



# ACI Sample Lab

Fabric External Data Collector  
Policies Configuration -  
**Callhome**



## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
<b>2</b>	<b>LAB REFERENCE &amp; TOPOLOGY INFORMATION .....</b>	<b>4</b>
<b>3</b>	<b>CREATE A SCHEDULER POLICY FOR CALL HOME FEATURE .....</b>	<b>5</b>
<b>4</b>	<b>CREATE EXTERNAL DATA COLLECTORS FOR CALL HOME FEATURE .....</b>	<b>9</b>
<b>5</b>	<b>CONFIGURE CALL HOME POLICY FOR THE ACI FABRIC .....</b>	<b>16</b>
<b>6</b>	<b>VERIFY CONFIGURATION OF CALL HOME FEATURE FOR THE ACI FABRIC</b>	<b>34</b>

## 1 Introduction

In the **External Data Collectors** menu of the Cisco APIC GUI, you can configure the sending of a variety of system data to **Call Home**, SNMP, or syslog destinations. When an event triggers the sending of a report, information from your selected queries is included in the report. You can configure a query based on a class name or a distinguished name, and you can further qualify the query based on subtrees and object properties.

Call Home provides an email-based notification for critical system policies. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to generate a case with the Technical Assistance Center.

- The Call Home feature can deliver alert messages containing information about diagnostics and environmental faults and events.
- The Call Home feature can deliver alerts to multiple recipients, referred to as Call Home destination profiles. Each profile includes configurable message formats and content categories. A predefined destination profile is provided for sending alerts to the Cisco TAC, but you also can define your own destination profiles.

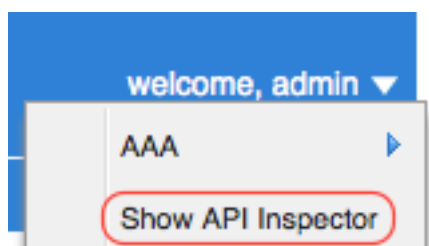
When you configure Call Home to send messages, Cisco ACI fabric system delivers Call Home messages in the following formats:

- **Short text format** which provides a one or two line description of the fault that is suitable for pagers or printed reports.
- XML machine readable format that uses **Extensible Markup Language (XML)** and **Adaptive Messaging Language (AML) XML schema definition (XSD)**. The AML XSD is published on the Cisco.com website at <http://www.cisco.com/>. The XML format enables communication with the Cisco Systems Technical Assistance Center.

The following lab involves configuring the Call Home feature in your Cisco ACI fabric system. This lab will show examples of configuring the Call Home feature utilizing the APIC Admin GUI and REST API (using POSTMAN). For this Lab, refer to Section 2 for your designated the fabric topology and your Pod's Application Server.

In regards to the REST API examples listed in this Lab, there is an assumption made that you have a REST CLIENT (like POSTMAN) installed on your workstation. This will be used for executing REST API requests to an APIC Controller. Also, while executing lab, you may want to open the API inspector

console from the APIC GUI. The API inspector displays the API POST requests used for the tasks performed. The Post Requests in the API inspector can be used for sending requests to APIC controllers.



## 2 Lab Reference & Topology Information

For the following sections in this lab, please use the following fabric information for the POD1 in your fabric pod assignments.

Device\Entity	NodeID	Fabric 1	Fabric 2
APIC 1 (OOB IP Address)	1	192.168.1.211	192.168.1.141
APIC 2 (OOB IP Address)	2	192.168.1.212	192.168.1.142
APIC 3 (OOB IP Address)	3	192.168.1.213	192.168.1.143
Spine 1 (OOB IP Address)	201	192.168.1.244	192.168.1.130
Spine 2 (OOB IP Address)	202	192.168.1.245	192.168.1.131
Leaf 1 (OOB IP Address)	101	192.168.1.241	192.168.1.128
Leaf 2 (OOB IP Address)	102	192.168.1.242	192.168.1.135
Leaf 3 (OOB IP Address)	103	192.168.1.243	192.168.1.136
Leaf 4 (OOB IP Address)	104	192.168.1.154	192.168.1.137
OOB Default Gateway		192.168.1.1 / 24	192.168.1.1 / 24

Device\Entity	IP ADDRESS	RDP Username>Password
ACI-P1-Server	192.168.1.77	Administrator\Cisco123!
ACI-P2-Server	192.168.1.78	Administrator\Cisco123!
ACI-P3-Server	192.168.1.79	Administrator\Cisco123!
ACI-P4-Server	192.168.1.120	Administrator\Cisco123!
ACI-P5-Server	192.168.1.207	Administrator\Cisco123!
ACI-P6-Server	192.168.1.208	Administrator\Cisco123!

