective of whether th emeter/endpoint is registered to FND or not. You can also add a dummy entry for meter/endpoint. If there is no real endpoint or meter to add at the point of testing CSMP certificate display. Apart from the CSMP certificate displayed in the GUI, you can also use the following methods to verify if IoT FND can access and retrieve the CSMP certificate from HSM: • **Method 1** Run the following command: cat /opt/cgms/server/cgms/log/server.log | grep -i HSM If you get the below message, then IoT FND and HSM communication is successful, and FND can retrieve the public key. %IOTFND-6-UNSPECIFIED: %[ch=HSMKeyStore][sev=INFO] [tid=MSC service thread 1-3]: Retrieved public key: 3059301306072a8648ce3d020106082a864 8ce3d03010703 420004d914167514ec0a110 f3170eef742a000572cea6f0285a3074db 87e43da398 ab016e40ca4be5b888c26c4 fe91106cbf685a04b0f61d599826bdbcff 25cf065d24 • **Method 2** Run the following command. The cmu list command checks if FND can see

Common Errors	Items to Check and/or Resolve Errors
	two objects stored in HSM partition, namely private keys and CSMP certificate. [root@iot-fnd ~]# cd /usr/safenet/lunaclient/bin [root@iot-fnd bin]# ./cmu list Certificate Management Utility (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved. Please enter password for token in slot 0 : ******* handle=2000001 label=NMS_SOUTHBOUND_KEY handle=2000002 label=NMS_SOUTHBOUND_KEY--cert0 You have new mail in /var/spool/mail/root
Error: Caused by FATAL: terminating connection due to idle-in-transaction timeout	**Note** This is applicable only to FND-Postgres ova deployments. Edit the idle_in_transaction_session_timeout property in postgresql.conf file. By default it is set to 3h. If any operation requires the transaction to be opened for more than 3h then on getting the above error, set the value for the idle_in_transaction_session_timeout property to more than 3h and restart Postgresql service for the property to take effect. **Note** • The postgresql.conf file is located in the path: /var/lib/pgsql/12/data. • The postgres version is 12. (replace this with the current version that you are using).

Common Errors	Items to Check and/or Resolve Errors
With IoT FND and HSM integration, the CSMP certificate will not load in IoT FND UI after the upgrade.	The inability of the certificate to load is mostly likely due to the upgrade process overwriting the old HSM client libraries (example: version 5.x) with the new client libraries (example: version 7.x or 10.x or higher) that are bundled with FND 4.4 and later releases. **Note** For more information on the HSM client version that is bundled with IoT FND, refer to the corresponding FND release notes. To restore the old libraries, perform the following on the Linux shell: cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms/jre/lib/ext/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms/jre/lib/ext/ cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms/safenet/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms/safenet/ To restore the tools package: cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms-tools/jre/lib/ext cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms-tools/jre/lib/ext cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms-tools/safenet/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms-tools/safenet/

Common Errors	Items to Check and/or Resolve Errors
ODM file will not update on the router Symptom: During Plug and Play (PnP) or ZTD, the ODM file on the router does not get updated, which results in failure to register the device.	Issue is most likely due to the following entry in the cgms.properties file: update-files-oncgr=false Either remove the entry above or change it to 'true' as shown below: update-files-oncgr=true
Any CGR running Cisco IOS 15.6.x will not register with FND 4.3 or newer release.	Problem occurs because the WPAN high-availability (HA) feature was introduced in FND 4.3. This feature requires a minimum Cisco IOS release of 15.7(M)4.

Common Errors	Items to Check and/or Resolve Errors
SSM certificate will not load.	After upgrading to FND 4.4 or newer versions, the SSM cert is no longer seen in the CSMP certificates page. This occurs because the web certificate is getting changed after every upgrade. The web cert is used for establishing secure communication with the SSM. This change was done as part of the security compliance in FND 4.4. and all subsequent releases of FND, which generates a unique web (browser) certificate upon install or upgrade. To fix, export the self-signed web certificate from FND GUI: **1.** Go to Admin > Certificates > web certificate tab. Use the base64 format. **2.** Transfer the file to the opt/cgms-ssm directory. **3.** Stop SSM service: service ssm stop. **4.** Enter cd /opt/cgms-ssm/bin. **5.** Execute: /ssm setup.sh. **6.** Select option 8 : Import a trusted certificate to SSM-Web keystore. **7.** Enter current ssm_web_keystore password: *ssmweb*. **8.** Enter the alias for import: *fnd*. **9.** Enter Certificate filename: /opt/cgms-ssm/certForWeb.pem. **10.** Start the SSM service: service ssm start.
Could not get CsmpSignatureKeyStore instance. Please verify HSM connection.	This is an HSM client library issue. The HSM client is not sending the correct slot ID to the FND server. Please follow up with HSM support.

Common Errors	Items to Check and/or Resolve Errors
fndserver1.test.com: %IOTFND-3-UNSPECIFIED: %[ch=CgmsAuthenticator][sev=ERROR] [tid=http-/0.0.0.0:443-4] [part=150156.1/55]: Exception when adding remote user to the db. fndserver1.test.com: %IOTFND-3-UNSPECIFIED: %[ch=CgmsAuthenticator][sev=ERROR] [tid=http-/0.0.0.0:443-4] [part=150156.2/55]: com.cisco.cgms.exceptions.AAAException: failed to decrypt stored shared secret	The IoT FND server certificate contents for HA setup is: • The Subject — Must have the FQDN of the VIP. Example: FNDSERVERVIP.TEST.COM • The Subject Alternative Name (SAN) — Added must include the FQDN of the VIP. Example: FNDSERVERVIP.TEST.COM (same as the subject) • The Subject Alternative Name — Must NOT have the individual server names. Example: It must not contain FNDSERVER1.TEST.COM, FNDSERVER2.TEST.COM

Troubleshoot Influx event migration to Postgres

This task helps you migrate influx events to postgres without issues.

- When devices remain registered in Cisco IoT FND, migrating Influx events to Postgres completes without issues.
- If a device is removed from Cisco IoT FND, the **Events** page shows in the Name column: **eid doesn't exist in the system anymore**. In this state, the events migration throws errors and the affected records do not migrate to Postgres.

Before you begin

Here is how you troubleshoot the influx event migration to postgres:

Procedure

Re-upload the device CSV to restore the device in Cisco IoT FND, then rerun the events migration.

Example:

```
[root@iot-fnd log]# tail -f influx2postgres.log
2025-08-17 12:59:32 INFO  Influx2PostgresMigrator:102 - ##### Migration of records from Influx to
Postgres process: Started 2025-08-17T16:59:32.589Z #####

2025-08-17 12:59:42 INFO  Influx2PostgresMigrator:119 - parallelize : true
2025-08-17 12:59:42 INFO  Influx2PostgresMigrator:131 - Total number of measurements to migrate: 1
2025-08-17 12:59:42 INFO  Influx2PostgresMigrator:132 - Measurements : [events]
```

```
2025-08-17 12:59:42 INFO  Influx2PostgresMigrator:157 - Measurement: events has a retention period
of 31 days.
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:162 - Successfully updated retention period for
postgres table: events
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:178 - Parallel migration enabled at measurements
level. Using maximum of 20 read threads and 32 write threads.
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:246 - Starting migration for measurement: events
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:249 - Migration for measurement: events started at
 2025-08-17T16:59:43.227Z
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres table events before
 migration: 0
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:334 - Processing measurement: events with fields:
 eventmessage,eventtypename,eventsev,eventsource,geohash,lat,lng,eid,issueid,domainname,netobjtypekind
 for target table: events
2025-08-17 12:59:43 INFO  Influx2PostgresMigrator:397 - Processed last chunk for measurement: events,
 fields:
eventmessage,eventtypename,eventsev,eventsource,geohash,lat,lng,eid,issueid,domainname,netobjtypekind,
 offset: 0
2025-08-17 12:59:43 ERROR PostgresWriteTask:106 - [BATCH] FAILED in INSERT SQL: INSERT INTO events
(eventtime,eventmessage,eventtypename,eventsev,eventsource,geohash,lat,lng,eid,issue_id,domain_id,id,eventseverity,event_type_id,net_element_id)
 VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) Retrying rows individually
2025-08-17 12:59:43 ERROR PostgresWriteTask:176 - [ERROR] Exception during individual row insert in
 postgres DB: ERROR: null value in column "net_element_id" of relation "events" violates not-null
constraint
  Detail: Failing row contains (5, null, 1755449042531, 0, asr1000, 891, interfaceUp, GRE tunnel on
 interface Virtual-Access2 connected to 3.3.3.221 c..., 1000, 1000, null, CSR1000V+9PB415ITOSI, 0,
INFO, 1).
org.postgresql.util.PSQLException: ERROR: null value in column "net_element_id" of relation "events"
 violates not-null constraint
  Detail: Failing row contains (5, null, 1755449042531, 0, asr1000, 891, interfaceUp, GRE tunnel on
 interface Virtual-Access2 connected to 3.3.3.221 c..., 1000, 1000, null, CSR1000V+9PB415ITOSI, 0,
INFO, 1).
    at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2725)
    at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2412)
    at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:371)
    at org.postgresql.jdbc.PgStatement.executeInternal(PgStatement.java:502)
    at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:419)
    at org.postgresql.jdbc.PgPreparedStatement.executeWithFlags(PgPreparedStatement.java:194)
    at org.postgresql.jdbc.PgPreparedStatement.executeUpdate(PgPreparedStatement.java:155)
    at
com.cisco.cgms.influx2postgres.threads.PostgresWriteTask.insertSingleRow(PostgresWriteTask.java:172)

    at
com.cisco.cgms.influx2postgres.threads.PostgresWriteTask.lambda$null$2(PostgresWriteTask.java:156)
    at java.util.stream.ReferencePipeline$2$1.accept(ReferencePipeline.java:174)
    at java.util.ArrayList$ArrayListSpliterator.forEachRemaining(ArrayList.java:1384)
    at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:482)
    at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:472)
    at java.util.stream.ReduceOps$ReduceTask.doLeaf(ReduceOps.java:747)
    at java.util.stream.ReduceOps$ReduceTask.doLeaf(ReduceOps.java:721)
    at java.util.stream.AbstractTask.compute(AbstractTask.java:327)
    at java.util.concurrent.CountedCompleter.exec(CountedCompleter.java:731)
    at java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:289)
    at java.util.concurrent.ForkJoinTask.doInvoke(ForkJoinTask.java:401)
    at java.util.concurrent.ForkJoinTask.invoke(ForkJoinTask.java:734)
    at java.util.stream.ReduceOps$ReduceOp.evaluateParallel(ReduceOps.java:714)
    at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:233)
    at java.util.stream.LongPipeline.reduce(LongPipeline.java:461)
    at java.util.stream.LongPipeline.sum(LongPipeline.java:419)
    at java.util.stream.ReferencePipeline.count(ReferencePipeline.java:593)
    at
com.cisco.cgms.influx2postgres.threads.PostgresWriteTask.lambda$handlePerRowParallel$3(PostgresWriteTask.java:156)

    at java.util.concurrent.ForkJoinTask$AdaptedCallable.exec(ForkJoinTask.java:1424)
```

```
    at java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:289)
    at java.util.concurrent.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1056)
    at java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1692)
    at java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:175)
2025-08-17 12:59:43 ERROR PostgresWriteTask:176 - [ERROR] Exception during individual row insert in
 postgres DB: ERROR: null value in column "net_element_id" of relation "events" violates not-null
constraint
  Detail: Failing row contains (2, null, 1755449042531, 0, asr1000, 891, interfaceUp, GRE tunnel on
 interface Virtual-Access3 connected to 3.3.3.231 c..., 1000, 1000, null, CSR1000V+9PB415ITOSI, 0,
INFO, 1).
org.postgresql.util.PSQLException: ERROR: null value in column "net_element_id" of relation "events"
 violates not-null constraint
  Detail: Failing row contains (2, null, 1755449042531, 0, asr1000, 891, interfaceUp, GRE tunnel on
 interface Virtual-Access3 connected to 3.3.3.231 c..., 1000, 1000, null, CSR1000V+9PB415ITOSI, 0,
INFO, 1).
    ...
2025-08-17 12:59:53 INFO  Influx2PostgresStatsLogger:124 - ##########################################
 INFLUX TO POSTGRES ##########################################
2025-08-17 12:59:53 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events: 1/6 records copied
(16.67%)
2025-08-17 12:59:53 INFO  Influx2PostgresStatsLogger:132 - ##########################################
 INFLUX TO POSTGRES ##########################################

2025-08-17 12:59:53 INFO  Influx2PostgresMigrator:209 - ##### Migration of records from Influx to
Postgres process: Completed 2025-08-17T16:59:53.422Z Total time taken: 00:00:20 #####
```

The migration completes successfully.

FND Debugging — How to Enable

To enable FND debugging, follow these steps:

Option 1:

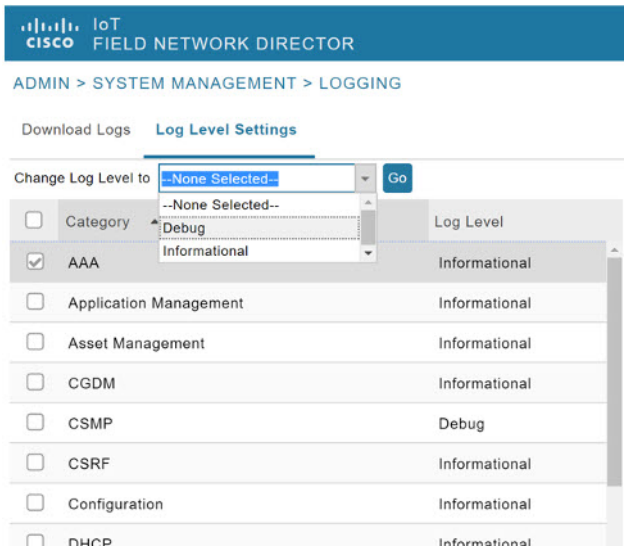
Procedure

Step 1 Choose **ADMIN** > **System Management** > **Logging**.

Step 2 In the screen that appears, select the **Log Level Settings** tab and then choose the **Debug** option from the drop-down menu (such as AAA as shown in Figure 1).

Step 3 Click the **Disk** icon to save (not shown).

Figure 2: Enabling Debug on FND (left-side of the screen)



Step 4 **Option 2:**Choose **ADMIN** > **System Management** > **Logging**.

Step 5 Select the **Log Level Settings** tab.

Step 6 Enter the EIDs for each system such in the debugging panel on the right of the screen (Figure 2) such as:

IR829GW- LTE-GA-EK9+FGL204220HB

See Figure 3.

Step 7 Click the **Disk** icon to save. A separate file is created for each EID in the log location. To locate that file enter the commands below with the relevant EID.

```
[root@iot-fnd ~]# ls /opt/fnd/logs/I*
/opt/fnd/logs/IR829GW-LTE-GA-EK9+FGL204220HB.log
```

Figure 3: Entering EIDs

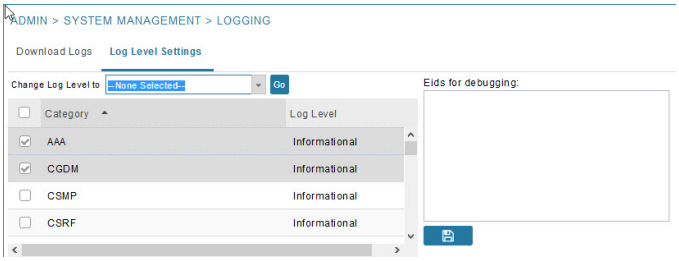
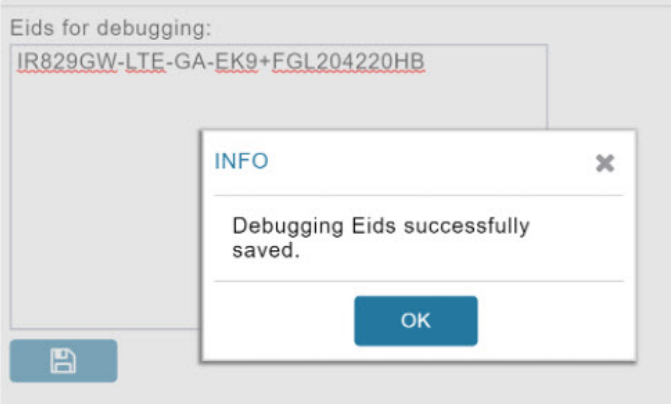


Figure 4: Populated EID panel



FND Debugging — Enable from FND Boot

Before you begin

You can enable debug logging from the start by setting an environment variable or by changing the cgms start script temporarily.

Procedure

Step 1 To start the script, enter: `opt/cgms/bin/cgms`.

Figure 5: Example script for FND Debugging

```
# The CG-NMS Web UI supports enabling debug level logging, but this setting is
# not persisted. When CG-NMS is restarted the logger service will initialize
# the log level to informational. This option instructs the logger service to
# initialize the log level to debug. As most CG-NMS services are dependent upon
# the logger service this option provides a way to obtain debug logs during
# CG-NMS startup.
if [ "x$DEBUG_LOG GING" != "x" ]; then
    JAVA_OPTS="$JAVA_OPTS -Dcom.cisco.cgms.logging.debug"
Fi
```

Step 2 Set DEBUG_LOGGING as non-empty. For example script, see Figure 4.

Java Debugging

Procedure

To determine which JAR file (.jar) is causing issues, add Java option: -verbose:class as shown in the WSMA testscript example below:

```
java -verbose:class -Dlog4j.configuration=file:
$HOME/conf/log4j.properties =Dconf-dire=$HOME/conf
-classpath "$CLASSPATH" com.cisco.cgms.tools.WsmaSimClient "$@"
```

```
[root@iot-fnd ~]# docker exec -it fnd-container /opt/cgms-tools/bin/wsma-
request https://10.48.43.249/wsma/exec fndadmin cisco123
/opt/cgms/server/cgms/conf "show version"
[Opened /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.Object from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.io.Serializable from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.Comparable from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.CharSequence from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.String from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.reflect.AnnotatedElement from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.reflect.GenericDeclaration from /opt/cgms-
tools/jre/lib/rt.jar]
[Loaded java.lang.reflect.Type from /opt/cgms-tools/jre/lib/rt.jar]
[Loaded java.lang.Class from /opt/cgms-tools/jre/lib/rt.jar]
```

Log Files

Note All log files are case-sensitive.

Postgres Log Files:

```
[root@iot-fnd ~]# ls -1 /var/lib/pgsql/9.6/data/pg_log/postgresql-*
/var/lib/pgsql/9.6/data/pg_log/postgresql-Fri.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Mon.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Sat.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Sun.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Thu.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Tue.log
/var/lib/pgsql/9.6/data/pg_log/postgresql-Wed.log
```

For a PostgreSQL install, you can find the log file at:

```
/var/lib/pgsql/9.6/data/pg_log/postgresql-XXX.log
```

where XXX=day, for example XXX = Wed.log.

Note The PostgreSQL version may differ given the FND release and/or OVA release.

FND log files:

You can find the main FND log file at the following path:

`/opt/cgms/server/cgms/logs/server.log`

- For an OVA install, you can find the log file at:
  - `/opt/fnd/logs/server.log`

    points to `/opt/cgms/server/cgms/logs` in the Docker container.
  - `tail -f + grep`

    on serial is often handy as the logs are very verbose.
- For an Oracle install, you can find the log file at:

  `/home/oracle/app/oracle/diag/rdbms/cgms/cgms/trace/alert_cgms.log`

Troubleshoot L+G device metrics refresh

An L+G router or endpoint is present in Cisco IoT FND, but device metrics or telemetry do not refresh. The device may appear reachable, while the `Last Metric Heard` field, charts, or metric values remain stale after a periodic refresh or after selecting `Refresh Metrics`.