Device\Entity	SMTP Server	Email SOURCE	Email TARGET
ACI-P1-Server	192.169.1.161	aci-p1-server@aci.bootcamp.local	aci-bc-p1@cisco.com
ACI-P2-Server	192.169.1.161	aci-p2-server@aci.bootcamp.local	aci-bc-p2@cisco.com
ACI-P3-Server	192.169.1.161	aci-p3-server@aci.bootcamp.local	aci-bc-p3@cisco.com
ACI-P4-Server	192.169.1.161	aci-p4-server@aci.bootcamp.local	aci-bc-p4@cisco.com
ACI-P5-Server	192.169.1.161	aci-p5-server@aci.bootcamp.local	aci-bc-p5@cisco.com
ACI-P6 Server	192.169.1.161	aci-p6-server@aci.bootcamp.local	aci-bc-p6@cisco.com

Note: If you RDP into your Application Services Server, you can find the files under the C:\ACI\ directory tree location.

### 3 Create a SCHEDULER Policy for Call Home Feature

For this lab section, you will create a Scheduler Policy for the Call Home Feature that will be configured in your designated ACI Fabric.

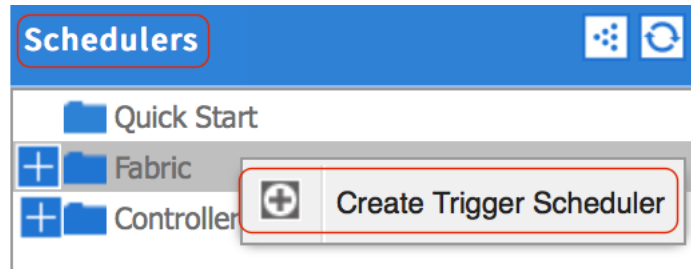
This lab section will:

- **Create a Scheduler Policy for Call Home Feature.**
- **Create a One-Time Scheduler Window for Scheduler Policy.**
- **Create a Recurring Scheduler Window for Scheduler Policy.**

*Note: You can re-use Scheduler Policies in the ACI Fabric Policies. This lab is configuring a Scheduler Policy specifically to be used with Call Home Feature.*

- On the menu bar, choose **ADMIN > SCHEDULERS**. In the Navigation pane, select **FABRIC**. Right-click and click **CREATE TRIGGER SCHEDULE**. In the Create Trigger Schedule dialog box, perform the following actions:
  - Enter name ([fab-callhome-sched](#))
  - Expand **SCHEDULE WINDOWS** by Clicking on the “ + “
- Create an **ONE-TIME** Window
  - Select Window Type ([One Time](#))
  - Enter Window Name ([fab-onetime](#))
  - Select Date ([Click Calendar icon](#), [Click NOW](#), and [Click OK](#))
  - Enter Maximum Concurrent Nodes ([10](#))
  - Enter Maximum Running Time ([unlimited](#))
  - Click OK

- Create a **RECURRING** Window
  - Select Window Type (**Recurring**)
  - Enter Window Name (**fab-recurring**)
  - Select Day (**every-day**)
  - Select Hour (**1**)
  - Select Minute (**0**)
  - Enter Maximum Concurrent Nodes (**10**)
  - Enter Maximum Running Time (**unlimited**)
  - Click OK
- Click **SUBMIT**

A screenshot of a dialog box titled "CREATE SCHEDULE WINDOW". The dialog contains the following fields and options:

- Window Type:** Radio buttons for "One Time" (selected) and "Recurring".
- Window Name:** Text input field containing "fab-onetime".
- Date:** Text input field containing "2015-01-09 16:15:55 PM". A calendar icon is visible to the right of the field.
- Format:** Text below the date field: "Format: YYYY-MM-DD HH:MM:SS AM/PM".
- Maximum Concurrent Nodes:** Spin box with the value "10".
- Maximum Running Time (dd:hh:mm:ss):** Text input field containing "unlimited".