Here are the possible causes:

- The L+G device was imported with an inconsistent EID or the EID case does not match the certificate or device profile.
- The device is not in the expected group, such as `Default-lgrouter` for routers or the correct L+G endpoint group for endpoints.
- The periodic metrics URL configured in IoT FND does not match the URL configured on the router or telemetry source.
- A firewall, proxy, NAT device, or load balancer blocks the configured metrics port or terminates long-lived WebSocket traffic.
- The device can register, but periodic metric or telemetry messages are not being sent, are rejected by IoT FND, or are failing during processing.

Before you begin

Use the following procedure to isolate whether the problem is device identity, group assignment, network reachability, IGMA/WebSocket communication, or metric processing.

Procedure

Step 1 From Cisco IoT FND menubar, choose **DEVICES** > **FIELD DEVICES** > **Browse Devices**.

Step 2 Select `ROUTER` for L+G routers or `ENDPOINT` for L+G endpoints.

Step 3 Open the L+G device details and record the EID, IP address, status, configuration group, `Last Heard`, and `Last Metric Heard`.

Step 4 Verify that the device is a member of the expected group. L+G routers should be in `Default-lgrouter` or the intended custom router group; L+G endpoints should be in the expected L+G endpoint group. Verify that the EID uses the same

letter case in the import file, certificate identity, device profile, Cisco IoT FND UI, and logs. If the same device appears with different EID case, correct the import or device identity before troubleshooting metrics.

Step 5 Trigger a manual metric refresh from the device details page and record the timestamp and enable Cisco IoT FND log levels for L+G device debugging:

a) Choose `ADMIN > Logging`. In releases where logging is grouped under system management, choose `ADMIN > System Management > Logging`.
b) Click the `Log Level Settings` tab.
c) Set `Lwm2m` to the required debug log level.
d) Set `Websocket` to the required debug log level.
e) Save the log level settings.
f) Enable EID-specific debug logging for the L+G device if the deployment uses per-EID debug logging.

he L+G device debug details are written to `server.log`.

Step 6 Check the Cisco IoT FND logs on the node that handled the request:

```
RPM deployments: /opt/cgms/server/cgms/logs/server.log
Primary L+G debug log: server.log
```

Step 7 Search server.log for device EID and for IGMA, LWM2M or WebSocket terms around the timestamp of the refresh.

Example:

```
grep -i "<L+G-device-EID>" /opt/cgms/server/cgms/logs/server.log
grep -Ei
"igma|lwm2m|websocket|metric|telemetry|register|timeout|reset|certificate|unauthorized|forbidden"
/opt/cgms/server/cgms/logs/server.log
```

Step 8 Verify the metrics endpoint in `ADMIN > SYSTEM MANAGEMENT > Provisioning Settings`.

- If no metric request reaches Cisco IoT FND, verify DNS, routing, firewall rules, NAT rules, proxy settings, load balancer settings, and device profile URLs from the L+G device network toward Cisco IoT FND.
- If the metric request reaches Cisco IoT FND but the UI remains stale, check `server.log` for parsing, authorization, certificate, database, or transaction errors.

Troubleshoot Metric Interval Errors

Runtime Exception Error

Problem

The error message `java.lang.NullPointerException` is a common runtime exception in Java. It occurs when a program attempts to use an object reference that has not been initialized. Here's an example error message with the logs:

```
java.lang.NullPointerException
fcw2536xabcd][ip=IPv6-Address][sev=ERROR][tid=IOS CGR Request-1][part=64856087.1/9]: Failed
 to processinventory for device. Reason:
64856088: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.2/9]:java.lang.NullPointerException
```

```
64856089: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.3/9]:
atcom.cisco.cgms.elements.ciscocgr.ios.operations.CiscoIosMetricsOperation.getCommandsFromData(CiscoIosMetricsOperation.java:529)
64856090: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.4/9]:
atcom.cisco.cgms.elements.ciscocgr.ios.operations.CiscoIosMetricsOperation.processMetricsData(CiscoIosMetricsOperation.java:606)
64856091: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.5/9]:
atcom.cisco.cgms.elements.ciscocgr.ios.operations.CiscoIosMetricsOperation.processInventoryUpload(CiscoIosMetricsOperation.java:416)
64856092: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.6/9]:
atcom.cisco.cgms.elements.ciscocgr.ios.operations.CiscoCgrMetricsProcess.run(CiscoCgrMetricsProcess.java:128)
64856093: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.7/9]:
atjava.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
64856094: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.8/9]:
atjava.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
64856095: fnd-server: Aug 30 2024 14:17:23.357 +0000: %IOTFND-3-UNSPECIFIED:
%[ch=CiscoCgrMetricsProcess][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=ERROR][tid=IOS
 CGR Request-1][part=64856087.9/9]:
atjava.lang.Thread.run(Thread.java:748)
64856096: fnd-server: Aug 30 2024 14:17:23.358 +0000: %IOTFND-7-UNSPECIFIED:
%[ch=TxConnectionListener][eid=IR1101-K9+fcw2536abcd][ip=IPv6-Address][sev=DEBUG][tid=IOS
CGR Request-1]: Unfinished local transaction wasrolled
back.org.jboss.jca.core.connectionmanager.listener.TxConnectionListener@39f31201[state=NORMAL
 managedconnection=org.jboss.jca.adapters.jdbc.local.LocalManagedConnection@568ee3c connection
 handles=0lastReturned=1725027443340 lastValidated=1725027054869 lastCheckedOut=1725027443341
 trackByTx=falsepool=org.jboss.jca.core.connectionmanager.pool.strategy.PoolBySubject@4c313ad0
mcp=SemaphoreConcurrentLinkedQueueManagedConnectionPool@216c3b91[pool=postgresDS]
xaResource=LocalXAResourceImpl@50f733b7[connectionListener=39f31201 connectionManager=9380d48
 warned=false
currentXid=null productName=PostgreSQL productVersion=12.12 jndiName=java:/DefaultDS]
txSync=null]
```

Solution

The router is onboarded using a lowercase EID (eid=IR1101-K9+fcw2536abcd), while the router certificate is generated with an uppercase EID (eid=IR1101-K9+FCW2536ABCD). This discrepancy causes a HASH mismatch, resulting in a `java.lang.NullPointerException`. For more information see,CSCwm39996. The following steps can help avoid the common runtime exception:

- Check the refresh metrics from Cisco IoT FND periodically.
- Trigger periodic metric profiles from the router with debugging enabled on both Cisco IoT FND and the router.
- Perform system, metric, and router bootstrapping debugs. Use the `debug cgna logging all` command to enable comprehensive logging.
- Ensure that the EID is entered consistently in all configurations and interfaces using the same case format.

File Upload Error

Problem

The `com.cisco.cgms.exceptions.CiscoIosFileUploadException` is an exception that occurs in Cisco IoT FND, when there is an issue related to file uploads, particularly involving Cisco IOS files. Here's an example error message:

```
com.cisco.cgms.exceptions.CiscoIosFileUploadException: Error occurred while verifying file
 upload operation for net element
IR1101-K9+FCW2425ABCD; Caused by: com.cisco.cgms.exceptions.CiscoIosFileUploadException:
Exception occured while
performing file upload operation for net element IR1101-K9+FCW2425ABCD; Caused by:
com.cisco.cgms.exceptions.CiscoIosFileUploadException: Error occured while performing file
 upload operation; Caused by:
com.cisco.cgms.wsma.client.CgmsWsmaException: Error in running file check
```

Solution

While onboarding routers, when the CA server validation or authentication fails, the error message `CiscoIosFileUploadException` appears. Here's how you can troubleshoot and resolve the issue:

- **Check the Router's Certificate Details**: Use the command `show crypto pki certificate verbose LDevID` on the router to display detailed information about the Local Device Identifier (LDevID) certificate. This information can help identify discrepancies or issues in the certificate.
- **Verify the SAN Field**: Ensure that the SAN field in the certificate is correctly populated. The SAN field should match the expected values required for successful validation.
- **Use Debug Commands on the Router**: The following debug commands can provide detailed insights into the certificate-related processes and help diagnose the cause of the failure:
  - **`debug crypto pki messages`**: Captures general messages related to Public Key Infrastructure (PKI) operations.
  - **`debug crypto pki scep`**: Provides information specific to the Simple Certificate Enrollment Protocol (SCEP), which is often used for certificate issuance and renewal.
  - **`debug crypto pki transactions`**: Logs all PKI transaction details to identify any issues during certificate processing.
  - **`debug crypto pki validation`**: Offers insights into the validation process of certificates, helping identify why a certificate might be considered invalid.
  - **`debug crypto pki server`**: If applicable, provides details on the PKI server interactions, which can be useful if the router acts as a CA.
- **Additional Considerations:**
  - Ensure network connectivity between the router and the CA server.
  - Verify that all configurations related to certificate enrollment and PKI are correct on both the router and the FND.
  - Consult Cisco support or documentation for specific guidance related to the version of the software in use if the issue persists.

Missing Configuration Error

Problem

The `com.cisco.cgms.properties.PropertyNotFoundException` is an exception indicating that a specific property or configuration setting expected by Cisco IoT FND could not be found. This exception typically arises when a required configuration parameter is missing or incorrectly specified within the system properties files or settings. Here is an example error message:

```
com.cisco.cgms.properties.PropertyNotFoundException: Required property [Administrator
Password(adminPassword)] has
not been configured on [IR1101-K9+FCW2425ABCD]
```

Solution

The `com.cisco.cgms.properties.PropertyNotFoundException` appears when the admin password of Cisco IoT FND doesn't match with the admin password of the router. Update the password for Cisco IoT FND in the CSV file that matches with the password of the router.

Retrieve Inventory Error

Problem

The error message `java.io.IOException: Failed to retrieve inventory from device. Reason: \[invalid cli command\]` indicates that there was an attempt to execute a command on a network device, but the command was not recognized or valid, leading to a failure in retrieving inventory information. Here's an example error message:

```
java.io.IOException: Failed to retrieve inventory from device. Reason: [invalid cli command]
 Sent [[show version | format
flash:/managed/odm/cg-nms.odm
show iox-service | format flash:/managed/odm/cg-nms.odm
show platform resources | format flash:/managed/odm/cg-nms.odm]]
```

Solution

When comparing the inventory from a router with the `cg-nms.odm` file, discrepancies during the bootstrapping process can lead to errors. Look at the specifics provided in the error message to identify which parts of the configuration are inconsistent. Adjust the router configuration to match the expectations defined in the `cg-nms.odm` file, if appropriate.

SOAP Error

Problem

The `javax.xml.ws.soap.SOAPFaultException` is an exception that occurs during Simple Object Access Protocol (SOAP) web service communication. This exception indicates that there was a problem processing a SOAP message, and a fault was returned by the web service.

Solution

To address connectivity issues between the Cisco IoT FND and a router, you can perform the following steps to isolate and diagnose the problem:

1. Use the ping command to test basic network connectivity between the Cisco IoT FND and the router. This helps determine if there is a reachable network path.
2. Use the following command to execute a Web Services Management Agent (WSMA) request to the router, checking if the WSMA communication is functioning correctly:

```
/opt/cgms-tools/bin/wsma-request https://<ROUTER IPv4 or IPv6 address>:443/wsma/exec
admin
          /opt/cgms/server/cgms/conf "show version | format
        flash:/managed/odm/cg-nms.odm"
```

3. Use `tcpdump --list-interfaces` to list available network interfaces on the Cisco IoT FND, helping you identify the correct interface for capturing traffic.
4. Use `tcpdump -i ens192 -w cisco.pcap` to capture network packets on the specified interface and write them to a file named `cisco.pcap`. Analyzing this file can help identify anomalies in network traffic.
5. The following debug commands on the router can help identify failures related to WSMA or HTTP communication:
   - **`debug cgna logging all`**: Enable comprehensive logging for Cisco Generic Network Architecture (CGNA) to capture detailed logs.
   - **`debug wsma agent`**: Monitor the WSMA agent for any issues in handling requests.
   - **`debug wsma profile`**: Check the WSMA profile configurations and operations.
   - **`debug ip http all`**: Capture all HTTP-related activities, which can be useful if WSMA over HTTP is involved.
6. Enter configuration mode with `conf t` and set the environment variable for logging configuration replacements with:

```
event manager environment LOG_CONFIG_REPLACE TRUE
exit
```

Timeout Error

Problem

The error message you're encountering involves a `javax.xml.ws.WebServiceException` and a `java.net.SocketTimeoutException`, indicating a timeout issue during a web service communication attempt. High latency in the network can cause timeouts if the response from the server takes too long.

Solution

To troubleshoot and isolate connectivity issues between the Cisco IoT FND and a router, particularly when dealing with WSMA or HTTP-related problems, you can perform the following steps:

1. Use the `ping` command to test basic connectivity from the Cisco IoT FND to the router.
2. Execute the following command to test WSMA communication:

```
/opt/cgms-tools/bin/wsma-request https://<ROUTER IPv4 or IPv6 address>:443/wsma/exec
admin /opt/cgms/server/cgms/conf "show version | format flash:/managed/odm/cg-nms.odm"
```

The output will help you understand if WSMA is functioning properly.

3. Run `tcpdump --list-interfaces` to see all available network interfaces on the FND. This is useful for identifying the correct interface to monitor.
4. Use the following command to capture network packets on a specific interface (e.g., `ens192`) and save them to a file (`cisco.pcap`):

```
tcpdump -i ens192 -w cisco.pcap
```

5. The following debug commands can provide insights into WSMA and HTTP operations on the router:
   - **`debug cgna logging all`**: Enables comprehensive logging for Cisco Generic Network Architecture (CGNA) to capture detailed logs.
   - **`debug wsma agent`**: Monitors the WSMA agent for any issues or errors during request handling.
   - **`debug wsma profile`**: Examines WSMA profile configurations and operations to identify misconfigurations.
   - **`debug ip http all`**: Captures all HTTP-related activities, useful if WSMA operates over HTTP.
6. Enter configuration mode and set an environment variable to log configuration changes:

```
conf t
event manager environment LOG_CONFIG_REPLACE TRUE
exit
```

Network Connectivity Errors

Problem

The following error messages indicate various network connectivity issues encountered during SOAP web service communication.

1. **Read Timed Out**: This error occurs when a connection is established, but the server does not send a response within the expected time frame. Here's an example:

```
javax.xml.ws.soap.SOAPFaultException: Read timed out; Caused by:
com.ctc.wstx.exc.WstxIOException: Read timed out
```

2. **Connect Timed Out**: This error indicates that the client was unable to establish a connection to the server within the allotted time. Here's an example:

```
javax.xml.ws.soap.SOAPFaultException: connect timed out; Caused by:
com.ctc.wstx.exc.WstxIOException: connect timed
out
```

3. **Connection Reset**: This occurs when an established connection is unexpectedly closed by the server. Here's an example:

```
javax.xml.ws.soap.SOAPFaultException: Connection reset; Caused by:
com.ctc.wstx.exc.WstxIOException: Connection reset
```

4. **Connection Refused**: This error means the server actively refused the connection attempt, often because the service is not running on the specified port. Here's an example:

```
javax.xml.ws.soap.SOAPFaultException: Connection refused (Connection refused); Caused
by:
com.ctc.wstx.exc.WstxIOException: Connection refused (Connection refused)
```

Solution

To effectively troubleshoot connectivity and communication issues between the Field Network Director (FND) and a router, you can perform the following diagnostic steps:

1. Perform a ping test from both Cisco IoT FND and the router.
2. Execute the following WSMA request command to test the ability to execute management commands on the router:

   ```
   /opt/cgms-tools/bin/wsma-request https://<ROUTER IPv4 or IPv6 address>:443/wsma/exec
   admin /opt/cgms/server/cgms/conf "show version | format flash:/managed/odm/cg-nms.odm"
   ```

3. Use `tcpdump --list-interfaces` to list all available network interfaces on the FND. This helps you identify which interface to monitor for packet capture.
4. Run `tcpdump` on the identified network interface to capture packets and write them to a file for analysis:

   ```
   tcpdump -i ens192 -w cisco.pcap
   ```

5. The following debug commands help identify failures related to WSMA or HTTP communication:
   - **`debug cgna logging all`**: Enables detailed logging for Cisco Generic Network Architecture, capturing extensive logs.
   - **`debug wsma agent`**: Monitors the WSMA agent for potential issues or errors in handling requests.
   - **`debug wsma profile`**: Examines WSMA profile configurations and operations to find misconfigurations.
   - **`debug ip http all`**: Captures all HTTP-related activities, useful for diagnosing HTTP-based WSMA issues.
6. Enter configuration mode and set an environment variable to log configuration changes, which can help track and diagnose issues:

   ```
   conf t
   event manager environment LOG_CONFIG_REPLACE TRUE
   exit
   ```

Webservice Error

Problem

The `javax.xml.ws.WebServiceException: Could not send Message` indicates a problem with sending a SOAP message to a web service endpoint, which resulted in an HTTP 504 Gateway Timeout error. This typically means that a server, acting as a gateway or proxy, did not receive a timely response from an upstream server it needed to access to complete the request. Here's an example error message:

```
javax.xml.ws.WebServiceException: Could not send Message.; Caused by:
org.apache.cxf.transport.http.HTTPException:
HTTP response '504: Gateway Time-out' when communicating with https://[IPV6 address / IPV4
 Address]:443/wsma/exec
```

Solution

The issue may be related to an invalid or improperly configured self-signed certificate, which can cause communication problems, especially in secure network environments. Self-signed certificates are often used for testing or internal purposes, but they need to be correctly configured and trusted by the systems that interact with them. Here are the steps to address the issue:

1. Ensure the current self-signed certificate is valid. If it's expired or improperly configured, re-generate it following the appropriate process for your device or system.
2. The following commands are used on network devices to manage HTTP and SSH services. They first disable and then re-enable these services, potentially refreshing or resetting their configurations:
   - **`no ip http server`**: Disables the HTTP server on the device.
   - **`no ip http authentication local`**: Removes local authentication for HTTP, if previously set.
   - **`no ip http secure-server`**: Disables the HTTPS server, which might be necessary before making changes to certificates.
   - **`no ip ssh version 2`**: Disables SSH version 2; however, this step might be intended to reset SSH configurations.
   - **`ip http server`**: Enables the HTTP server on the device.
   - **`ip http authentication local`**: Sets local authentication for HTTP access.
   - **`ip http secure-server`**: Enables the HTTPS server, allowing secure web-based management.
   - **`ip ssh version 2`**: Re-enables SSH version 2, which is the more secure version of the SSH protocol.
3. After re-generating the self-signed certificate, ensure that any clients or systems interacting with this device trust the certificate. This might involve importing the certificate into their trust store.

SSL Debugging

Procedure

Set DEBUG_SSL to ‘true’ in `/opt/bin/cgms/bin/cgms.conf` as shown in the steps below:

```
[root@fnd bin]# cat opt/cgms/bin/cgms.conf
MAX_JAVA_HEAP_SIZE=8g
DEBUG_SSL=true
[root@fnd bin] service cgms restart
```

Troubleshoot High CPU Issues

Problem

Often facing a spike in the CPU usage.

Solution

Here are the instructions to analyze and troubleshoot some of the high CPU issues:

1. Analyze if the CPU spike is intermittent or constant. If intermittent, make a note of the duration.
2. From the Cisco IoT FND Menubar, choose **Devices** > **Servers**.
3. Choose the problematic Cisco IoT FND server which has CPU issues and analyze the charts to spot any CPU spike issues.
4. Run the following commands as root using the SSH command:

Command	Description
lscpu	Gathers CPU architecture information from sysfs and /proc/cpuinfo, information includes number of CPUs, threads, cores, sockets etc.
ps-aux	View statistics about system's running processes, like the percent of CPU and memory that the process is using etc
ps aux k-pcpu | head -6	This command sorts the list of running processes by CPU usage in descending order (using `k-pcpu`) and then displays the top 6 processes (`head -6`). It's useful for identifying which processes are consuming the most CPU.
ps aux | sort -nrk 3,3 | head -n 5	This command sorts the running processes by the third column (which is typically %CPU) in numerical reverse order (`-nrk 3,3`) and displays the top 5 entries (`head -n 5`). Similar to the previous command, it helps identify CPU-intensive processes.
top	View a real-time view of running processes in Linux and displays kernel-managed tasks, enables user to monitor CPU-intensive tasks
top -b -n 1 | head -n 12 | tail -n 5	This command runs `top` in batch mode (`-b`) for one iteration (`-n 1`). It then takes the first 12 lines of the output and displays the last 5 lines (`head -n 12 | tail -n 5`). This typically shows the header information and the most CPU-intensive processes at a given moment.

Command	Description
sar -t -r -f	The `sar` command is used for collecting, reporting, or saving system activity information. The `-r` flag is for memory usage, and `-t` specifies time format. The `-f` flag is used to read data from a file, typically a `sar` data file. This would provide historical data on memory usage over time.
sar -t -q -f	Similar to the previous `sar` command, but with the `-q` flag, which is used to report queue length and load average. This helps in understanding how the system load has been distributed over time.
iostat	Monitor CPU utilization, system input/output statistics for all the disks and partitions
free -m	Get detailed report on the system's memory usage
mpstat	View information about CPU utilization and performance
vmstat	View information about processes, memory, disk, and CPU activity

Here's a sample output:

```
ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 191428 4332 ? Ss Jul09 20:48 /usr/lib/systemd/systemd --switched-roo
root 2 0.0 0.0 0 0 ? S Jul09 0:00 [kthreadd]
root 4 0.0 0.0 0 0 ? S< Jul09 0:00 [kworker/0:0H]
root 6 0.0 0.0 0 0 ? S Jul09 3:22 [ksoftirqd/0]
root 7 0.0 0.0 0 0 ? S Jul09 0:06 [migration/0]
root 8 0.0 0.0 0 0 ? S Jul09 0:00 [rcu_bh]
root 9 0.0 0.0 0 0 ? R Jul09 59:17 [rcu_sched]
root 10 0.0 0.0 0 0 ? S< Jul09 0:00 [lru-add-drain]
root 11 0.0 0.0 0 0 ? S Jul09 0:49 [watchdog/0]
root 12 0.0 0.0 0 0 ? S Jul09 0:17 [watchdog/1]
root 52 0.0 0.0 0 0 ? S Jul09 0:20 [watchdog/9]
root 53 0.0 0.0 0 0 ? S Jul09 0:19 [migration/9]
root 54 0.0 0.0 0 0 ? S Jul09 0:28 [ksoftirqd/9]
root 56 0.0 0.0 0 0 ? S< Jul09 0:00 [kworker/9:0H]
```

Gather the Heap and Thread Dumps

When you notice slowness in your Cisco IoT FND, collect the JAVA heap and thread dumps. Use the following instructions to collect the heap and thread dumps:

1. Download the JDK package which is an OpenJDK distribution for Linux (64-bit) from https://corretto.aws/downloads/latest/amazon-corretto-8-x64-linux-jdk.tar.gz.
2. Extract the JDK tar.gz file.

   For example:

   ```
   tar -xzf amazon-corretto-8-x64-linux-jdk.tar.gz
   ```

3. Transfer the extracted JDK file to the Cisco IoT FND server using the scp command.

   For example:

   ```
   scp -r amazon-corretto-8-x64-linux-jdk user@fnd-server:/path/to/destination
   ```

4. Navigate to the `bin` directory of the JDK installation to access the `jmap` utility. Example:

   ```
   cd /path/to/destination/amazon-corretto-8-x64-linux-jdk/bin
   ```

5. Find the docker ID using the `docker ps` command. Here's an example:

   ```
   [root@iot-fnd ~]# docker ps
   CONTAINER ID        IMAGE               COMMAND                  CREATED
   STATUS              PORTS

                                                                          NAMES
   54e349b30f46        fogd-image:active   "/bin/sh -c /usr/loc…"   12 days ago
   Up 12 days          443/tcp

                                                                          fogd-container
   f67d89e4188b        fnd-image:active    "/bin/sh -c /opt/fnd…"   12 days ago
   Up 2 days           0.0.0.0:80->80/tcp, 0.0.0.0:162->162/udp, 0.0.0.0:443->443/tcp,
   0.0.0.0:9120-9121->9120-9121/tcp, 0.0.0.0:5683->5683/udp,
   0.0.0.0:61624-61626->61624-61626/udp, 0.0.0.0:9124-9125->9124-9125/tcp,
   0.0.0.0:61628->61628/udp   fnd-container
   ```

6. Copy the tar.gz file to the docker using the following command and paste the docker ID:

   ```
   docker cp /amazon-corretto-8.422.05.1-linux-x64.tar.gz f67d89e4188b
   ```

7. Log in to the docker using the following command:

   ```
   [root@iot-fnd ~]# docker exec -i -t fnd-container /bin/bash
   ```

8. Unzip the tar.gz file using the `tar-xvf<complete path>` command. Here's an example:

   ```
   [root@iot-fnd ~]# tar -xvf /CISCO/amazon-corretto-8.422.05.1-linux-x64.tar.gz]
   ```

9. Find the CGMS process ID using the `ps -eaf | grep cgms` command. Here's an example:

   ```
   [root@iot-fnd ~]# ps -eaf | grep cgms
   root      308261       1  0 Jul29 pts/0    00:00:00 runuser -c bin/cgms -s /bin/bash
   root
   root      308262  308261  0 Jul29 pts/0    00:00:00 /bin/sh bin/cgms
   root      308265  308262  0 Jul29 pts/0    00:00:00 /bin/sh bin/standalone.sh
   -Djboss.server.log.dir=bin/../server/cgms/log --server-config=standalone.xml -b 0.0.0.0
   root      308382  308265  2 Jul29 pts/0    05:55:42 java -D[Standalone] -verbose:gc
   -Xloggc:/opt/cgms/server/cgms/log/gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps
   -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=3M
   -XX:-TraceClassUnloading -Xms1g -Xmx6g -XX:MaxPermSize=512m
   -Dcom.cisco.cgms.ciscolog.host=fnd-server -server -XX:+HeapDumpOnOutOfMemoryError
   -XX:HeapDumpPath=bin/../server/cgms/log -Djava.security.egd=file:/./dev/urandom
   -XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
   -XX:LogFile=bin/../server/cgm/log/cgms_stacktrace.log -XX:-OmitStackTraceInFastThrow
   -Dorg.terracotta.quartz.skipUpdateCheck=true -Dbase.dir=bin
   -Dorg.jboss.boot.log.file=/opt/cgms/server/cgms/log/server.log
   -Dlogging.configuration=file:/opt/cgms/standalone/configuration/logging.properties
   -jar /opt/cgms/jboss-modules.jar -mp /opt/cgms/modules org.jboss.as.standalone
   -Djboss.home.dir=/opt/cgms -Djboss.server.base.dir=/opt/cgms/standalone
   -Djboss.server.log.dir=bin/../server/cgms/log --server-config=standalone.xml -b 0.0.0.0
   root      430277  430190  0 17:34 pts/1    00:00:00 grep --color=auto cgms
   ```

10. Use the following command to generate the heap dump:

    ```
    [root@iot-fnd ~]# /CISCO/amazon-corretto-8.422.05.1-linux-x64/bin/jmap
    -dump:file=heap_dump_11082024.hprof 308382
    ```

```
Dumping heap to /ENEDIS/heap_dump_11082024.hprof ...
Heap dump file created
[root@fnd-server CISCO]#
[root@fnd-server CISCO]#
[root@fnd-server CISCO]# ls -l | grep -i hprof
-rw------- 1 root root 1148131128 Aug 11 11:34 heap_dump_11082024.hprof
```

11. Use the following command to generate the thread dump:

```
[root@iot-fnd ~]# kill -3<CGNMSpid>
```

12. Upload both the cgms_stacktrace.log to TAC SR for further analysis.

Zero Touch Deployment — Tunnel Provisioning

```
Received tunnel provisioning request from [IR1101-K9+FCW22520078]
Adding tunnel provisioning request to queue for FAR ID=
Provisioning tunnels on element [IR1101-K9+FCW22520078]
Retrieved current configuration of element [IR1101-K9+FCW22520078] before tunnel provisioning
Retrieved status of file [flash:/before-registration-config] on [IR1101-K9+FCW22520078].
File does not
exist
Retrieved status of file [flash:/before-tunnel-config] on [IR1101-K9+FCW22520078]. File
does not exist.
Copied running-config of [IR1101-K9+FCW22520078] to [flash:/before-tunnel-config]
Opened a NETCONF session with element [HTABT-TGOT-DC-RT1] at [163.88.181.2]
Sending [show interfaces | include Description: | Encapsulation | address is | line protocol
 | packets
input, | packets output, | Tunnel protection | Tunnel protocol| Tunnel source] to element
[HTABT-TGOT-DC-RT1]
Received response to [show interfaces | include Description: | Encapsulation | address is
| line
protocol | packets input, | packets output, | Tunnel protection | Tunnel protocol| Tunnel
source] from
element [HTABT-TGOT-DC-RT1]
Sending [show ip nhrp | include ^[0-9A-F]| Tunnel| NBMA] to element [HTABT-TGOT-DC-RT1]
Received response to [show ip nhrp | include ^[0-9A-F]| Tunnel| NBMA] from element
[HTABT-TGOT-DC-RT1]
Sending [show ipv6 nhrp | include ^[0-9A-F]| Tunnel| NBMA] to element [HTABT-TGOT-DC-RT1]
Received response to [show ipv6 nhrp | include ^[0-9A-F]| Tunnel| NBMA] from element
[HTABT-TGOT-DC-RT1]
Sending [show ipv6 interface | include address | protocol | subnet] to element
[HTABT-TGOT-DC-RT1]
Received response to [show ipv6 interface | include address | protocol | subnet] from element
[HTABT-TGOT-DC-RT1]
Closed NETCONF session with element [HTABT-TGOT-DC-RT1]
Obtained current configuration of element [HTABT-TGOT-DC-RT1] before tunnel provisioning
Configured tunnels on [IR1101-K9+FCW22520078]
Retrieved current configuration of element [IR1101-K9+FCW22520078] after tunnel provisioning.
Processed tunnel template for element [ASR1001+93UA2TVWZAR]. Time to process [5 ms].
Configured element [IR1101-K9+FCW223700AG] to register with IoT-FND at
[https://10.48.43.229:9121/cgna/ios/registration]
-OR -
Tunnel provisioning request for element [IR1101-K9+FCW22520078] failed
```

ZTD Easy Mode for PNP

```
[UPDATING_ODM]
[COLLECTING_INVENTORY]
```

```
[VALDIATING_CONFIGURATION]
[PUSHING_BOOTSTRAP_CONFID_FILE]
[CONFIGURING+STARTUP_CONFIG]
[APPLYING_CONFIG]
[TERMINATING_BS_PROFILE]
[BOOTSTRAP_DONE]
```

Zero Touch Deployment Steps — Log Entries for Plug and Play

```
Received pnp request from [IR1101-K9+FCW22520078]
state: NONE
state: CONFIGURING_HTTP_FOR_SUDI
state: CONFIGURED_HTTP_FOR_SUDI
state: CREATING_FND_TRUSTPOINT msgType: PNP_GET_CA
state: CREATING_FND_TRUSTPOINT msgType: PNP_WORK_REQUEST
state: AUTHENTICATING_WITH_CA
state: AUTHENTICATED_WITH_CA
state: UPDATING_TRUSTPOINT
state: UPDATED_TRUSTPOINT
state: UPDATING_ODM msgType: PNP_GET_ODM
state: UPDATING_ODM msgType: PNP_WORK_RESPONSE
state: UPDATING_ODM_VERIFY_HASH msgType: PNP_WORK_REQUEST
state: UPDATING_ODM_VERIFY_HASH msgType: PNP_WORK_RESPONSE
state: UPDATED_ODM msgType
state: COLLECTING_INVENTORY
state: COLLECTED_INVENTORY
state: VALIDATING_CONFIGURATION
state: VALIDATED_CONFIGURATION
state: PUSHING_BOOTSTRAP_CONFIG_FILE msgType: PNP_GET_BSCONFIG
state: PUSHING_BOOTSTRAP_CONFIG_FILE msgType: PNP_WORK_RESPONSE
state: PUSHING_BOOTSTRAP_CONFIG_VERIFY_HASH msgType: PNP_WORK_REQUEST
state: PUSHING_BOOTSTRAP_CONFIG_VERIFY_HASH msgType: PNP_WORK_RESPONSE
state: PUSHED_BOOTSTRAP_CONFIG_FILE
state: CONFIGURING_STARTUP_CONFIG
state: CONFIGURED_STARTUP_CONFIG
state: RELOADING
Updating PnP state to: [BOOTSTRAP_DONE]
[eid=IR1101-K9+FCW22520078][ip=91.91.91.10][sev=INFO][tid=tunnelProvJetty-263]: Status
updated
to:[bootstrapped]
```

ZTD Step by Step — Entries for IXM Registration

```
Got IGMA POST with authtype: CLIENT_CERT
Received registration request for LoRaWAN Gateway with eid: [IXM-LORA-800-H-V2+FOC20133FJQ]
Executing registration request for LoRaWAN Gateway with EID: [100082].Processing LoRa Gateway
Registration Request
Processing LoRaWAN Gateway Command...
Tunnel1 Ip and/or prefix not received from LoRa Gateway. Tunnel Ip may not be updated
properly.
Tunnel2 Ip and/or prefix not received from LoRa Gateway. Tunnel Ip may not be updated
properly.
Processed LoRaWAN Gateway Command...
Processing LoRa Gateway Configuration
Processing Post Configuration
Processing Packet Forwarder Installation
Processed Packet Forwarder Installation
LoRaWAN Gateway Registration Process Complete
```

ZTD Step by Step — Log Entries for IXM Tunnel

```
Received Tunnel Prov Request for LoRaWAN Gateway with eid: [IXM-LORA-800-H-V2+FOC20133FJQ]
Checking if file:[before-registration-config] exist. Delete if Present. Tunnel Reprovisioning
 Request
File [before-tunnel-config] not found on the element. Creating the file.
Processed LoRaWAN Gateway Tunnel Provisioning
```

ZTD Step by Step — Log Entries for Registration

```
Received registration request from element: [IR1101-K9+FCW22520078]
Element IR1101-K9+FCW22520078 is running supported firmware version 16.10.01.
Continuing with element configuration
Retrieved status of file [flash:/before-registration-config] on [IR1101-K9+FCW22520078].
File does not
exist.
Copied running-config of [IR1101-K9+FCW22520078] to [flash:/before-registration-config]
Successfully deactivated the cgna registration profile and copied the running-config to
start-up config
for the element IR1101-K9+FCW22520078
Completed configuration of element [IR1101-K9+FCW22520078]
Registration phase completed for element [IR1101-K9+FCW22520078]
```

Upgrading the Virtual Machine

Before you begin

- Create a backup or snapshot of the virtual machines.
- Make sure the .vmdk files are available to the ESXi host on VMFS3, VMFS5, and NFS datastore.
- Make sure the virtual machine is stored on VMFS3, VMFS5, or NFS datastores.
- Make sure the compatibility settings for the virtual machines are not set to the latest supported version.
- Determine the ESXi versions that you want the virtual machines to be compatible with.

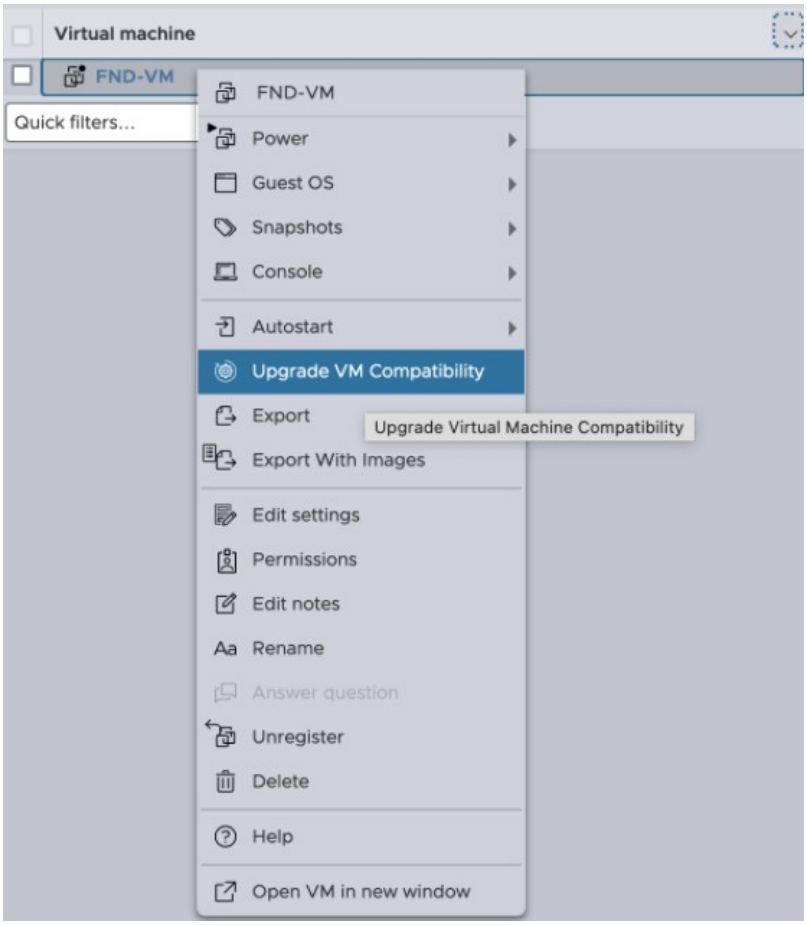
Note In Cisco IoT FND running Cisco IoT FND Release 4.12 and earlier releases, the maximum vCPU count is limited to 32 due to virtual machine compatibility being set to ESXi 5.0. Starting from Cisco IoT FND 5.0 version onwards, you can increase the CPU count beyond 32.

In case you upgrade your Cisco IoT FND from an older release to a newer release, follow the steps below to upgrade virtual machine compatibility.

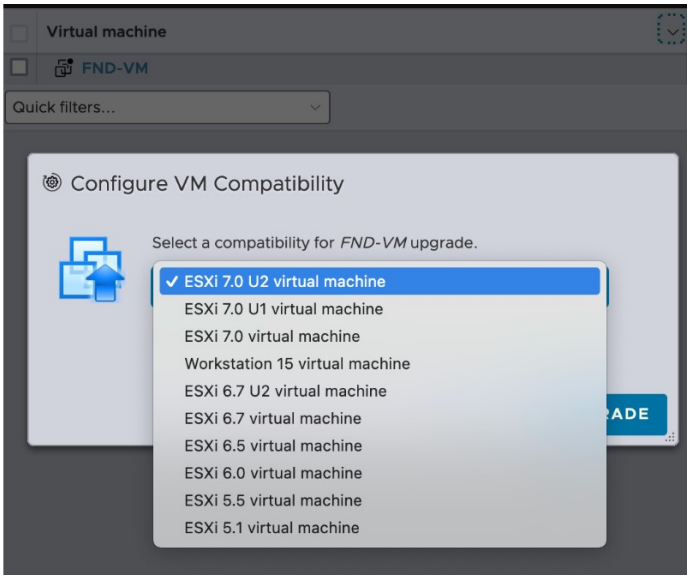
Procedure

Step 1 Power off the virtual machine.