At the bottom right, there are "OK" and "CANCEL" buttons.

## CREATE SCHEDULE WINDOW

Create Schedule Window

Window Type:  One Time  
 Recurring

Window Name: fab-recurring

Day: every-day

Hour: 1

Minute: 0

Maximum Concurrent Nodes: 10

Maximum Running Time (dd:hh:mm:ss): unlimited

OK CANCEL

## CREATE TRIGGER SCHEDULER

Create scheduler with schedule windows

### SCHEDULER

Name: fab-callhome-sched

Description: Task 3 - Create a SCHEDULER Policy for Call Home Feature

Schedule Windows:

Name	When	Max Concurrent Nodes	Max Running Time (dd:hh:mm:ss)
fab-onetime	One time at Fri, 09 Jan 2015 21:15:55 GMT	10	Unlimited
fab-recurring	Repeat every-day at 01:00	10	Unlimited

## Trigger Scheduler - fab-callhome-sched

POLICY HISTOR

PROPERTIES

Name: **fab-callhome-sched**

Description: Task 3 - Create a SCHEDULER Policy for Call Home Feature

Recurring Windows:

NAME	DAY	HOUR	MINUTE	MAXIMUM CONCURRENT NODES	MAXIMUM RUNNING TIME (DD:HH:MM:SS)
fab-recurring	every-day	1	0	10	Unlimited

One Time Windows:

NAME	DATE	MAXIMUM CONCURRENT NODES	MAXIMUM RUNNING TIME (DD:HH:MM:SS)
fab-onetime	2015-01-09T16:15:55.0...	10	Unlimited

### API EXAMPLE

#### Task 3 - Create a SCHEDULER Policy for Call Home Feature

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/schedp-fab-callhome-sched.json>

payload

```
{ "trigSchedP": { "attributes": { "dn": "uni/fabric/schedp-fab-callhome-sched", "descr": "Task 3 - Create a SCHEDULER Policy for Call Home Feature", "name": "fab-callhome-sched", "rn": "schedp-fab-callhome-sched", "status": "created", "children": [ { "trigAbsWindowP": { "attributes": { "dn": "uni/fabric/schedp-fab-callhome-sched/abswinp-fab-onetime", "name": "fab-onetime", "date": "2015-01-09T21:15:55.000Z", "concurCap": "10", "rn": "abswinp-fab-onetime", "status": "created", "children": [] }, "trigRecurrWindowP": { "attributes": { "dn": "uni/fabric/schedp-fab-callhome-sched/recurwinp-fab-recurring", "name": "fab-recurring", "hour": "1", "concurCap": "10", "rn": "recurwinp-fab-recurring", "status": "created", "children": [] } } ] } } }
```



## 4 Create External Data Collectors for Call Home Feature

For this lab section, you will configure External Data Collectors for the Call Home Feature to use in your designated ACI fabric.

This lab section will:

- **Create Callhome Monitoring Destination Group.**
- **Create Callhome Query Group.**

*Note: As mentioned above, Please use Section 2 above to use for a reference for your designated Fabric, your Application Server, and your Call Home Email configuration settings.*