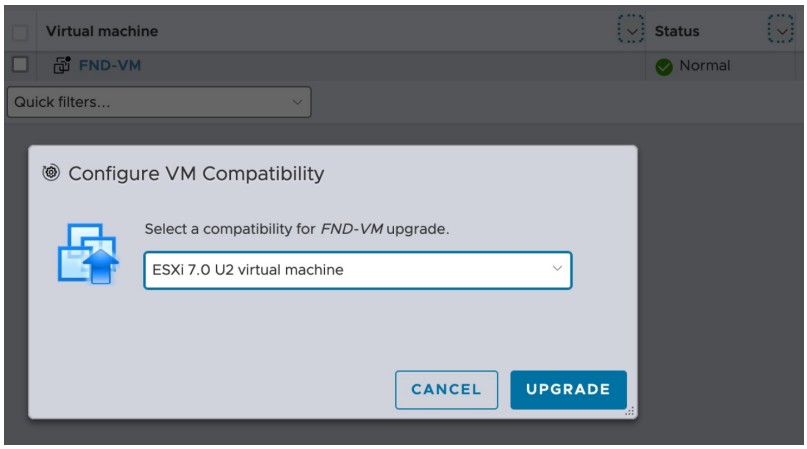
Step 2 Right-click the virtual machine from the **Virtual machine** drop down list and select **Upgrade VM Compatibility** from the popup menu.

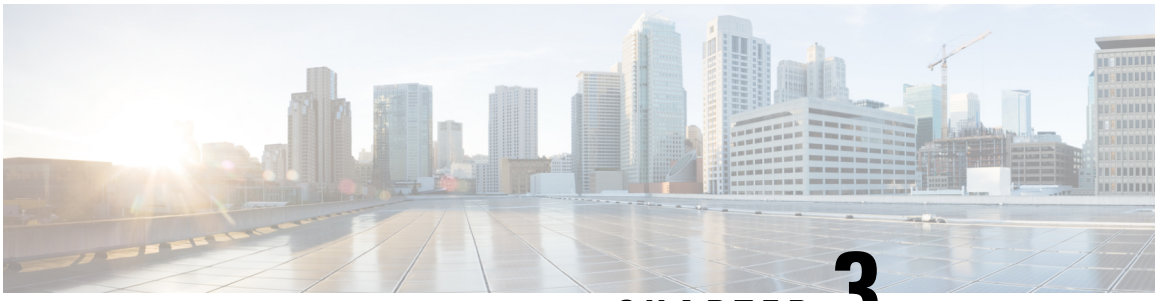


Step 3 Choose the latest supported version from the drop-down box.



Step 4 Cick **Upgrade**.





CHAPTER 3

Troubleshooting for Specific IoT FND Components

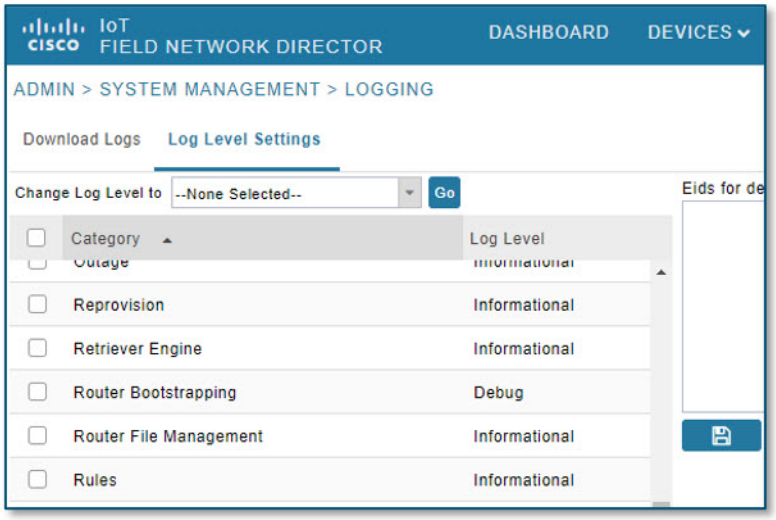
This chapter explains some of the component-specific IoT FND issues and possible resolutions.

- Troubleshoot PNP, on page 44
- Debug IGMA websocket sessions for L+G border routers, on page 45
- Troubleshooting Steps to Upload ODM File, on page 47
- Troubleshoot TCL Scripts, on page 59
- Troubleshoot Certificate Enrollment, on page 59
- Certificate Enrollment — Test Manual, on page 60
- Certificate Enrollment — Example Output, on page 60
- Troubleshoot WSMA, on page 61
- Troubleshoot Tunnel Provisioning, on page 62
- Troubleshoot Netconf: FND—HER Communications, on page 62
- Troubleshoot Configuration Deployment, on page 64
- Troubleshoot HSM Connectivity, on page 64
- Issues Faced During HSM Client Upgrade, on page 65

Troubleshoot PNP

Procedure

Figure 6: ADMIN > SYSTEM MANAGEMENT > LOGGING > Log Level Settings



Step 1 Check the FND-server logs by doing the following:

a. Increase the log level: Choose **ADMIN** > **SYSTEM MANAGEMENT** > **LOGGING**.

b. Select the **Log Level Settings** tab.

c. Select the box next to the **Router Bootstrapping** option; and, select the **Debug** option from the **Change Log Level to** drop-down menu.

d. Click **Go**.

You can find the generated logs in the following location:

```
opt/cgms/server/cgms/logs/server.log (RPM) and opt/fnd/logs/server.log (OVA)
```

Step 2 Debug on FAR by entering the following commands:

```
debug pnp
debug ip http client
```

Step 3 Check certificates and the ‘fnd’ trustpoint.

Step 4 Check provisioning link in settings.

Step 5 Check archive configuration and directory.

Debug IGMA websocket sessions for L+G border routers

An L+G border router cannot complete registration, metrics refresh, tunnel-related messaging, firmware-related transfer, or another operation that uses IGMA or a WebSocket-backed communication path.

IGMA profiles identify the Cisco IoT FND registration, metrics, and tunnel endpoints that a device uses for Cisco IoT FND communication. When WebSocket is used, the device and Cisco IoT FND establish an HTTP or HTTPS session and then use the same TCP connection for bidirectional communication. If a firewall, proxy, load balancer, certificate, URL, or idle-timeout setting interrupts that session, Cisco IoT FND might show stale metrics, incomplete registration, failed file transfer, or no visible response to a device operation.

Use this procedure when an L+G border router loses connectivity to Cisco IoT FND, stops reporting metrics, or experiences intermittent disconnects.

Procedure

Step 1 Identify the failing symptom:

Example:

- The device loses connectivity to Cisco IoT FND.
- The device stops reporting telemetry.
- The device repeatedly disconnects and reconnects.
- Registration, metrics refresh, tunnel messaging, configuration push, or firmware transfer fails.

Step 2 Confirm network connectivity between the L+G border router and Cisco IoT FND:

Example:

- Verify that the configured Cisco IoT FND IP address or host name is reachable from the device network.
- Verify that the configured Cisco IoT FND port is reachable.
- Check firewall rules, proxy rules, NAT rules, and ACLs that may block communication.
- If a load balancer is in the path, confirm that it allows long-lived WebSocket sessions.

Step 3 Check device-side IGMA service status and system resources:

Example:

- Confirm that the IGMA process is running.
- Confirm that all required dependent services are operational.
- If the IGMA service is unresponsive, restart it according to the device operating procedure.
- Check available memory, CPU, and disk space.
- Verify that temporary directories and log directories are accessible and not full.
- Investigate resource exhaustion events reported by the operating system.

Step 4 Validate the device configuration, IGMA URLs, and certificate requirements:

Example:

- Confirm that the device configuration matches the intended deployment.
- Verify controller, Cisco IoT FND, and telemetry settings.
- Check for configuration changes that occurred before the issue started.
- Verify that the IGMA registration, metrics, and tunnel URLs match the Cisco IoT FND provisioning settings and deployment mode.
- Confirm that the device trusts the certificate chain presented by Cisco IoT FND.
- Confirm that Cisco IoT FND trusts the device certificate when client certificate authentication is used.
- Confirm that the device clock and Cisco IoT FND server clock are synchronized.

```
https://<fnd-hostname>:9121/igma/register
https://<fnd-hostname>:9121/igma/metric
http://<fnd-hostname>:9124/igma/metric
```

Use the URL generated by the deployment template or provisioning settings for the actual device. Do not assume that `/igma/metric` and `/igma/metrics` are interchangeable.

Step 5 Enable Cisco IoT FND log levels for L+G border router debugging:

a) Choose `ADMIN > Logging`. In releases where logging is grouped under system management, choose `ADMIN > System Management > Logging`.
b) Click the `Log Level Settings` tab.
c) Set `Lwm2m` to the required debug log level.
d) Set `Websocket` to the required debug log level.
e) Save the log level settings.
f) If available, enable EID-specific debug logging for the L+G border router.

Step 6 Reproduce the problem and collect timestamped logs:

Example:

- Record the time when the issue is observed or reproduced.
- Download `server.log` from all Cisco IoT FND cluster nodes for the same timestamp range.
- Collect IGMA logs and system logs from the L+G border router.
- In a cluster, remember that the WebSocket session or metric request may be handled by a different node than the node you are using in the browser.

Step 7 Examine and correlate logs:

Example:

- Review IGMA logs on the device for errors, warnings, reconnect events, or repeated failures.
- Review FND `server.log` for the L+G border router EID and IGMA/WebSocket messages.
- Correlate IGMA logs, Cisco IoT FND logs, system logs, and network events by timestamp.

- Identify patterns such as recurring disconnects, communication failures, resource-related errors, or repeated reconnect cycles.

```
grep -i "<L+G-router-EID>" /opt/cgms/server/cgms/logs/server.log
grep -Ei
"igma|websocket|lwm2m|upgrade|handshake|register|metric|telemetry|tunnel|close|timeout|reset|certificate|401|403|404|500"
 /opt/cgms/server/cgms/logs/server.log
```

Step 8 Interpret the result and take corrective action:

Example:

- No IGMA or WebSocket log entry for the EID: the router is not reaching Cisco IoT FND, or it is using the wrong URL, port, DNS name, or certificate.
- Registration succeeds but metrics are stale: check the metrics profile URL, periodic metrics URL, metrics port, telemetry settings, and Cisco IoT FND `server.log`.
- The WebSocket session opens and then closes: check proxy/firewall idle timeout, TLS inspection, load balancer behavior, and device-side timeout settings.
- Firmware upload fails over IGMA: verify the transfer path and, if the upload exceeds the default timeout, review the `igma-idle-timeout` setting in `cgms.properties`.

Temporary communication interruptions may result in automatic reconnection attempts. Persistent failures, repeated reconnect cycles, or prolonged loss of telemetry should be investigated using the connectivity, service status, system resource, configuration, and log review steps above.

Troubleshooting Steps to Upload ODM File

At times, during the periodic metrics refresh, the IoT FND UI fails to provide the device metrics updates due to the absence of the ODM file (`cg-nms.odm`). To resolve this issue, you can download the `cg-nms.odm` file from the FND server and upload the file to the `/managed/odm` folder of the device from the Device File Management page of the FND UI.

- Retrieve Inventory Error, on page 47
- Upload the ODM File from FND UI, on page 56

Note This workaround is applicable to all Cisco IOS and IOS-XE device types that FND supports.

Retrieve Inventory Error

Here are the instructions for you to update the `cg-nms.odm` file to prevent the retrieve inventory error:

Before you begin

When you upgrade from Cisco IoT FND Release 4.10 to Cisco IoT FND Release 4.11.x or 4.12.x with Cisco IR8100 or Cisco IR1800 routers, you'll view the following error message:

```
java.io
.IOException: Failed to retrieve inventory from device.Reason [Invalid cli command] sent
[show platform hardware battery short] | formatflash:/managed/odm/cgms.odm platform hardware
 gnss details |
format flash:/managed/odm/cgms.odm,
show platform resources |
formatflash:/managed/odm/cgms.odm
```

There is a significant change in the `cg-nms.odm` file starting from Cisco IoT FND Release 4.10.x that triggers the error message.

Procedure

Step 1 Log in to the Cisco IoT FND server using the SSH terminal.

Step 2 Navigate to the folder location `/opt/cgms/standalone/deployments`

Step 3 Copy the `cgms.ear` file to a different directory

Here's an example:

```
[root@fnd410-107 deployments]# mkdir /opt/cgms-ear
[root@fnd410-107 opt]# cd /opt/cgms/standalone/deployments
[root@fnd410-107 deployments]#
cp cgms.ear /opt/cgms-ear/
```

Step 4 Unzip the `cgms.ear` file.

```
 [root@fnd410-107 cgms-ear]# cd /opt/cgms-odms/
[root@fnd410-107 cgms-ear]# ls
cgms.ear
[root@fnd410-107 cgms-ear]#
unzip cgms.ear
Archive:
cgms.ear
creating: META-INF/
inflating: META-INF/MANIFEST.MF
inflating: META-INF/application.xml
inflating: META-INF/jboss-deployment-structure.xml
inflating: cgms-aggr.jar
inflating: cgms-base.jar
inflating: cgms-c800.jar
inflating: cgms-ap800.jar
inflating: cgms-cgdm.jar
inflating: cgms-cgnms.jar
inflating: cgms-service.jar
inflating: cgms-cgrserver.jar
inflating: cgms-ciscoasr.jar
inflating: cgms-ciscoc8000.jar
inflating: cgms-ciscocgr.jar
inflating: cgms-mesh.jar
inflating: cgms-ciscocgos.jar
inflating: cgms-ciscocm.jar
inflating: cgms-ciscosbr.jar
inflating: cgms-ciscoesr5900.jar
inflating: cgms-ciscoir500.jar
inflating: cgms-iox.jar
inflating: cgms-ciscoios.jar
inflating: cgms-ciscoisr.jar
inflating: cgms-ciscoir800.jar
inflating: cgms-ciscolora.jar
inflating: cgms-common.jar
inflating: cgms-csmp.jar
```

```
inflating: cgms-dashboard.jar
inflating: cgms-db.jar
inflating: cgms-dhcp.jar
inflating: cgms-event.jar
inflating: cgms-her.jar
inflating: cgms-jobengine.jar
inflating: cgms-initialized.jar
inflating: cgms-labels.jar
inflating: cgms-logger.jar
inflating: cgms-markdown.jar
inflating: cgms-metrics.jar
inflating: cgms-odms.jar
inflating: cgms-nbapi.jar
inflating: cgms-netconf.jar
inflating: cgms-outage.jar
inflating: cgms-reprovision.jar
inflating: cgms-rules.jar
inflating: cgms-scheduler.jar
inflating: cgms-seam.jar
inflating: cgms-security.jar
inflating: cgms-snmp.jar
inflating: cgms-templates.jar
inflating: cgms-webapp.war
inflating: cgms-wsma.jar
inflating: cgms-osssh.jar
inflating: cgms-itroncam.jar
inflating: cgms-itronact.jar
inflating: cgms-itronbact.jar
inflating: cgms-lwm2m.jar
inflating: cgms-coap18.jar
inflating: cgms-cellnode.jar
inflating: cgms-iotgateway.jar
inflating: cgms-consul.jar
inflating: cgms-profile.jar
inflating: cgms-oracle.jar
inflating: cgms-tsdbadapter.jar
inflating: cgms-httpcoapproxy.jar
inflating: cgms-pnp.jar
inflating: cgms-ioxclient.jar
inflating: cgms-messageproducer.jar
inflating: cgms-ciscoir1100.jar
inflating: cgms-ciscoir1800.jar
inflating: cgms-ciscoir8100.jar
inflating: cgms-bsdiff.jar
inflating: cgms-websocket.jar
inflating: cgms-app.jar
inflating: cgms-ciscogenericdevice.jar
inflating: cgms-ciscogenericendpoint.jar
inflating: cgms-slservice.jar
inflating: cgms-lgnn.jar
inflating: cgms-lgelectric.jar
inflating: cgms-lgradio.jar
inflating: cgms-lglfn.jar
inflating: cgms-ciscoieswitch.jar
creating: lib/
inflating: lib/HikariCP-java7-2.4.13.jar
inflating: lib/antlr-runtime-3.2.jar
inflating: lib/c3p0-0.9.5.4.jar
inflating: lib/cglib-nodep-2.2.jar
inflating: lib/cgms-mibs.jar
inflating: lib/checker-qual-3.8.0.jar
inflating: lib/commons-beanutils-1.9.4.jar
inflating: lib/commons-codec-1.15.jar
inflating: lib/commons-collections-3.2.2.jar
```

```
inflating: lib/commons-configuration-1.10.jar
inflating: lib/commons-digester-1.8.1.jar
inflating: lib/commons-exec-1.1.jar
inflating: lib/commons-fileupload-1.5.jar
inflating: lib/commons-io-2.11.0.jar
inflating: lib/commons-lang-2.6.jar
inflating: lib/commons-lang3-3.11.jar
inflating: lib/commons-logging-1.2.jar
inflating: lib/commons-net-3.0.1.jar
inflating: lib/commons-validator-1.6.jar
inflating: lib/deltaspike-bean-validation-module-api-1.9.5.jar
inflating: lib/deltaspike-bean-validation-module-impl-1.9.5.jar
inflating: lib/deltaspike-cdictrl-api-1.9.5.jar
inflating: lib/deltaspike-cdictrl-weld-1.9.5.jar
inflating: lib/deltaspike-core-api-1.9.5.jar
inflating: lib/deltaspike-core-impl-1.9.5.jar
inflating: lib/deltaspike-data-module-api-1.9.5.jar
inflating: lib/deltaspike-data-module-impl-1.9.5.jar
inflating: lib/deltaspike-jpa-module-api-1.9.5.jar
inflating: lib/deltaspike-jpa-module-impl-1.9.5.jar
inflating: lib/deltaspike-jsf-module-api-1.9.5.jar
inflating: lib/deltaspike-jsf-module-impl-1.9.5.jar
inflating: lib/deltaspike-partial-bean-module-api-1.9.5.jar
inflating: lib/deltaspike-partial-bean-module-impl-1.9.5.jar
inflating: lib/deltaspike-proxy-module-api-1.9.5.jar
inflating: lib/deltaspike-proxy-module-impl-asm-1.9.5.jar
inflating: lib/deltaspike-security-module-api-1.9.5.jar
inflating: lib/deltaspike-security-module-impl-1.9.5.jar
inflating: lib/deltaspike-servlet-module-api-1.9.5.jar
inflating: lib/deltaspike-servlet-module-impl-1.9.5.jar
inflating: lib/encoder-1.2.2.jar
inflating: lib/error_prone_annotations-2.5.1.jar
inflating: lib/esper-4.10.0.jar
inflating: lib/failureaccess-1.0.1.jar
inflating: lib/freemarker-2.3.16.jar
inflating: lib/ftplet-api-1.1.1.jar
inflating: lib/ftpserver-core-1.1.1.jar
inflating: lib/guava-30.1.1-jre.jar
inflating: lib/j2objc-annotations-1.3.jar
inflating: lib/jackson-core-asl-1.9.9.redhat-4.jar
inflating: lib/jackson-mapper-asl-1.9.9.redhat-4.jar
inflating: lib/java-saml-2.6.0.jar
inflating: lib/java-saml-core-2.6.0.jar
inflating: lib/jboss-3.2.3.jar
inflating: lib/jboss-cache-1.4.1.GA.jar
inflating: lib/jboss-el-1.0_02.CR4.jar
inflating: lib/joda-time-2.10.6.jar
inflating: lib/jsch-0.1.59.jar
inflating: lib/json-20160810.jar
inflating: lib/jsr305-3.0.2.jar
inflating: lib/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
inflating: lib/mchange-commons-java-0.2.15.jar
inflating: lib/mibble-mibs-2.9.3.jar
inflating: lib/mibble-parser-2.9.3.jar
inflating: lib/mina-core-2.2.1.jar
inflating: lib/netty-3.2.1.Final.jar
inflating: lib/opencsv-2.1.jar
inflating: lib/openwebbeans-spi-2.0.23.jar
inflating: lib/org.apache.servicemix.bundles.snmp4j-2.2.5_1.jar
inflating: lib/org.everit.json.schema-1.5.1.jar
inflating: lib/picketlink-api-2.7.1.Final.jar
inflating: lib/picketlink-common-2.7.1.Final.jar
inflating: lib/picketlink-idm-api-2.7.1.Final.jar
inflating: lib/picketlink-idm-impl-2.7.1.Final.jar
```

```
inflating: lib/picketlink-impl-2.7.1.Final.jar
inflating: lib/prettytime-3.2.4.Final.jar
inflating: lib/protobuf-java-3.22.2.jar
inflating: lib/protobuf-java-format-1.2.jar
inflating: lib/quartz-2.3.2.jar
inflating: lib/snakeyaml-2.0.jar
inflating: lib/snmp4j-2.2.5.jar
inflating: lib/spring-core-5.2.23.RELEASE.jar
inflating: lib/spring-jcl-5.2.23.RELEASE.jar
inflating: lib/stax2-api-4.2.jar
inflating: lib/stringtemplate-3.2.jar
inflating: lib/syslog4j-0.9.30.jar
inflating: lib/woodstox-core-5.2.1.jar
inflating: lib/xmlsec-2.2.0.jar
inflating: lib/xmp_log4j_manager-8.0.27.jar
```