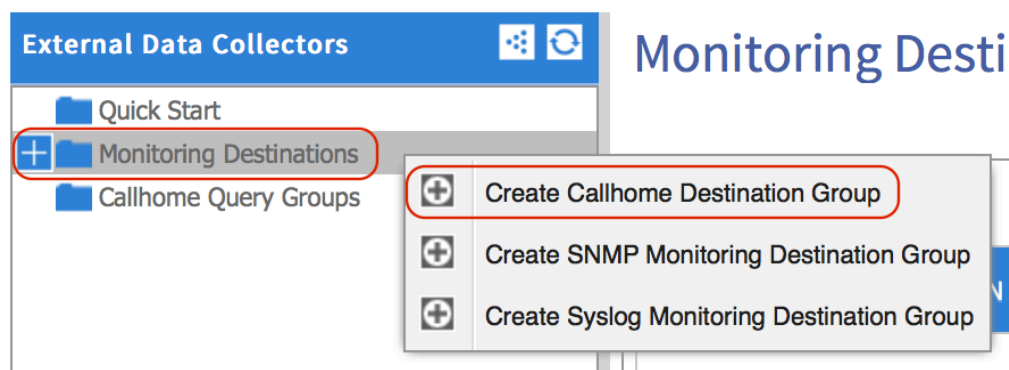
### STEP 1: Create Callhome Monitoring Destination Group.

Callhome data can be collected and exported to a destination group for logging and evaluation. The Create Callhome Destination Group screen contains properties for specifying a callhome destination group.

The Callhome Destination Group will be used by the Cisco ACI Fabric for sending Callhome messages. Use the GUI to create Callhome Destination Group. Use the API Inspector to capture the API POST information from this configuration.

**Task 1.1** Use the GUI to a create callhome destination group. For this task, use the admin user "admin" and the password "Aci123bc".

- On the menu bar, choose **ADMIN > EXTERNAL DATA COLLECTORS**. In the Navigation pane, select **MONITORING DESTINATIONS**. Right-click and click **CREATE CALLHOME DESTINATION GROUP**. In the Create Callhome destination group dialog box, perform the following actions:



## Task 1.1.1 DEFINE A GROUP NAME AND PROFILE

- Enter **Name** (*fab-callhome-destGrp*)
- Enter **Description** (*ACI Bootcamp Lab for CALLHOME*)
- Select **Admin State** (*enabled*)
- Enter **Port Number** (*25*)
- Enter **SMTP Server** (*192.169.1.161*)
- Select **Management EPG** (*default (Out-of-Band)*)
- Enter **From Email** (*aci-p1-server@aci.bootcamp.local*)
- Enter **Reply To Email** (*aci-bc-p1@cisco.com*)
- Enter **Customer Contact Email** (*aci-bc-p1@cisco.com*)
- Enter **Phone Contact** (*+1-919-392-1234*)
- Enter **Contact Information** (*Robert Hurst*)
- Enter **Street Address** (*123 ACI Bootcamp, Cisco*)
- Enter **Contact Id** (*1234567890*)
- Enter **Customer Id** (*12345*)
- Enter **Site Id** (*67890*)
- Click **NEXT**

## CREATE CALLHOME DESTINATION GROUP

STEP 1 > PROFILE

1. PROFILE

Define Group Name and Profile

Name:	<input type="text" value="fab-callhome-destGrp"/>
Description:	<input type="text" value="ACI Bootcamp Lab for CALLHOME"/>
Admin State:	<input type="text" value="enabled"/>
Port Number:	<input type="text" value="25"/>
SMTP Server:	<input type="text" value="173.37.93.161"/>
Management EPG:	<input type="text" value="default (Out-of-Band)"/>
From Email:	<input type="text" value="aci-p1-server@aci.bootcamp.local"/>
Reply To Email:	<input type="text" value="aci-bc-p1@cisco.com"/>
Customer Contact Email:	<input type="text" value="aci-bc-p1@cisco.com"/>
Phone Contact:	<input type="text" value="+1-919-392-1234"/> <small>e.g., +1-011-408-555-1212</small>
Contact Information:	<input type="text" value="Robert Hurst"/>
Street Address:	<input type="text" value="123 ACI Bootcamp, Cisco"/>
Contract Id:	<input type="text" value="1234567890"/>
Customer Id:	<input type="text" value="12345"/>
Site Id:	<input type="text" value="67890"/>

## Task 1.1.2 CREATE CALLHOME DESTINATION GROUP

From the "Create Callhome Monitoring Destination Group" wizard, Create Callhome Remote Destinations. Click on the "+" to **CREATE CALLHOME DESTINATION**. In the Create Callhome Destination dialog box, perform the following actions:

- Enter **Name** (ACI-P1-Server)
- Select **Admin State** (enabled)
- Select **Level** (critical)
- Select **Email** (aci-bc-p1@cisco.com)
- Select **Format** (xml)
- Select **Maximum Size (Bytes)** (1000000)
- Click **UPDATE**

### CREATE CALLHOME DESTINATION GROUP

STEP 2 > DESTINATIONS

1. PROFILE 2. DESTINATIONS

Create Destinations

Name	Admin State	Level	Email	Format	Maximum Size (Bytes)
ACI-P1-Server	enabled	information	aci-bc-p1@cisco.com	xml	1000000

*Note: Create 2 more Callhome Destinations. Configure each callhome destination to use different email format ( short-txt and aml). Depending if you are in an Odd or Even Pod, use the remaining Odd\Even Servers in the grouping for your remaining Callhome Destinations. For example, (P1, P3, P5) and (P2, P4, P6).*

### CREATE CALLHOME DESTINATION GROUP

STEP 2 > DESTINATIONS

1. PROFILE 2. DESTINATIONS

Create Destinations

Name	Admin State	Level	Email	Format	Maximum Size (Bytes)
ACI-P1-Server	enabled	information	aci-bc-p1@cisco.com	xml	1000000
ACI-P3-Server	enabled	information	aci-bc-p3@cisco.com	short-txt	1000000
ACI-P5-Server	enabled	information	aci-bc-p5@cisco.com	aml	1000000

- Click **FINISH**

## Monitoring Destinations



NAME	ADMIN STATE	PORT NUMBER	FROM EMAIL	REPLY TO EMAIL	CUSTOMER CONTACT EMAIL
fab-callhome-destGrp	enabled	25	aci-p1-server@aci.b...	aci-bc-p1@cisco.com	aci-bc-p1@cisco.com

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Callhome Monitoring Destination Group. You can use this APIC Example and use POSTMAN REST Client to create the callhome monitoring destination group.

### **API EXAMPLE - STEP 1: Create Callhome Monitoring Destination Group.**

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/chgroup-fab-callhome-destGrp.json>

payload

```
{
  "callhomeGroup": {
    "attributes": {
      "dn": "uni/fabric/chgroup-fab-callhome-destGrp",
      "name": "fab-callhome-destGrp",
      "descr": "ACI Bootcamp Lab for CALLHOME",
      "rn": "chgroup-fab-callhome-destGrp",
      "status": "created",
      "children": [
        {
          "callhomeDest": {
            "attributes": {
              "dn": "uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P1-Server",
              "name": "ACI-P1-Server",
              "urgency": "info",
              "email": "aci-bc-p1@cisco.com",
              "rn": "dest-ACI-P1-Server",
              "status": "created",
              "children": []
            },
            "callhomeDest": {
              "attributes": {
                "dn": "uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P3-Server",
                "name": "ACI-P3-Server",
                "urgency": "info",
                "email": "aci-bc-p3@cisco.com",
                "format": "short-txt",
                "rn": "dest-ACI-P3-Server",
                "status": "created",
                "children": []
              },
              "callhomeDest": {
                "attributes": {
                  "dn": "uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P5-Server",
                  "name": "ACI-P5-Server",
                  "urgency": "info",
                  "email": "aci-bc-p5@cisco.com",
                  "format": "aml",
                  "rn": "dest-ACI-P5-Server",
                  "status": "created",
                  "children": []
                },
                "callhomeProf": {
                  "attributes": {
                    "dn": "uni/fabric/chgroup-fab-callhome-destGrp/prof",
                    "from": "aci-p1-server@aci.bootcamp.local",
                    "replyTo": "aci-bc-p1@cisco.com",
                    "email": "aci-bc-p1@cisco.com",
                    "phone": "+1-919-392-1234",
                    "contact": "Robert Hurst",
                    "addr": "123 ACI Bootcamp, Cisco",
                    "contract": "1234567890",
                    "customer": "12345",
                    "site": "67890",
                    "rn": "prof",
                    "status": "created",
                    "children": [
                      {
                        "callhomeSmtServer": {
                          "attributes": {
                            "dn": "uni/fabric/chgroup-fab-callhome-destGrp/prof/smt",
                            "host": "192.169.1.161",
                            "rn": "smt",
                            "status": "created",
                            "children": [
                              {
                                "fileRsARemoteHostToEpg": {
                                  "attributes": {
                                    "tDn": "uni/tn-mgmt/mgmtmp-default/oob-default",
                                    "status": "created",
                                    "children": []
                                  }
                                }
                              }
                            ]
                          }
                        }
                      }
                    ]
                  }
                }
              }
            }
          }
        }
      ]
    }
  }
}
```

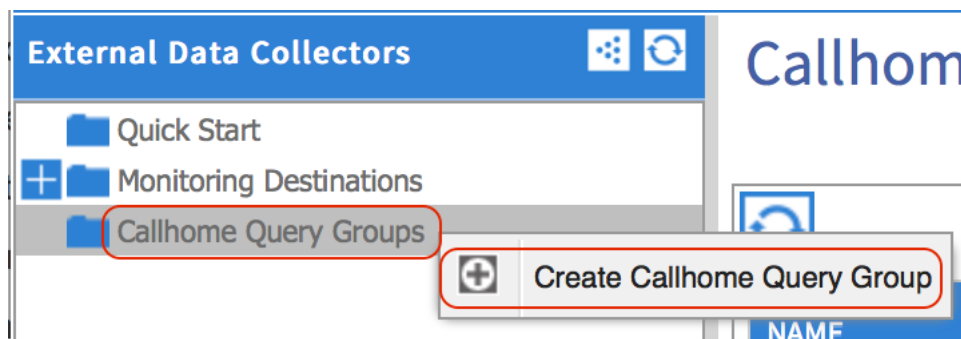
## STEP 2: Create Callhome Query Group.

A Call Home query group, which is a set of Call Home query messages to be sent on returned objects. The Callhome Query Groups panel provides the options to create or delete a callhome query group and displays a summary table that lists the existing query groups.

The Callhome Query Group will be used by the Cisco ACI Fabric for sending Callhome messages. Use the GUI to create Callhome Query Group. Use the API Inspector to capture the API POST information from this configuration.

**Task 2.1 Use the GUI to a create callhome query group. For this task, use the admin user "admin" and the password "Aci123bc".**

- On the menu bar, choose **ADMIN > EXTERNAL DATA COLLECTORS**. In the Navigation pane, select **CALLHOME QUERY GROUPS**. Right-click and click **CREATE CALLHOME QUERY GROUP**. In the Create Callhome Query Group dialog box, perform the following actions:
  - Enter **Name** (*fab-queryGroup*)
  - Click on the " + " to **ADD QUERIES**. In the Create Query dialog box, perform the following actions:
    - Enter **Name** (*fab-query-1*)
    - Select **Type** (*dn*)
    - Enter **DN or Class Name** (*leave blank*)
    - Select **Target** (*subtree*)
    - Select **Response Subtree** (*full*)
    - Select **Response Subtree Include** (*Check "ALL" Boxes*)
    - Click **OK**
  - Click **SUBMIT**



# CREATE QUERY

## Define the Query

Name:

Type:  class  
 dn

DN or Class Name:

Target:  subtree

children

self

Response Subtree:  full

no

children

- Response Subtree Include:
- event-logs
  - count
  - stats
  - state
  - port-deployment
  - tasks
  - relations-with-parent
  - health
  - add-mo-list
  - fault-count
  - local-prefix
  - config-only
  - record-subtree
  - no-scoped
  - relations
  - health-records
  - audit-logs
  - deployment
  - required

OK

### CREATE CALLHOME QUERY GROUP

Define Name and Queries

Name:

ADD QUERIES

Name	Query Type	DN or Class Name	Query Target	Response Subtree	Response Subtree Include
					relations-with-parent health add-mo-list fault-count local-prefix config-only record-subtree no-scoped relations health-records audit-logs deployment required faults fault-records

SUBMIT CANCEL

## Callhome Query Groups

NAME	TYPE	DN OR CLASS NAME	TARGET	RESPONSE SUBTREE	RESPONSE SUBTREE INCLUDE
fab-queryGroup					
fab-query-1	dn		subtree	full	add-mo-list,audit-log...