Step 5 Unzip the **cgms-odms.jar** file. Here's an example:

```
[root@fnd410-107 cgms-ear]# mkdir /opt/cgms-odms
[root@fnd410-107 cgms-ear]#
[root@fnd410-107 cgms-ear]# cp cgms-odms.jar /opt/cgms-odms/
[root@fnd410-107 cgms-ear]#
[root@fnd410-107 cgms-ear]# cd /opt/cgms-odms/
[root@fnd410-107 cgms-odms]#
[root@fnd410-107 cgms-odms]#
unzip cgms-odms.jar
Archive:
cgms-odms.jar
creating: META-INF/
inflating: META-INF/MANIFEST.MF
creating: com/
creating: com/cisco/
creating: com/cisco/cgms/
creating: com/cisco/cgms/ios/
creating: com/cisco/cgms/ios/odm/
creating: com/cisco/cgms/protocols/
creating: com/cisco/cgms/protocols/igma/
creating: com/cisco/cgms/protocols/igma/tlv/
inflating: com/cisco/cgms/ios/odm/DirectoriesDef$Directory$Files$Entry.class
inflating: com/cisco/cgms/ios/odm/DirectoriesDef$Directory$Files.class
inflating: com/cisco/cgms/ios/odm/DirectoriesDef$Directory.class
inflating: com/cisco/cgms/ios/odm/DirectoriesDef.class
inflating: com/cisco/cgms/ios/odm/ObjectFactory.class
inflating: com/cisco/cgms/ios/odm/SOAPEnvelope$SOAPBody$Response$ResultEntry$Success.class
inflating: com/cisco/cgms/ios/odm/SOAPEnvelope$SOAPBody$Response$ResultEntry.class
inflating: com/cisco/cgms/ios/odm/SOAPEnvelope$SOAPBody$Response.class
inflating: com/cisco/cgms/ios/odm/SOAPEnvelope$SOAPBody.class
inflating: com/cisco/cgms/ios/odm/SOAPEnvelope.class
inflating: com/cisco/cgms/ios/odm/ShowBootDef.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$CdmaAccountInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$CdmaDataProfileInfo$CdmaDataProfile.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$CdmaDataProfileInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$DataConnectionInfo$DataConnectionProfile.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$DataConnectionInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$GpsInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$HardwareInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$ModemInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$NetworkInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$ProfileInfo$ActiveProfile.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$ProfileInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$RadioInformation$CdmaRadioInfo.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef$RadioInformation.class
inflating: com/cisco/cgms/ios/odm/ShowCellularAllDef.class
inflating: com/cisco/cgms/ios/odm/ShowControllersCellularDef.class
```

```
inflating: com/cisco/cgms/ios/odm/ShowDot11AssociationsDef$Interface$SsidAssoc$Clients$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowDot11AssociationsDef$Interface$SsidAssoc$Clients.class
inflating: com/cisco/cgms/ios/odm/ShowDot11AssociationsDef$Interface$SsidAssoc.class
inflating: com/cisco/cgms/ios/odm/ShowDot11AssociationsDef$Interface.class
inflating: com/cisco/cgms/ios/odm/ShowDot11AssociationsDef.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef$AutodetectEntryCriteria.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef$AutodetectExitCriteria.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef$FrequencySupport.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef$PeriodicScanningStatus.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef$ScanningParameters.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioControllersDef.class
inflating: com/cisco/cgms/ios/odm/ShowDot16RadioInterfacesAssociationDef.class
inflating: com/cisco/cgms/ios/odm/ShowEnvironmentTemperatureDef$TemperatureTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowEnvironmentTemperatureDef$TemperatureTable.class
inflating: com/cisco/cgms/ios/odm/ShowEnvironmentTemperatureDef.class
inflating: com/cisco/cgms/ios/odm/ShowEventManagerPolicyAvailableDef$AvailablePolicies$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowEventManagerPolicyAvailableDef$AvailablePolicies.class
inflating: com/cisco/cgms/ios/odm/ShowEventManagerPolicyAvailableDef.class
inflating: com/cisco/cgms/ios/odm/ShowEventManagerPolicyRegisteredDef$RegisteredPolicyEntry.class
inflating: com/cisco/cgms/ios/odm/ShowEventManagerPolicyRegisteredDef.class
inflating: com/cisco/cgms/ios/odm/ShowHostsDef.class
inflating: com/cisco/cgms/ios/odm/ShowInterfaceDot16RadioLedsDef.class
inflating: com/cisco/cgms/ios/odm/ShowInterfacesDef$Interface.class
inflating: com/cisco/cgms/ios/odm/ShowInterfacesDef.class
inflating: com/cisco/cgms/ios/odm/ShowInventoryDef$InventoryEntry.class
inflating: com/cisco/cgms/ios/odm/ShowInventoryDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAllAppsDetailsDef$AppList.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAllAppsDetailsDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAppStatusDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAppsListDef$PerGosInfo$PerGosAppList$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAppsListDef$PerGosInfo$PerGosAppList.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAppsListDef$PerGosInfo.class
inflating: com/cisco/cgms/ios/odm/ShowIoxAppsListDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxHostDetailDef$IoxHost.class
inflating: com/cisco/cgms/ios/odm/ShowIoxHostDetailDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxPerAppDetailsDef.class
inflating: com/cisco/cgms/ios/odm/ShowIoxServiceDef.class
inflating: com/cisco/cgms/ios/odm/ShowIpInterfaceBriefDef$IPInterfaces$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowIpInterfaceBriefDef$IPInterfaces.class
inflating: com/cisco/cgms/ios/odm/ShowIpInterfaceBriefDef.class
inflating: com/cisco/cgms/ios/odm/ShowIpv6DhcpDef.class
inflating: com/cisco/cgms/ios/odm/ShowIpv6InterfaceDef$Interface$GlobalUnicastAddress.class
inflating: com/cisco/cgms/ios/odm/ShowIpv6InterfaceDef$Interface.class
inflating: com/cisco/cgms/ios/odm/ShowIpv6InterfaceDef.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$AntennaSignalQuality.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$GnssLocationData.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$MobileConnectionInformation.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$MobileNetworkStatus.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$ModemInformation.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$PimInformation.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$SignalQuality.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$SystemStatus.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef$WatchdogStatistics.class
inflating: com/cisco/cgms/ios/odm/ShowLte450AllDef.class
inflating: com/cisco/cgms/ios/odm/ShowMemoryStatisticsDef$MemoryStatistics$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowMemoryStatisticsDef$MemoryStatistics.class
inflating: com/cisco/cgms/ios/odm/ShowMemoryStatisticsDef.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysDef$MeshInterfaces$MeshSecurityKey.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysDef$MeshInterfaces.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysDef.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysLfnDef$MeshInterfaces$MeshSecurityKey.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysLfnDef$MeshInterfaces.class
inflating: com/cisco/cgms/ios/odm/ShowMeshSecurityKeysLfnDef.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModSerialNumMap$SerialNumMap$Entry.class
```

```
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModSerialNumMap$SerialNumMap.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModSerialNumMap.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModuleInfo$ModuleType$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModuleInfo$ModuleType.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef$ModuleInfo.class
inflating: com/cisco/cgms/ios/odm/ShowModuleDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryDetailsDef$BatteryUnit$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryDetailsDef$BatteryUnit.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryDetailsDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryShortDef$BatteryUnitInfo$BatteryUnit$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryShortDef$BatteryUnitInfo$BatteryUnit.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryShortDef$BatteryUnitInfo.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformBatteryShortDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformDoorDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformGpsDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformGuestOsDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformHardwareGnssDetailsDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformHypervisorDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedDef$LedSummary$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedDef$LedSummary.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$CellularLedSummary$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$CellularLedSummary.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$LedSummary$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$LedSummary.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$WimaxLedSummary$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$WimaxLedSummary.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$WpanLedSummary$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef$WpanLedSummary.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformLedSummaryDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformResourcesDef.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformSoftwareLicenseDef$SoftwareLicense.class
inflating: com/cisco/cgms/ios/odm/ShowPlatformSoftwareLicenseDef.class
inflating: com/cisco/cgms/ios/odm/ShowPrivilegeDef.class
inflating: com/cisco/cgms/ios/odm/ShowProcessesCpuDef.class
inflating: com/cisco/cgms/ios/odm/ShowProcessesCpuSortedDef.class
inflating: com/cisco/cgms/ios/odm/ShowProcessesMemorySortedDef.class
inflating: com/cisco/cgms/ios/odm/ShowRadioDetailsDef$RadioInformation.class
inflating: com/cisco/cgms/ios/odm/ShowRadioDetailsDef.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpSessionsDef$SessionsTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpSessionsDef$SessionsTable.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpSessionsDef.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticDef$StatisticsTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticDef$StatisticsTable.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticDef.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticsDef$StatisticsTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticsDef$StatisticsTable.class
inflating: com/cisco/cgms/ios/odm/ShowRawTcpStatisticsDef.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef$ScadaProtocol$Channel$TcpConnection.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef$ScadaProtocol$Channel.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef$ScadaProtocol$Sector.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef$ScadaProtocol$Session.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef$ScadaProtocol.class
inflating: com/cisco/cgms/ios/odm/ShowRunningConfigDef.class
inflating: com/cisco/cgms/ios/odm/ShowScadaStatisticsDef$Channel.class
inflating: com/cisco/cgms/ios/odm/ShowScadaStatisticsDef.class
inflating: com/cisco/cgms/ios/odm/ShowScadaTcpDef$Channel$Connection.class
inflating: com/cisco/cgms/ios/odm/ShowScadaTcpDef$Channel$TotalConnections.class
inflating: com/cisco/cgms/ios/odm/ShowScadaTcpDef$Channel.class
inflating: com/cisco/cgms/ios/odm/ShowScadaTcpDef.class
inflating: com/cisco/cgms/ios/odm/ShowSdCardStatusDef.class
inflating: com/cisco/cgms/ios/odm/ShowSnmpMibIfmibIfindexDef$IfIndex.class
inflating: com/cisco/cgms/ios/odm/ShowSnmpMibIfmibIfindexDef.class
```

```
inflating: com/cisco/cgms/ios/odm/ShowVersionDef$LicenseUdiTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowVersionDef$LicenseUdiTable.class
inflating: com/cisco/cgms/ios/odm/ShowVersionDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaModemGpsStatusDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaModemInfoDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaModemStatisticsDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaModemStatusDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaPktForwarderInfoDef.class
inflating: com/cisco/cgms/ios/odm/ShowVirtualLpwaPktForwarderStatusDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanConfigDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanHADetailDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanHardwareVersionDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanPacketCountDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplBriefDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplItableDef$RPLRouteTable$Entry.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplItableDef$RPLRouteTable.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplItableDef.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplStreeDef$Node.class
inflating: com/cisco/cgms/ios/odm/ShowWpanRplStreeDef.class
inflating: com/cisco/cgms/ios/odm/SnmpEngineDef.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigCliReply$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigCliReply$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigCliReply.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigCliReplyOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigTerminalReply$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigTerminalReply$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigTerminalReply.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ConfigTerminalReplyOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$Dir$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$Dir$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$Dir.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$DirOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFile$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFile$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFile.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFileOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFileReply$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFileReply$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFileReply.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$GetFileReplyOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostCli$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostCli$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostCli.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostCliOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostConfig$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostConfig$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostConfig.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostConfigOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostFile$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostFile$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostFile.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$PostFileOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowAesKey$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowAesKey$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowAesKey.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowAesKeyOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdInfo$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdInfo$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdInfo.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdInfoOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdStatus$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdStatus$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdStatus.class
```

```
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowCommonPktFwdStatusOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowFpga$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowFpga$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowFpga.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowFpgaOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInterfaces$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInterfaces$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInterfaces.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInterfacesOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInventory$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInventory$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInventory.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowInventoryOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowIpsecStatusInfo$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowIpsecStatusInfo$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowIpsecStatusInfo.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowIpsecStatusInfoOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowLedStatus$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowLedStatus$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowLedStatus.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowLedStatusOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderInfo$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderInfo$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderInfo.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderInfoOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderStatus$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderStatus$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderStatus.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPktForwarderStatusOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPlatformStatus$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPlatformStatus$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPlatformStatus.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowPlatformStatusOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowRadio$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowRadio$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowRadio.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowRadioOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowReplyNoFormat$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowReplyNoFormat$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowReplyNoFormat.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowReplyNoFormatOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowVersion$1.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowVersion$Builder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowVersion.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway$ShowVersionOrBuilder.class
inflating: com/cisco/cgms/protocols/igma/tlv/IotGateway.class
creating: META-INF/odm/
inflating: META-INF/odm/cg-nms-ap800.odm
inflating: META-INF/odm/cg-nms-ap800r.odm
inflating: META-INF/odm/cg-nms-c800.odm
inflating: META-INF/odm/cg-nms-esr5900.odm
inflating: META-INF/odm/cg-nms-ie4000.odm
inflating: META-INF/odm/cg-nms-ie4010.odm
inflating: META-INF/odm/cg-nms-ir1100.odm
inflating: META-INF/odm/cg-nms-ir1800.odm
inflating: META-INF/odm/cg-nms-ir800.odm
inflating: META-INF/odm/cg-nms-ir8100.odm
inflating: META-INF/odm/cg-nms-sbr.odm
inflating: META-INF/odm/cg-nms.odm
```

Step 6 The latest `ODM` files are available at `/opt/cgms-odms/META-INF/odm/`. Here's an example:

```
[root@fnd410-107 cgms-odms]# ls -l /opt/cgms-odms/META-INF/odm/
total 504
```

```
-rw-r--r-- 1 root root
8658 May 8 2024 cg-nms-ap800.odm
-rw-r--r-- 1 root root
8916 May 8 2024 cg-nms-ap800r.odm
-rw-r--r-- 1 root root 28192 May 8 2024 cg-nms-c800.odm
-rw-r--r-- 1 root root 26950 May 8 2024 cg-nms-esr5900.odm
-rw-r--r-- 1 root root 16884 May 8 2024 cg-nms-ie4000.odm
-rw-r--r-- 1 root root 16884 May 8 2024 cg-nms-ie4010.odm
-rw-r--r-- 1 root root 68545 May 8 2024 cg-nms-ir1100.odm
-rw-r--r-- 1 root root 69096 May 8 2024 cg-nms-ir1800.odm
-rw-r--r-- 1 root root 73980 May 8 2024 cg-nms-ir800.odm
-rw-r--r-- 1 root root 78343 May 8 2024 cg-nms-ir8100.odm
-rw-r--r-- 1 root root 19867 May 8 2024 cg-nms-sbr.odm
-rw-r--r-- 1 root root 70156 May 8 2024 cg-nms.odm
```

Step 7 Download the following on your local and rename them. Here's an example:

```
cg-nms-ir1800.odm or cg-nms-ir8100 and RENAME to cg-nms.odm
```

Step 8 Login to Cisco IoT FND, navigate to **CONFIG** > **Device Management**.

Step 9 Select configuration groups of the routers reporting the error message and click **Upload**.

Step 10 Click the **Select File** from the **List Window** and click **Add File**.

Note
You see the file upload successful message.

Step 11 Choose the **cg-nms.odm** file downloaded for upgrading to Cisco IoT FND Release 4.11.x or 4.12.x. Click **Add File** and then upload the file.

Step 12 In the **Select File From List** pane, click **Upload File**.

Step 13 In the **Upload File to Routers** page, select and override the list of routers on which the new Cisco IoT FND 4.11.x or 4.12.x cg-nms.odm files need to be upgraded. Click **Upload**.

Step 14 The **cg-nms.odm** file is uploaded to the router directory `bootflash:/managed/odm/ dirctory`.

Step 15 When you see the **Upload Complete** message, the refresh metrics are updates for the respective routers.

What to do next

Upload the ODM File from FND UI, on page 56

Upload the ODM File from FND UI

To upload the ODM file from FND UI:

Note Ensure that the ODM file renamed as `cg-nms.odm` is available in your PC.

Before you begin

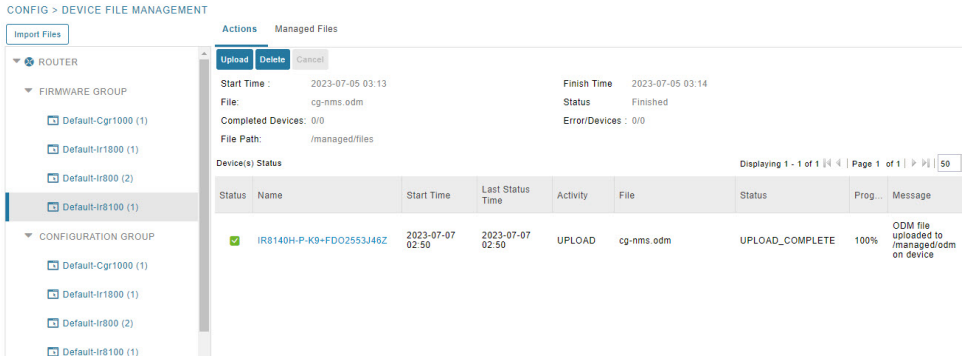
Retrieve Inventory Error, on page 47.

Procedure

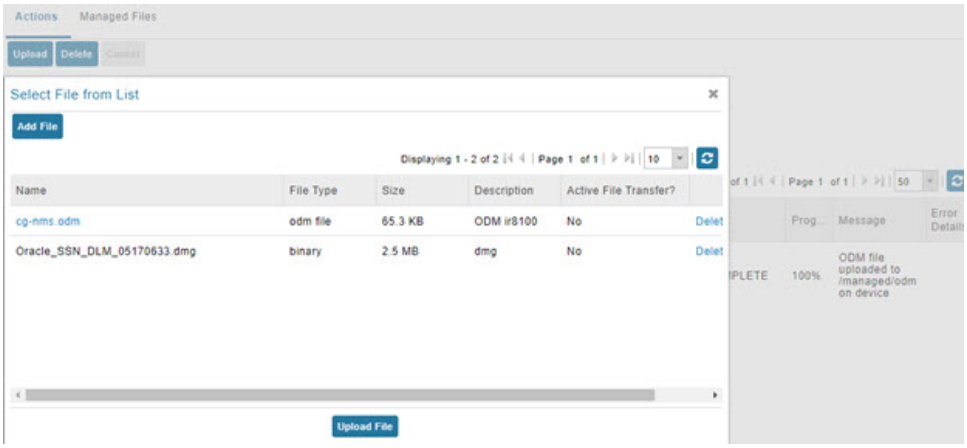
Step 1 Log in to IoT FND UI using a browser.