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Callhome Monitoring Destination Group. You can use this APIC Example and use POSTMAN REST Client to create the Callhome monitoring destination group.

### API EXAMPLE - STEP 2: Create Callhome Query Group.

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/chquerygroup-fab-queryGroup.json>

payload

```
{
  "callhomeQueryGroup": {
    "attributes": {
      "dn": "uni/fabric/chquerygroup-fab-queryGroup",
      "name": "fab-queryGroup",
      "rn": "chquerygroup-fab-queryGroup",
      "status": "created"
    },
    "children": [
      {
        "callhomeQuery": {
          "attributes": {
            "dn": "uni/fabric/chquerygroup-fab-queryGroup/chquery-fab-query-1",
            "name": "fab-query-1",
            "target": "subtree",
            "rspSubtree": "full",
            "rspSubtreeInclude": "event-logs,count,stats,state,port-deployment,tasks,relations-with-parent,health,add-mo-list,fault-count,local-prefix,config-only,record-subtree,no-scoped,relations,health-records,audit-logs,deployment,required,faults,fault-records",
            "rn": "chquery-fab-query-1",
            "status": "created"
          },
          "children": []
        }
      }
    ]
  }
}
```

## 5 Configure Call Home Policy for the ACI Fabric

For this lab section, you will configure Call Home Feature for Controllers and LeafSpine Switches in your designated ACI fabric.

This lab section will:

- **Create Switch Policy for CALLHOME Inventory Policy.**
- **Configure FABRIC > FABRIC POLICIES to send CALLHOME messages to Callhome Destinations.**
- **Configure FABRIC > ACCESS POLICIES to send CALLHOME messages to Callhome Destinations.**

*Note: As mentioned above, Please use Section 2 above to use for a reference for your designated Fabric and your Application Server.*

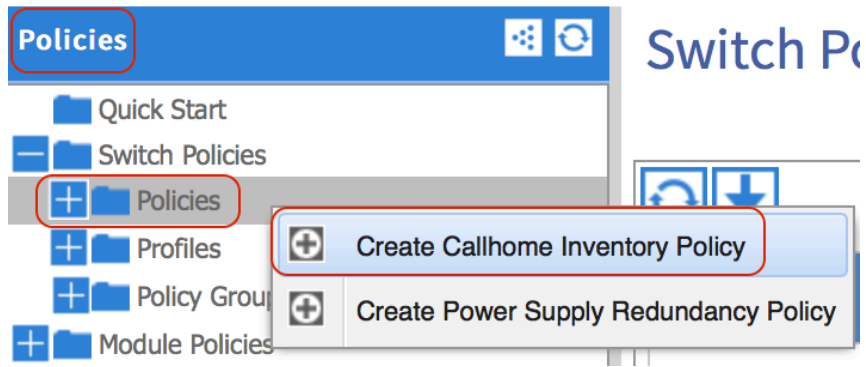
### STEP 1: Create Switch Policy for CALLHOME Inventory Policy.

The call home inventory policy is used to send Call Home messages with system inventory information in relation to your designated Cisco ACI Fabric. Use the GUI to create the Switch Policy for the Call Home Feature. Use the API Inspector to capture the API POST information from this configuration.

**Task 1.1 Use the GUI to a create Switch Policy for Callhome Inventory Policy. For this task, use the admin user "admin" and the password "Aci123bc".**

- On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **SWITCH POLICIES**. Select **POLICIES**. Right-click and click **CREATE CALLHOME INVENTORY POLICY**. In the Create Callhome Inventory Policy dialog box, perform the following actions:
  - Enter **Name** (*fab-callhome-inventory*)
  - Select **Start Now** (*enabled*)
  - Select **Maximum Retry Count** (*1*)
  - Select **Destination Group** (*fab-callhome-destGrp*) **\*\* created earlier \*\***
  - Select **Scheduler** (*fab-callhome-sched*) **\*\* created earlier \*\***
  - Click **SUBMIT**