Step 2 Navigate to **CONFIG** > **Device File Management** page.

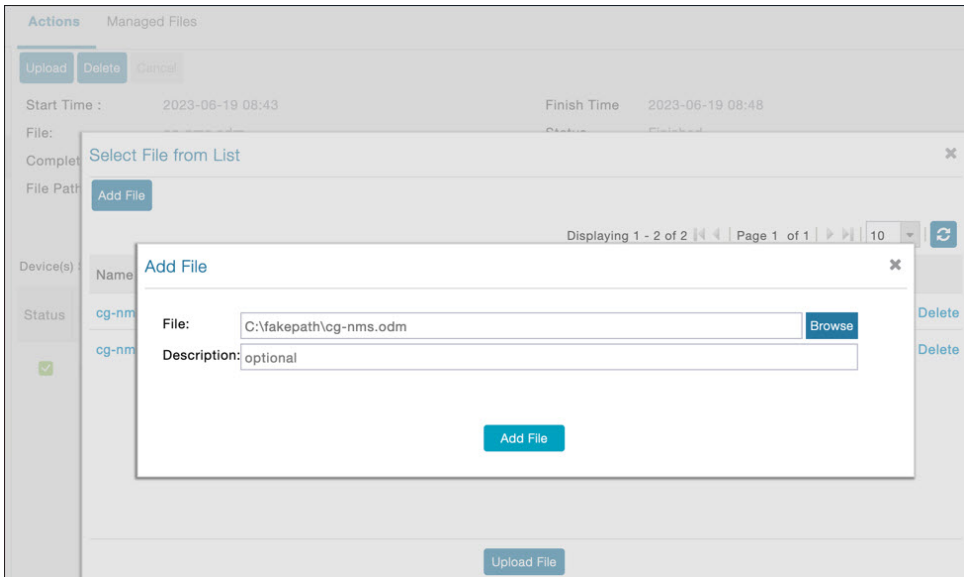
Step 3 In the Device File Management page, select the **Actions** tab and click **Upload**.



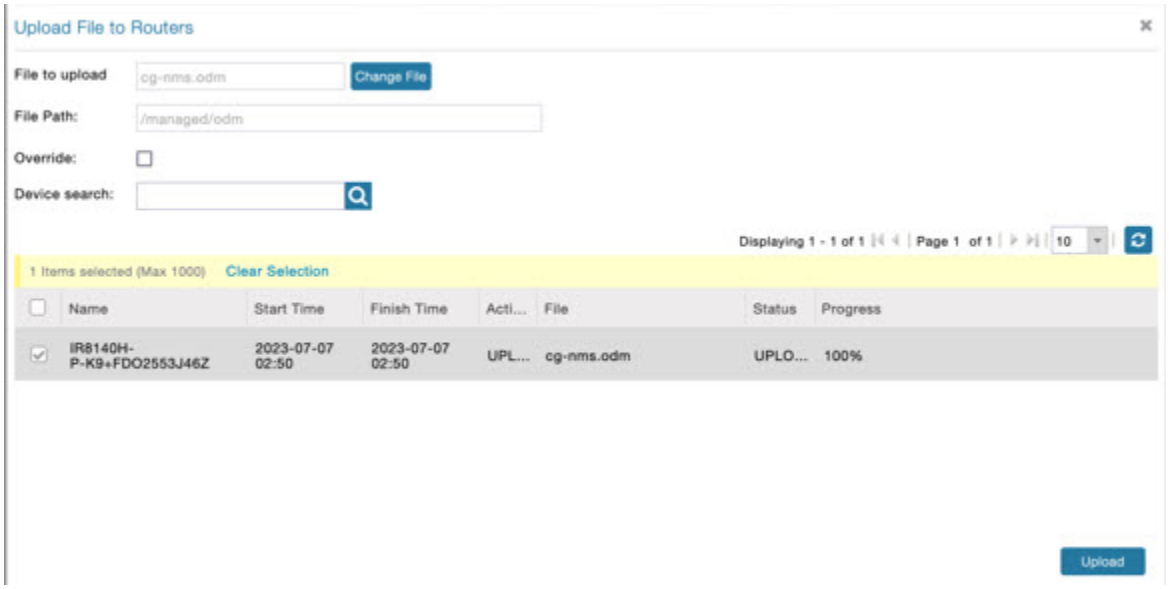
Step 4 In the **Select File from List** window, click **Add File**.



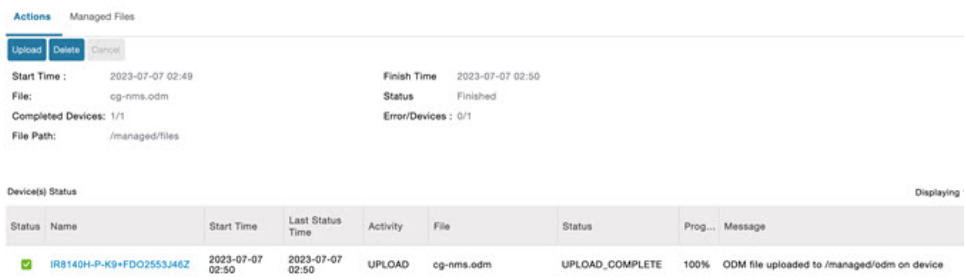
Step 5 Browse to the ODM file path (`cg-nms.odm`) and click **Add File** and then **Upload File**.



Step 6 Select the check box of the device(s) in the **Upload File to Routers** window and click **Upload**.



On successful completion of the upload, the Device Status table displays the upload completion message as shown below.



Note

Only the `cg-nms.odm` file gets uploaded to the `/managed/odm` folder, while the other files get uploaded to the `/managed/files` folder.

Troubleshoot TCL Scripts

Procedure

You can find the TCL scripts on a FAR at: `tmpsys:/lib/tcl/eem_scripts`.

Step 1 Debug using the `debug event manager tcl` commands.

Step 2 List planned scripts: `sh event manager statistics` policy.

Step 3 Manual execution: `event manager run tm_ztd_scep.tcl`.

Figure 7: Supported Troubleshooting TCL Scripts

```
CGR1240/K9+FTX2137G01G-Bootstrap#dir tmpsys:/lib/tcl/eem_scripts
Directory of tmpsys:/lib/tcl/eem_scripts/
   12  -r--        7458                  <no date>
ap_perf_test_base_cpu.tcl
   16  -r--       19119                  <no date>  cl_show_eem_tech.tcl
   76  -r--       20211                  <no date>  no_config_replace.tcl
   11  -r--        3327                  <no date>  no_perf_test_init.tcl
   13  -r--        4245                  <no date>  sl_intf_down.tcl
   10  -r--        6112                  <no date>  tm_cli_cmd.tcl
   14  -r--        8271                  <no date>  tm_crash_reporter.tcl
   15  -r--        5464                  <no date>  tm_fsys_usage.tcl
   18  -r--       15928                  <no date>  tm_rplpsn.tcl
   17  -r--       48910                  <no date>  tm_wanmon.tcl
   75  -r--       28940                  <no date>  tm_ztd_scep.tcl
```

Troubleshoot Certificate Enrollment

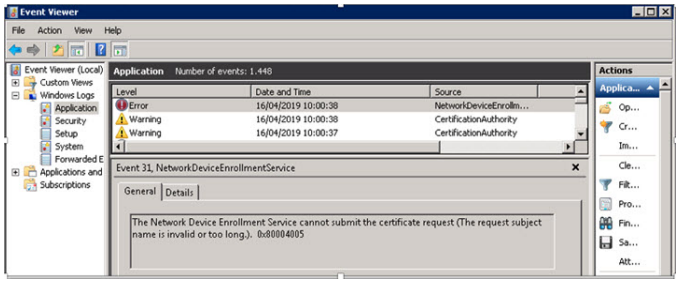
Procedure

Debug EEM and TCL on a FAR by entering the following command:

```
event manager environment ZTD_SCEP_Debug TRUE
```

- Manually perform trustpoint authentication and enrollment.
- Check Time and NTP
- Check NDES logs

Figure 8: Event Viewer



Certificate Enrollment — Test Manual

Procedure

Step 1 Save the current crypto config:

```
FGL204220HB# sh run | s crypto pki profile enrollment LDevID
FGL204220HB# sh run | s crypto pki trustpoint LDevID
```

Step 2 Remove crypto trustpoint in order to reset state and remove certificates:

```
no crypto pki trustpoint LDevID
```

Step 3 Re-add the saved configuration:

```
configure terminal
FGL204220HB# sh run | s crypto pki profile enrollment LDevID
FGL204220HB# sh run | s crypto pki trustpoint LDevID
```

Step 4 Authenticate with SCEP:

```
crypto pki authenticate LDevID
```

Step 5 Request Certificate:

```
crypto pki enroll LDevID
```

Certificate Enrollment — Example Output

```
CGR1120/K9+FOC21255M(config)#crypto pki authenticate LDevID
Certificate has the following attributes:
Fingerprint MD5: 438C8EB4 145564EF 4BACAFDB E5A338BB
Fingerprint SHA1: 0CF137AC F108235C F7125434 A0383728 852508D5
Trustpoint Fingerprint: 0CF137AC F108235C F7125434 A0383728 852508D5
Certificate validated - fingerprints matched.
Trustpoint CA certificate accepted.
CGR1120/K9+FOC21255M(config)#crypto pki enroll LDevID
```

```
%
% Start certificate enrollment...
% The subject name in the certificate will include: serialNumber=PID:CGR1120
SN:xxxxxxxxxx,CN=yyyyyyyyy
% The fully-qualified domain name will not be included in the certificate
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose LDevID' command will show the fingerprint.
CGR1120/K9+FOC21255M(config)#
Mar 21 08:13:38.475 UTC: CRYPTO_PKI: Certificate Request Fingerprint MD5: 34AE797C E6A9DB7E
 8EAA43E8
DC50CC45
Mar 21 08:13:38.475 UTC: CRYPTO_PKI: Certificate Request Fingerprint SHA1: F79DD9C7 015B8B7D
 E37130B7
543F2721 330E235C
Mar 21 08:13:43.201 UTC:%PKI-6-CERTRET: Certificate received from Certificate Authority
```

Troubleshoot WSMA

Before you begin

You must have cgms-tools installed before you can troubleshoot WSMA.

Procedure

Step 1 To execute:

```
/opt/cgms-tools/bin/wsma-request https://10.48.43.249:443/wsma/exec fndadmin cisco123
/opt/cgms/server/cgms/conf "show version | format flash:/managed/odm/cg-nms.odm"
```

Step 2 For an OVA install:

```
docker exec -it fnd-container /opt/cgms-tools/bin/wsma-request https://<FAR IP>:443/wsma/exec
<username> <password> /opt/cgms/server/cgms/conf "show version | format flash:/managed/odm/cg-nms.odm"
```

```
Example Output:
[root@iot-fnd ~]# docker exec -it fnd-container /opt/cgms-tools/bin/wsma-request
https://10.48.43.249/wsma/exec fndadmin cisco123 /opt/cgms/server/cgms/conf "show version | format
flash:/managed/odm/cg-nms.odm"
sending command: show version | format flash:/managed/odm/cg-nms.odm
<?xml version="1.0" encoding="UTF-8"?>
<ShowVersion xmlns="ODM://bootflash:/managed/odm/cg-nms.odm//show_version">
<Version>17.01.01</Version>
<VersionNonXe>17.1.1</VersionNonXe>
<HostName>IR1101</HostName>
<Uptime>1 week, 6 days, 3 hours, 3 minutes</Uptime>
<SystemImageFile>&amp;quot;bootflash:ir1101-universalk9.17.01.01.SPA.bin&amp;quot;</SystemImageFile>
<ReloadReason>Reload Command</ReloadReason>
<HardwareRevision>1.2 GHz</HardwareRevision>
<ProcessorBoardId>FCW223700AV</ProcessorBoardId>
<FastEthernetIntfCnt>4</FastEthernetIntfCnt>
<GigabitEthernetIntfCnt>2</GigabitEthernetIntfCnt>
<LicenseUdiTable>
</LicenseUdiTable></ShowVersion>
```

Troubleshoot Tunnel Provisioning

Procedure

Step 1 Substitute variables in the Router Tunnel Addition template (Figure 9) and check if the configuration is valid.

Step 2 Check server.log and optionally increase the log level.

Step 3 Check the head-end router (HER) Flex VPN.

Step 4 Debug on FAR using the following commands:

```
debug crypto sess
debug crypto ikev2
debug crypto ipsec
```

Figure 9: CONFIG > Tunnel Provisioning



Troubleshoot Netconf: FND—HER Communications

Procedure

Step 1 Start netconf session:

```
[root@iot-fnd ~]# ssh -l admin 10.48.43.228 -s netconf
Password:
```

Step 2 Device sends hello:

```
<?xml version="1.0" encoding="UTF-8"?><hello
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:capability:writeable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:startup:1.0</capability>
<capability>urn:ietf:params:netconf:capability:url:1.0</capability>
<capability>urn:cisco:params:netconf:capability:pi-data-model:1.0</capability>
<capability>urn:cisco:params:netconf:capability:notification:1.0</capability></capabilities><session-id>2036979584</session-id></hello>]]>]]>
```

Step 3 Send a hello yourself:

```
<?xml version=”1.0” encoding=”UTF-8”?>

<hello>

<capabilities>

<capability>urn:ietf:params:netconf:base:1.0</capability>

</capabilities>

</hello>]]>]]>
```

Step 4 Request running config (for example):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:rpc xmlns:ns2="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">

<ns2:get-config>

<source>

<ns2:running/>

</source>

</ns2:get-config>

</ns2:rpc>]]>]]>
```

Step 5 Device Response:

```
<?xml version=”1.0" encoding="UTF-8"?><rpc-reply message-id="1"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><data><cli-config-data-block>!

! Last configuration change at 16:10:25 UTC Thu Apr 4 2019 by admin

! NVRAM config last updated at 16:20:47 UTC Thu Apr 4 2019 by admin

!

version 16.3

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

platform console auto

!

hostname fnd4her
```

Troubleshoot Configuration Deployment

Procedure

Step 1 Substitute configuration and try manually line by line:

Step 2 Check device events: **Devices** > **Inventory** > **Select Device**.

Step 3 Debug CGNA/WSMA:

```
show cgna profile-state all
debug cgna logging ?
debug wsma agent
```

Troubleshoot HSM Connectivity

Procedure

To troubleshoot HSM connectivity:

```
[root@FNDPRDAPP01 bin]# /opt/cgms-tools/bin/signature-tool print

Certificate:

Data:

Version: 1

Serial Number: xxxxxxxxxx

Signature Algorithm: SHA256withECDSA

Issuer: CN=CGNMS, OU=CENBU, O=Cisco, L=San Jose, ST=CA, C=US

Validity

Not Before: Tue Feb 19 19:10:29 ICT 2019

Not After: Fri Feb 19 19:10:29 ICT 2049

Subject: CN=CGNMS, OU=CENBU, O=Cisco, L=San Jose, ST=CA, C=US

Fingerprints:

MD5: 4D:BB:C7:7A:02:2D:74:E5:99:62:AC:92:4A:8D:01:66

SHA1: 9B:C5:8F:BF:0B:7D:BF:4E:5F:E1:DB:8D:86:FC:8C:D0:C9:A1:F3:BA

Subject Public Key Info:

Public Key Algorithm: EC
```

```
…
Signature Algorithm: SHA256withECDSA
```

Issues Faced During HSM Client Upgrade

IoT FND accesses the HSM Server using the HSM Client.

In order for IoT FND to access the HSM Server, the HSM Client corresponding to the HSM Server version must be installed on the Linux server where the IoT FND application server is installed.

IoT FND is integrated with the HSM Client by using the HSM client API. The HSM client assigns a slot number to the HSM Server and also to the HA Group. On HSM Client 5.4 or earlier, the slot numbering started from one (1). However, in HSM Client 6.x and later, the slot numbering starts from zero (0).



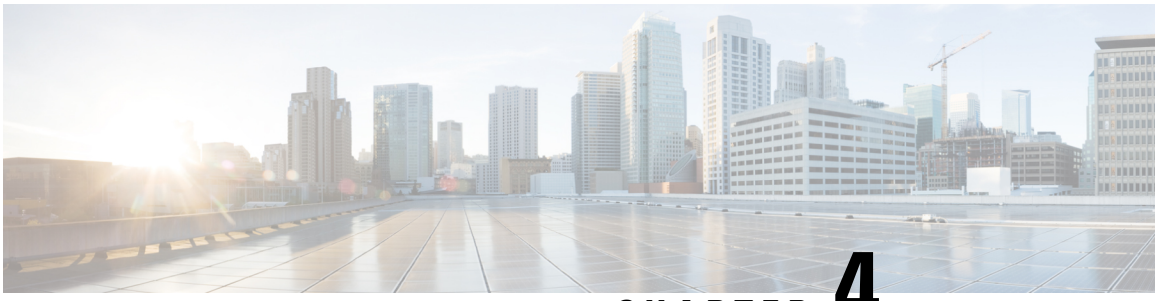
Note IoT FND gets the slot value dynamically from the HSM Client API. Sometimes during an upgrade from 5.4 to 7.3, the slot ID change is not dynamically populated. (CSCvz38606).



Note HSM Client 5.4 uses slot ID 1 (one). However, HSM Client 6.x and onward, slot ID 0 (zero) is used by the HSM client. The IoT FND application gets the value of the slot ID dynamically from the HSM client. The slot ID change will be communicated to the FND server by the HSM Client API upon restart of the IoT FND application. However, in some cases, the HSM client fails to send the correct value of the slot to the FND application server.

In such cases, where the FND Application Server has a value of 1 for the slot ID, but the HSM Client is using slot 0, and the HSM Client API is not giving the correct value dynamically, we can set the slot ID manually to one (1) in the HSM Client configuration file -/etc/Chrystoki.conf with the below:

```
Presentation = {OneBaseSlotID=1;}
```



CHAPTER 4

Troubleshoot Tunnel Provisioning Issues

Problem

Require guidance on troubleshooting issues related to tunnel provisioning failures by examining server logs and adjusting logging levels.

Solution

Use server.log Command

Here are the steps to troubleshoot tunnel provisioning issues:

1. Log in to the Cisco IoT FND server using the SSH command.
2. Navigate to the directory where your logs are stored using the cd command. For example: `/opt/fnd/logs` or `/opt/cgms/server/cgms/logs/`.
3. Run the `server.log` command to view the log files.

Use Cisco IoT FND GUI

1. Log in to Cisco IoT FND.
2. From the Cisco IoT FND menubar, choose **ADMIN** > **Logging** > **Log Level Settings**.
3. Ensure that **Tunnel Provisioning** option is checked.
4. Choose the **Log Level** as **Debug**.

Identify Issues In Logs

Look for specific error messages in the logs. For example, a common issue is the **Tunnel provisioning request queue is full**. This error indicates that the request queue has reached its capacity and new requests are being dropped.

Understand the Queue Limitation

Until Cisco IoT FND Release 4.12.1, the tunnel provisioning queue size is limited to 12 requests. If the queue is full, additional requests are not processed.

Resolution

- Consider upgrading to a Cisco IoT FND version later than Cisco IoT FND Release 4.12. The queue size is increased.
- Try restarting Cisco IoT FND which might temporarily resolve the issue by clearing the queue.

- Troubleshoot Archive Directory Creation During ZTD, on page 68
- Troubleshoot Zero Touch Deployment, on page 69
- Troubleshoot Tunnel Provisioning DHCP Configuration Issue, on page 71
- Troubleshoot Incorrect CGNA Profile URL, on page 72
- Troubleshoot HER FlexVPN, on page 72
- Troubleshoot Profile Environment Variable, on page 72
- Debug FAR Devices Using Commands, on page 73
- Verify TPS and DNS Connectivity, on page 73
- Verify Tunnel Configuration Information, on page 74

Troubleshoot Archive Directory Creation During ZTD

Problem

Cisco IoT FND fails to create the `bootflash:archive` directory during the bootstrap phase. This causes the ZTD (Zero Touch Deployment) onboarding to fail, resulting in configuration rollback error during tunnel provisioning.

Note If the directory already exists, the issue does not occur.

Solution

Here are the steps to address the directory creation issue:

1. Check the **cgms.properties** if the `reload-during-bootstrap` property is set to `true` or `false`.

Note By default, the `reload-during-bootstrap` property is set to `false`.

2. Take action based on the `reload-during-bootstrap` value:

   If `reload-during-bootstrap` is set to `true`.

   Create the archive directory using one of the following methods:

   - Manually create the directory `bootflash:archive` on the router before initiating the bootstrap process.
   - Automate the process using an EEM (Embedded Event Manager) script and add it to the bootstrap config. Here's an example:

```
event manager applet post_pnp
event timer watchdog time 30
action 10.0 cli command "enable"
```

```
action 15.0 cli command "file prompt quiet"
action 18.0 cli command "do mkdir bootflash:archive"
action 20.0 cli command "archive"
action 21.0 cli command "path bootflash:/archive/"
action 22.0 cli command "maximum 8"
action 24.0 cli command "no event manager applet post_pnp"
```

3. If `reload-during-bootstrap` is set to false, add `file prompt quiet` in the bootstrap template:

   Ensure the `file prompt quiet` command is included to suppress interactive confirmation prompts during directory creation. Here's an example:

```
!
<#-- Creating archive directory on bootflash -->
file prompt quiet
do mkdir bootflash:archive
!
archive
 path bootflash:/archive/
 maximum 8
!
```

4. Alternatively, you can use an alternative archive path:

   Specify a different directory path, such as `bootflash:`, to avoid the issue. Here's an example:

```
!
<#-- Configuring archive path to bootflash -->
archive
 path bootflash:
 maximum 8
!
```

5. Initiate the ZTD to confirm the issue is resolved after applying one of the solutions.

Troubleshoot Zero Touch Deployment

Problem

Unable to find the process for Zero Touch Deployment (ZTD) related to tunnel provisioning in a network setup.

Solution

Here are the steps involved in ZTD for tunnel provisioning:

1. Register your device on Cisco IoT FND.
2. Ensure that the FAR is booted and connected to WAN.
3. Obtain the initial configuration settings.
4. Request a Local Device Identifier (LDevID) certificate from a Public Key Infrastructure (PKI) or Registration Authority (RA).
5. Communicate with the Tunnel Provisioning Service (TPS) to initiate tunnel setup.
6. Establish a FlexVPN tunnel to the HER.

7. The FAR contacts the Cisco IoT FND for device registration.
8. The configuration is sent from Cisco IoT FND to FAR.
9. The FAR is fully operational and registered in the Cisco IoT FND.

Here's an example tunnel provisioning log:

```
Received tunnel provisioning request from [IR1101-K9+FCW22520078]
Adding tunnel provisioning request to queue for FAR ID=
Provisioning tunnels on element [IR1101-K9+FCW22520078]
Retrieved current configuration of element [IR1101-K9+FCW22520078] before tunnel provisioning
Retrieved status of file [flash:/before-registration-config] on [IR1101-K9+FCW22520078].
File does not exist
Retrieved status of file [flash:/before-tunnel-config] on [IR1101-K9+FCW22520078]. File
does not exist.
Copied running-config of [IR1101-K9+FCW22520078] to [flash:/before-tunnel-config]
Opened a NETCONF session with element [HTABT-TGOT-DC-RT1] at [163.88.181.2]
Sending [show interfaces | include Description: | Encapsulation | address is | line protocol
 | packets input, | packets output, | Tunnel protection | Tunnel protocol| Tunnel source]
to element [HTABT-TGOT-DC-RT1]
Received response to [show interfaces | include Description: | Encapsulation | address is
| line protocol | packets input, | packets output, | Tunnel protection | Tunnel protocol|
Tunnel source] from element [HTABT-TGOT-DC-RT1]
Sending [show ip nhrp | include ^[0-9A-F]| Tunnel| NBMA] to element [HTABT-TGOT-DC-RT1]
Received response to [show ip nhrp | include ^[0-9A-F]| Tunnel| NBMA] from element
[HTABT-TGOT-DC-RT1]
Sending [show ipv6 nhrp | include ^[0-9A-F]| Tunnel| NBMA] to element [HTABT-TGOT-DC-RT1]
Received response to [show ipv6 nhrp | include ^[0-9A-F]| Tunnel| NBMA] from element
[HTABT-TGOT-DC-RT1]
Sending [show ipv6 interface | include address | protocol | subnet] to element
[HTABT-TGOT-DC-RT1]
Received response to [show ipv6 interface | include address | protocol | subnet] from element
 [HTABT-TGOT-DC-RT1]
Closed NETCONF session with element [HTABT-TGOT-DC-RT1]
Obtained current configuration of element [HTABT-TGOT-DC-RT1] before tunnel provisioning
Configured tunnels on [IR1101-K9+FCW22520078]
Retrieved current configuration of element [IR1101-K9+FCW22520078] after tunnel provisioning.
Processed tunnel template for element [ASR1001+93UA2TVWZAR]. Time to process [5 ms].
Configured element [IR1101-K9+FCW223700AG] to register with IoT-FND at
[https://10.48.43.229:9121/cgna/ios/registration]
```



Note

- If the router tunnel addition template is empty, Cisco IoT FND proceeds with the standard registration process.
- Cisco IoT FND establishes a rollback point to ensure that the the Cisco Connected Grid Router (CGR) can revert to a trusted configuration. This is crucial before attempting tunnel provisioning or further device configuration phases. During the process, different configuration files are generated to serve as trust points. These files enable FND to revert the CGR's configuration if there are trust issues or if specific updates are needed. The configuration files are stored in the CGR's flash memory.
- Establish a rollback point to support the provisioning of new tunnel configurations.

Troubleshoot Tunnel Provisioning DHCP Configuration Issue

Problem

Seeing a **Tunnel Provisioning Failure** event in the Cisco IoT FND logs.

Solution

You need to monitor the address allocation process using the following steps:

1. Check the Cisco IoT FND server.log file to determine if Cisco IoT FND is sending a DHCP request during tunnel provisioning.
2. Check the DHCP server.log file to determine if the DHCP request from Cisco IoT FND reached the DHCP server.
3. If the requests are not reaching the server. From the Cisco IoT FND menubar, choose **ADMIN** > **System Management** > **Provisioning Settings** and ensure that the DHCP **Server Address** is right.
4. Check for network problems between Cisco IoT FND server and the DHCP server.
5. Analyze the DHCP server log files to identify errors in the configuration and fix them.

Example

The Cisco Connected Grid Routers time out during Tunnel Provisioning and then rollback to before-tunnel-config repeatedly:

Error seen in server.log:

```
238471507: imcltcgnmsp01: Dec 13 2017 03:24:03.709 +0000: %CGMS-3-UNSPECIFIED:
%[ch=CiscoIosTunnelProvServlet$CiscoIosTunnelProvProcess][eid=CGR1240/K9+FTX2130G06C][ip=10.189.110.137][sev=ERROR][tid=IOS
 CGR Tunnel-892]: Tunnel provisioning request for element [CGR1240/K9+FTX2130G06C] failed
[com.cisco.cgms.dhcp.proxyclient.v4.TimeoutWaitingForOfferException: Timed out waiting for
 a DHCPOFFER message].
```

In the above error, you can see that the Cisco IoT FND server has sent a DHCPDISCOVER request and is waiting for a DHCPOFFER message from the DHCP server and it times out waiting.

Here are the DHCP logs:

```
Bluecat DHCP: Customer's DHCP
Dec 13 15:24:44 cltuofdhcp dhcpd: DHCPRELEASE of 10.190.140.110 from 00:00:00:00:00:00 via
 10.200.230.34 (found)
Dec 13 15:24:44 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPOFFER on 10.190.140.110 to 00:00:00:00:00:00 via
10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPREQUEST for 10.190.140.110 (10.200.230.33) from
00:00:00:00:00:00 via 10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPACK on 10.190.140.110 to 00:00:00:00:00:00 via
10.200.230.34
Dec 13 15:24:46 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
Dec 13 15:24:50 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
Dec 13 15:24:58 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
```

From the above logs, you can see that there is an issue which prevents IPs in the 10.190.192.0/18 scope from being allocated. DHCP configuration needs to be fixed in order for tunnel provisioning to work in this case.

Troubleshoot Incorrect CGNA Profile URL

Problem

Seeing an incorrect CGNA profile configuration URL error.

Solution

Verify if you've entered the URL in the right format. The right format of the CGNA profile configuration URL is https://<FND IP>:9120/cgna/ios/tunnel

Troubleshoot HER FlexVPN

Problem

Unable to configure the FlexVPN client on a Cisco Connected Grid Router (CGR).

Solution

Here are the instructions to configure the FlexVPN on a Cisco Connected Grid Router:

1. Access the Cisco Connected Grid Router's configuration using the SSH command.
2. Run `show running-config`, `show crypto ikev2 sa`, and `show cypto ipsec sa` to review the FlexVPN configuration.
3. Use `show interfaces` to check the status of the dynamically created tunnel interface.
4. Verify that the tunnel interface associated with FlexVPN is up and running.
5. Run `show logging` and review the logs for any errors or warnings related to FlexVPN.
6. Look for messages indicating issues in tunnel establishment or connectivity.
7. Use `show template` to review the applied templates and confirm if they match the intended configuration for FlexVPN.

   Verify that the HER FlexVPN configuration on the Cisco IOS Connected Grid Router is set up and operational. This ensures that the CGR can dynamically create tunnel endpoints and maintain secure communication with the HER.

Troubleshoot Profile Environment Variable

Problem

Need to verify that the correct profile is configured for the `ZTD_SCEP_CGNA_Profile` environment variable.

Solution

You'll need to check the Cisco Global Network Architecture (CGNA) profile settings on your device:

1. Log in to the device using the SSH command.
2. Run `show cgna profile-all` and see all the configured CGNA profiles and check which one is active.
3. Look for `ZTD_SCEP_CGNA_Profile` environment variable ensuring it is set to the intended profile.
4. Confirm that the TCL script `tm_ztd_scep.tcl` has run without errors. This script is responsible for activating the correct CGNA profile.
5. Ensure that the LDevID certificate is correctly retrieved from the PKI, as this is necessary for the profile activation.
6. Check logs for any errors or warnings that might indicate problems with profile configuration or activation.

Debug FAR Devices Using Commands

Problem

Unable to debug FAR devices.

Solution

Here are the instructions to debug FAR devices using commands:

1. Login to Cisco IoT FND server as root using the SSH command.
2. Run the following commands to debug FAR devices:

```
show crypto session
debug crypto ikev2
debug crypto ipsec
```

You see the debugging logs for you to take action on.

Verify TPS and DNS Connectivity

Problem

How to ensure TPS and DNS connectivity?

Solution

Here are the steps to ensure TPS and DNS connectivity:

1. Ensure that the TPS service is running on Cisco IoT FND.
2. Review the TPS proxy logs located at: `/opt/cgms-tpsproxy/log/tpsproxy.log` to identify any issues.

3. If a Fully Qualified Domain Name (FQDN) is used, ensure the CGNA profile correctly resolves the TPS FQDN through the DNS.
4. Verify connectivity between the DNS and the FAR. Attempt to ping the TPS FQDN from the FAR to confirm connectivity.
5. Ensure the DNS records are correctly set up to resolve the TPS FQDN. This involves checking the DNS server for accurate entries.
6. Check the TPS FQDN configuration in the CGNA URL to ensure it is correctly set up and pointing to the intended server.

Verify Tunnel Configuration Information

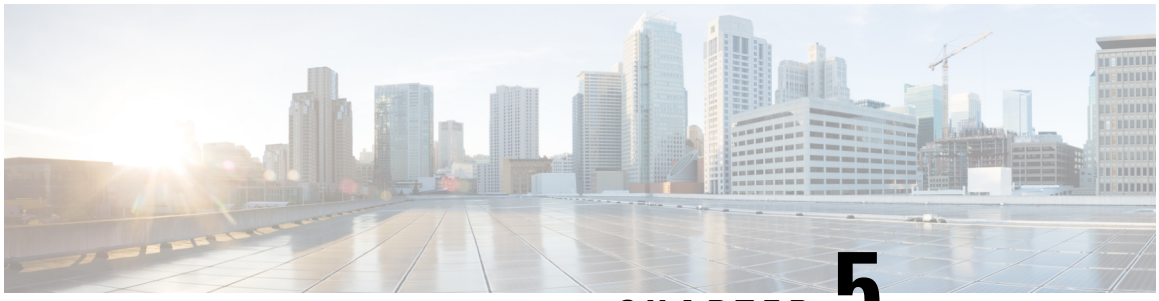
Problem

Unable to verify the tunnel configuration information.

Solution

Here are the steps to verify the tunnel configuration information:

1. From the Cisco IoT FND menubar, choose **CONFIG** > **Tunnel Provisioning** > **Router Tunnel Addition**.
2. You can see the entire router tunnel addition information in the tab.



CHAPTER 5

FAR Management Issues

This chapter explains some of the FAR management issues and workaround for them.

- Certificate Exception, on page 75
- FAR Keeps Reloading and Does Not Switch to the Up State, on page 75
- Incorrect FAR State in IoT FND, on page 76

Certificate Exception

If this exception appears in the server.log file stored on the IoT FND server when a FAR attempts to register with IoT FND, the cgms_keystore file does not contain the CA server certificates or the CA certificates that were imported into the cgms_keystore file are incorrect:

```
SSLException: Received fatal alert: unknown_ca
```

For information about how to import certificates into the cgms_keystore file, see “Generating and Installing Certificates in the Cisco IoT Installation Guide, 4.0.x and greater.

FAR Keeps Reloading and Does Not Switch to the Up State

When a FAR is continuously reloading every time it contacts IoT FND, it could be because the configuration pushed to the FAR by IoT FND is not being applied successfully.

Check the server.log file on the IoT FND server for clues on the cause of the configuration push failure. Sometimes, typos in the in the Field Area Router Tunnel Addition template cause this failure (IoT FND does not provide template validation).

Note When a FAR registers with IoT FND, IoT FND queries the FAR with show commands. IoT FND then configures the FAR based on the configuration commands in the Field Area Router Tunnel Addition template.

Other reasons for continuous reloads may be:

- A bad WAN link that drops packets and does not allow the registration to complete.
- Firewall issues.

Ensure that the firewall allows traffic in both directions and that traffic to and from the correct ports is allowed to pass.

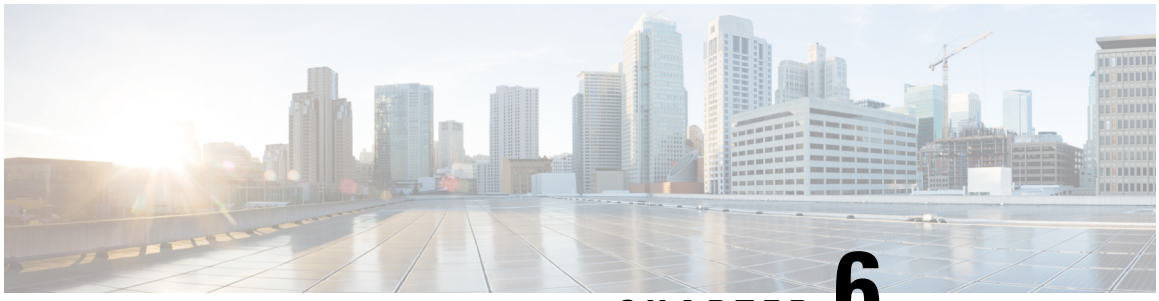
Incorrect FAR State in IoT FND

In IoT FND, a FAR might appear in a Down state even though you can ping and trace the route to it without a problem.

IoT FND manages the FAR via the IoT-DM service running on the FAR. So even though the FAR is pingable and reachable, it is important to verify that the jetty server and call home features are enabled on the FAR:

```
show run callhome
```

should have 'enable' in the config and `sh jvm status`



CHAPTER 6

Mesh Endpoint Management Issues

This chapter explains some of the mesh endpoint issues and possible resolutions.

- Mesh Endpoints Not Registering with IoT FND, on page 77
- Mesh Endpoint Registration Reason Codes, on page 77
- Reasons for Mesh Endpoint WPAN Changes, on page 78

Mesh Endpoints Not Registering with IoT FND

Procedure

Verify that the mesh endpoints have joined the FAR and are pingable from IoT FND over IPv6. If they are pingable, verify the following:

- The clock is in sync.
- The DHCP server used by the mesh endpoints is programmed with the correct IoT FND IP address.
- The mesh endpoints are running an image compatible with the current version of IoT FND.
- If HSM is used, HSM must be online and responding correctly.

Mesh Endpoint Registration Reason Codes

Registration Reason Code	Code	Event Type Name	Severity	Message	Description
REASON_UNKNOWN	0	unknownRegReason	INFO	Mesh node registered for unknown reason	N/A

Registration Reason Code	Code	Event Type Name	Severity	Message	Description
REASON_COLDSTART	1	coldBoot	INFO	Mesh node registered due to cold boot.	This message includes the new IP address of the mesh endpoint.
REASON_ADMIN	2	manualReRegistration	INFO	Mesh node registered due to manual registration.	The endpoint received an NMSRedirectRequest without a URL field.
REASON_IP_CHANGE	3	rejoinedWithNewIP	INFO	Mesh node registered with new IP address	This message includes the new IP address of the mesh endpoint.
REASON_NMS_CHANGE	4	nmsAddrChange	INFO	Mesh node registered due to NMS address change.	The IoT FND IP address changed OUTSIDE of an NMSRedirect (a new DHCPv6 option value was received)
REASON_NMS_REDIRECT	5	nmsNMSAddrChange	INFO	Mesh node registered due to manual NMS address change.	Endpoint received an error from IoT FND.
REASON_NMS_ERROR	6	nmsError	INFO	Mesh node registered due to NMS error.	Endpoint received an error from IoT FND.

Reasons for Mesh Endpoint WPAN Changes

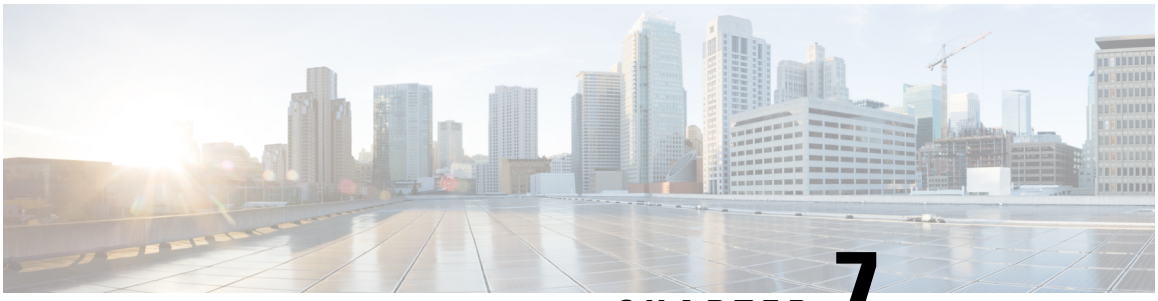
In addition to generating events when mesh endpoints register with IoT FND, IoT FND also generates events after receiving a WPAN change TLV WPANStatus.

```
Event logged: Event(id=0, eventTime=1335304407974,
eventSeverity=0, eventSource=cgmesh,
evenMessage=WPAN change due to migration to better
PAN: [lastChanged: 0, lastChangedReason: 4],
NetElement, id=10044,
EventType, name=null, lat=1000.0,
lng=1000.0, geoHash=null)
```

Table 6: Reasons for Mesh Endpoint WPAN Changes

Registration Reason Code	Code	Event Name	Severity Type	Description
IEEE154_PAN_LEAVE_UNKNOWN	-1	unknownWPANChange	MAJOR	WPAN change for unknown reason.
IEEE154_PAN_LEAVE_INIT	0	meshInit	N/A	No event is generated for this code.
IEEE154_PAN_LEAVE_SYNC_TIMEOUT	1	meshConnectivityLost	MAJOR	WPAN change due to mesh connectivity loss.
IEEE154_PAN_LEAVE_GTK_TIMEOUT	2	meshLinkKeyTimeout	MAJOR	WPAN change due to mesh link key timeout.
IEEE154_PAN_LEAVE_NO_DEF_ROUTE	3	defaultRouteLost	MAJOR	WPAN change for no default route.
IEEE154_PAN_LEAVE_OPTIMIZE	4	migratedToBetterPAN	MAJOR	WPAN change due to migration to better PAN.

For these events, the message includes the time elapsed since the mesh endpoint left the network to when it rejoined. IoT FND displays the amount of time the mesh endpoint was offline since the event was logged (for example, 4 hours 23 minutes ago).



CHAPTER 7

Operational Issues You May Encounter

This chapter explains some of the operational issues and possible resolutions.

- Tunnel Provisioning DHCP Configuration Issues, on page 81
- Recovering an Expired Database Password, on page 82
- Unlocking the IoT FND Database Password, on page 82
- IoT FND Service Will Not Start, on page 82
- Exception in the server.log File on the IoT FND Server, on page 83
- Resetting the root Password, on page 84
- Second IoT FND Server Not Forming a Cluster, on page 84
- IoT FND Service Restarts Automatically, on page 84
- Fallback URL When SSO Fails, on page 85
- Router Registration with IoT FND Fails over Cellular Network after Successful Tunnel Provisioning, on page 85
- DB Migration fails due to Incorrect Incremental Size , on page 86
- Missing Endpoint Marker Line to CGR, on page 88
- Troubleshoot remote authentication failures when NTLMv2-only policy is enforced , on page 88

Tunnel Provisioning DHCP Configuration Issues

If there is a problem allocating an address, IoT FND logs a Tunnel Provisioning Failure event. The log entry includes details of the error.

To monitor the address allocation process:

- Check the IoT FND server.log file to determine if IoT FND is sending a DHCP request during tunnel provisioning.
- Check your DHCP server log file to determine if the DHCP request from IoT FND reached the DHCP server.

If requests are not reaching the server:

- Ensure that the DHCP server address is correct on the **Provisioning Settings** page in IoT FND (**Admin** > **System Management** > **Provisioning Settings**).
- Check for network problems between IoT FND and the DHCP server.

If the DHCP server is receiving the request but not responding:

- View the DHCP server log file, and ensure that the DHCP server is configured to support requests from the link address included in the DHCP requests. The link address is defined in the tunnel provisioning template.
- Ensure that the DHCP server has not exhausted its address pool.

If the DHCP server is responding, but IoT FND is not processing the response:

- Ensure that the lease time is infinite. Otherwise, IoT FND will not process the response.
- View the DHCP server logs and IoT FND server logs for other errors.

Recovering an Expired Database Password

Procedure

To recover from an expired password, run these commands:

```
su - oracle
sqlplus sys/cgmsDbaAccount@cgms as sysdba
alter user cgms_dev identified by test;
alter user cgms_dev identified by password;
exit;
```

Unlocking the IoT FND Database Password

Procedure

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;
```

IoT FND Service Will Not Start

If the OS version is RHEL 8.x, then use **systemctl** command instead of the **service** command as given in the table.

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> cgms`
7.x	`service cgms <status/start/restart/stop>`

Note To check the OS version, run the following command:

```
cat /etc/os-release
```

If the IoT FND service does not start:

Procedure

Step 1 Validate connectivity to the database:

a) Log in as root on the IoT FND server.

b) Enter the following at the command prompt:

```
service cgms status
```

c) Verify the database server IP address and that IoT FND can connect to the database.

Note
If the IP address is incorrect or if IoT FND cannot access the database, run setupCgms.sh and enter the correct values.

d) Run the **service cgms status** command and verify connectivity.

e) Start IoT FND. is correct (see the System Requirements chapter).

Step 2 Verify that the JRE version installed on the server is correct. (See FND release notes).

Step 3 Verify that database migration was performed successfully.

Exception in the server.log File on the IoT FND Server

If there is an exception in the server.log file indicating that IoT FND could not open the cgms_keystore file, then the cgms_keystore password stored in the cgms.properties file on the IoT FND server is incorrect.

The password for the cgms_keystore file is encrypted and stored in the `/opt/cgms/server/cgms/conf/cgms.properties` file.

To encrypt or decrypt the password:

Procedure

Step 1 Use the script: `/opt/cgms/bin/ encryption_util.sh`

Step 2 Verify or update the password in the cgms.properties file, and if an update is required, restart IoT FND after modifying the password.

Resetting the root Password

Procedure

If you forget the password of the IoT FND root user account, reset the password by running the script:

```
/opt/cgms/bin/password_admin.sh
```

Second IoT FND Server Not Forming a Cluster

Typically, discovery of nodes in a IoT FND cluster is automatic. As long as the IoT FND servers are on the same subnet, they form a cluster.

If you install an IoT FND server and it does not join the cluster:

Procedure

Step 1 Verify that your servers are on the same subnet, can ping each other, and share the same cluster name.

Step 2 Check the status of all members by running the script:

```
/opt/cgms/bin/print_cluster_view.sh
```

Step 3 Modify the cluster name, as follows:

a) Change the value of the HA_PARTITION_NAME parameter on all IoT FND cluster nodes, and then restart them.
b) Change the value of the UDP_MULTICAST_ADDR parameter (unique multicast address) to match on all nodes in the cluster.
c) Change the value of the CLUSTER_BIND_ADDR parameter to the interface to which you want the NMS to bind.

Step 4 Verify that all the cluster nodes are configured to use NTP (see Configuring NTP Service).

Step 5 Check the `/etc/hosts` file and verify that the IP address is correctly mapped to the hostname of the local server.

IoT FND Service Restarts Automatically

When the IoT FND services are started, the watchdog script is invoked. The watchdog script checks the health of the IoT FND services. If the watchdog script detects an anomaly, it logs the conditions in the `/opt/cgms/server/cgms/log/cgms_watchdog.log` file.

The watchdog script tries three times to determine if the anomaly condition improved. If not, it restarts the IoT FND services automatically, unless the database has become unreachable. If the database is not reachable, the watchdog stops the IoT FND services. Check the log files, including server.log, to determine what is causing the restarts.

Procedure

Manually disable the watchdog process by running the following script on the IoT server as root.

```
/opt/cgms/bin/deinstall_cgms_watchdog.sh
```

Fallback URL When SSO Fails

Use the FND console URL as a fallback URL to configure the authentication settings when SSO login fails. The root users and the users with administrative privileges only can access the FND console URL.

Table 7: Console URL

IoT FND Releases	Console URL
IoT FND Release 4.10.0	https://<FND-IP>/consolelogin.seam
IoT FND Releases 4.9.x and 4.8.x	https://<FND-IP>/console/home.seam

Note The FND console URL is not used for the IDP authentication.

Router Registration with IoT FND Fails over Cellular Network after Successful Tunnel Provisioning

Router registration with IoT FND fails with “SSL peer shutdown incorrectly” error over the cellular network after successful tunnel provisioning.

Time	Event Name	Severity	Message
2022-07-26 14:53:28:707	Registration Failure	INFO	javax.xml.ws.soap.SOAPFaultException: Remote host closed connection during handshake; Caused by: com.ctc.wstx.exc.WstxIOException: Remote host closed connection during handshake; Caused by: javax.net.ssl.SSLHandshakeException: Remote host closed connection during handshake; Caused by: java.io.EOFException: SSL peer shut down incorrectly
2022-07-26 14:53:18:401	Registration Request	INFO	Registration request from device.

The provisioned tunnels did not have the MTU set correctly for the cellular network. The MTU settings are on both sides of the tunnel; the HeadEnd Router (HER) and the Field Area Router (FAR).

On HER, the sample configuration below shows the settings required for MTU and MSS:

```
interface Virtual-Template1 type tunnel
ip unnumbered Loopback0
ip mtu 1300
```

```
ip tcp adjust-mss 1260
ipv6 unnumbered Loopback0
ipv6 enable
ipv6 mtu 1280
ipv6 tcp adjust-mss 1220
nat64 enable
tunnel source GigabitEthernet0/0/1
tunnel path-mtu-discovery
tunnel protection ipsec profile FlexVPN_IPsec_Profile
!
ip tcp mss 1460
ip tcp path-mtu-discovery
!
```

On the FAR side, the sample configuration below shows the settings required for MTU and MSS:

```
interface Tunnel10
 description to HER
 no ip address
 ipv6 unnumbered Loopback0
 ipv6 mtu 1280
 ipv6 tcp adjust-mss 1240
 tunnel source Cellular0/3/0
 tunnel destination dynamic
 tunnel protection ipsec profile FlexVPN_IPsec_Profile
!
interface Cellular0/3/0
 ip address negotiated
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip tcp adjust-mss 1390
 load-interval 30
 dialer in-band
 dialer idle-timeout 0
 dialer watch-group 1
 dialer-group 1
 ipv6 enable
 pulse-time 1
!
ip tcp mss 1460
ip tcp path-mtu-discovery
```

The MTU settings above 1300 on cellular backhauls can cause the registration error message.

DB Migration fails due to Incorrect Incremental Size

During IoT FND upgrade to 4:10, DB migration fails due to incorrect incremental size. The database schema and IoT FND should match in order to avoid the failure during DB migrate, and hence the hibernate jars were upgraded in Cisco IoT FND 4.10.0. The following error message appears when the sequence name (for example, audit_trail_id_seq) in the db is not properly updated:

```
Current schema version: A.4.9.0.20220809.01
Migrating to version A.4.10.0.20230316.01
Migration completed. Successfully applied 1 migration.
07-18-2023 14:48:43 EDT: INFO: Migration completed.
07-18-2023 14:48:43 EDT: INFO: Performing post migration. This may take a while. Please
wait ...
2023-07-18 18:48:47,709:ERROR:main:CgmsDbMigrationDriver: Migration failed. Exception:
Reason :
javax.persistence.PersistenceException: [PersistenceUnit: common] Unable to build Hibernate
```

```
SessionFactory
at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.persistenceException
(EntityManagerFactoryBuilderImpl.java:1336)
at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.build
(EntityManagerFactoryBuilderImpl.java:1262)
at org.hibernate.jpa.HibernatePersistenceProvider.createEntityManagerFactory
(HibernatePersistenceProvider.java:56)
at javax.persistence.Persistence.createEntityManagerFactory(Persistence.java:55)
at com.cisco.cgms.tools.CommandLineInit.initDataSource(CommandLineInit.java:57)
at com.cisco.cgms.tools.CommandLineInit.<init>(CommandLineInit.java:26)
at com.cisco.cgms.tools.CgmsDbMigrationDriver.<init>(CgmsDbMigrationDriver.java:41)
at com.cisco.cgms.tools.CgmsDbMigrationDriver.main(CgmsDbMigrationDriver.java:93)
Caused by: org.hibernate.tool.schema.spi.SchemaManagementException: Schema-validation:
sequence [cgms_dev.audit_trail_id_seq] defined inconsistent increment-size; found [1000]
but exp ecting [1]
at org.hibernate.tool.schema.internal.AbstractSchemaValidator.validateSequence
(AbstractSchemaValidator.java:191)
at org.hibernate.tool.schema.internal.AbstractSchemaValidator.performValidation
(AbstractSchemaValidator.java:100)
at org.hibernate.tool.schema.internal.AbstractSchemaValidator.doValidation
(AbstractSchemaValidator.java:68)
at org.hibernate.tool.schema.spi.SchemaManagementToolCoordinator.performDatabaseAction
(SchemaManagementToolCoordinator.java:192)
at org.hibernate.tool.schema.spi.SchemaManagementToolCoordinator.process
(SchemaManagementToolCoordinator.java:73)
at org.hibernate.internal.SessionFactoryImpl.<init>(SessionFactoryImpl.java:316)
at
org.hibernate.boot.internal.SessionFactoryBuilderImpl.build(SessionFactoryBuilderImpl.java:469)
at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.build
(EntityManagerFactoryBuilderImpl.java:1259)
... 6 more
2023-07-18 18:48:47,713:ERROR:main:CgmsDbMigrationDriver: Migration failed. Exception:
07-18-2023 14:48:47 EDT: ERROR: Post migration failed. See log file for more information.
```

Execute the following query in the DB:

```
DECLARE
seq_name VARCHAR2(100);
current_increment PLS_INTEGER;
BEGIN
FOR seq IN (SELECT sequence_name FROM all_sequences WHERE sequence_owner = 'CGMS_DEV') LOOP
seq_name := seq.sequence_name;
IF seq_name NOT IN ('CONFIG_GROUPS_ID_SEQ', 'FIRMWARE_GROUPS_ID_SEQ',
'NET_C8000_METRICS_ID_SEQ',
'NET_C8000_PROPERTIES_ID_SEQ', 'NET_LGLFN_METRICS_ID_SEQ','NET_LGLFN_PROPERTIES_ID_SEQ')
THEN
-- Get the current increment for the sequence
EXECUTE IMMEDIATE 'SELECT INCREMENT_BY FROM ALL_SEQUENCES WHERE SEQUENCE_NAME = :seq and
sequence_owner = ''CGMS_DEV'''
INTO current_increment
USING seq_name;

IF current_increment <> 1 THEN
-- If the current increment is not 1, update it to 1
EXECUTE IMMEDIATE 'ALTER SEQUENCE ' || seq_name || ' INCREMENT BY 1';
END IF;
END IF;
END LOOP;
END;
/
```

Missing Endpoint Marker Line to CGR

If the FND is directly upgraded from version 4.6.1 to 4.8.1 by skipping the version 4.7.1, the location details are missing in the ROOT CAMs.

To manually add the location details:

Procedure

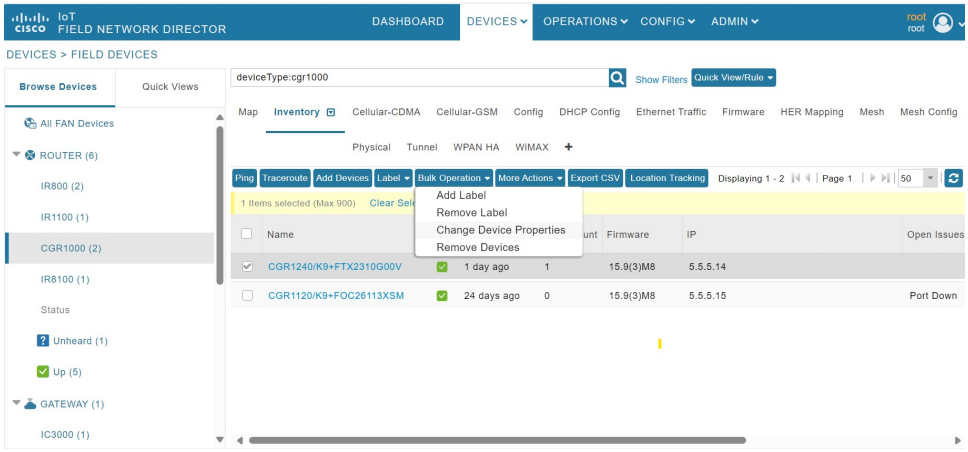
Step 1 Choose **DEVICES** > **Field Devices**.

Step 2 Select the **Browse Devices** tab from the left pane.

Step 3 Select **CGR1000** from the Router list and select the **Inventory** tab.

Step 4 Select the appropriate CGR check box.

Step 5 Click the **Bulk Operation** drop-down menu and select **Change Device Properties** option.



Step 6 Add the cam device latitude and logitude manually in the csv file.

Step 7 Click **Change** to upload the csv file.

Troubleshoot remote authentication failures when NTLMv2-only policy is enforced

When Cisco IoT FND remote authentication is configured with Microsoft NPS/Active Directory, remote GUI logins may fail if the AD environment enforces NTLMv2-only authentication.

This issue typically occurs when these policies are set:

- **Registry Configuration:**
  - **Path**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`
  - **Value**: LmCompatibilityLevel

- **Action**: Set to `5`

- **Group Policy (GPO):**
  - **Policy Path**: **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**
  - **Setting**: Network security: LAN Manager authentication level
  - **Action**: Set to **`Send NTLMv2 response only. Refuse LM & NTLM`**

With NTLMv2-only authentication, NPS may reject authentication attempts with Reason Code 16, and Cisco IoT FND logs may show errors such as "Remote authentication failed" or `com.cisco.cgms.exceptions.AAAException`.

Follow these steps to resolve remote authentication failures:

Procedure

Step 1 Open the Registry Editor (`regedit`) with administrative privileges.

Step 2 Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`.

Step 3 Locate the **RemoteAccess** key and expand it.

If the **RemoteAccess** key does not exist,

a) Right-click the **Services** key.
b) Select **New** > **Key**, and name it **RemoteAccess**.

Step 4 Expand the **RemoteAccess** to find the **Policy** key.

If the **Policy** key does not exist,

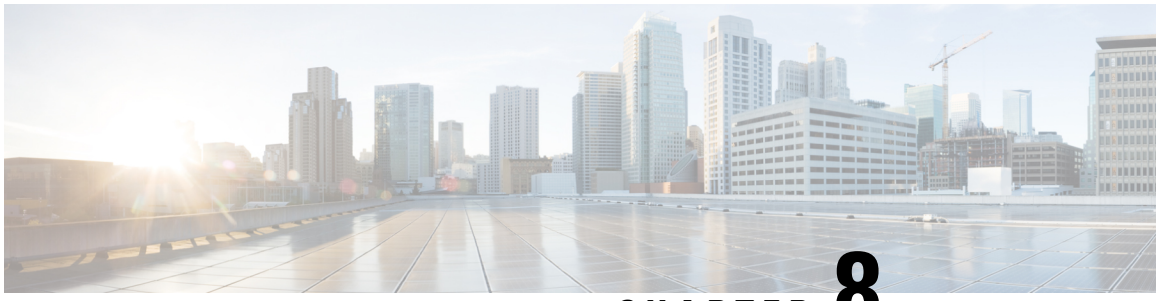
a) Right-click the **RemoteAccess** key.
b) Select **New** > **Key**, and name it **Policy**.

Step 5 With the **Policy** key selected, right-click the right-hand pane, select **New** > **DWORD (32-bit) Value**, and name it **Enable NTLMv2 Compatibility**.

Step 6 Double-click the **Enable NTLMv2 Compatibility** value and set its **Value data** to `1`.

Step 7 Restart both Network Policy Server and Remote Access services.

a) Open `services.msc`.
b) Right-click **Network Policy Server**, and select **Restart**.
c) Right-click **Routing and Remote Access**, and select **Restart**.



CHAPTER 8

Licensing Issues

This chapter explains some of the license issues and the possible resolutions.

- Device Import Failure, on page 91
- License File Upload Failure, on page 91

Device Import Failure

The importing of devices into IoT FND is dependent on the number of allotted IoT FND server licenses. Verify that your IoT FND server has the adequate license count available for the number and type of devices being imported into the IoT FND database.

Only unique device EIDs are allowed in IoT FND. Check that no one else imported this device EID into IoT FND or is currently trying to import the same device EID. Verify that no other user is simultaneously importing the same device into IoT FND.

License File Upload Failure

An expired license file will cause an error. Check the license file validity and expiration date.