## CREATE CALLHOME INVENTORY POLICY i x

Define the Inventory Policy

Name:

Start Now:  enabled  
 disabled

Maximum Retry Count:

Destination Group:

Scheduler:

## Callhome Inventory Policy - fab-callhome-inventory

**PROPERTIES**

Name: **fab-callhome-inventory**

Start Now:  enabled  
 disabled

Maximum Retry Count: 1

Destination Group: fab-callhome-destGrp

Scheduler: fab-callhome-sched

Last Success: **never**

Last Attempt: **1969-12-31T20:00:00.000-04:00**

Retry Count: **0**

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Switch Policy for CALLHOME Inventory Policy. You can use this APIC Example and use POSTMAN REST Client to create the Switch Policy for CALLHOME Inventory Policy.

### API EXAMPLE

#### **STEP 1: Create Switch Policy for CALLHOME Inventory Policy.**

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/chinvp-fab-callhome-inventory.json>

payload

```
{"callhomeInvP":{"attributes":{"dn":"uni/fabric/chinvp-fab-callhome-inventory","name":"fab-callhome-inventory","rn":"chinvp-fab-callhome-inventory","status":"created"},"children":[{"callhomeRsDestGroupRel":{"attributes":{"tDn":"uni/fabric/chgroup-fab-callhome-destGrp","status":"created"},"children":[]},"callhomeRsInvScheduler":{"attributes":{"tnTrigSchedPName":"fab-callhome-sched","status":"created,modified"},"children":[]}}]}
```

## Task 1.2 CREATE SWITCH POLICY GROUP

This Task may not be necessary in an “existing” ACI Fabric with previously configured switch policies. A Switch Policy Group for Leaf and Spine Switches may already be in use. If so, all you would need to do is EDIT the existing Policy Group by adding the Callhome Inventory Policy you created earlier. For this Lab, there should not be an existing Switch Policy Group. So you will need to create Switch Policy Groups for your designated ACI Fabric.

This lab section will:

- **Create Spine Switch Policy Group.**
- **Create Leaf Switch Policy Group.**

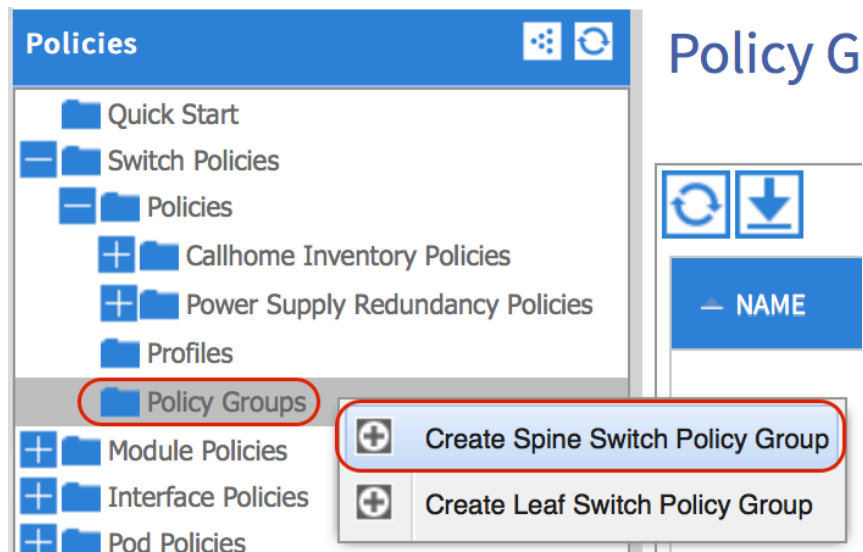
### Task 1.2.1 CREATE SPINE SWITCH POLICY GROUP

On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **SWITCH POLICIES**. Expand **POLICIES**. Select **POLICY GROUPS**. Right-click and click **CREATE SPINE SWITCH POLICY GROUP**. In the Create Spine Switch Policy Group dialog box, perform the following actions:

- Enter **Name** (*fab-spine-policyGroup*)
- Select **Inventory Policy** (*fab-callhome-inventory*) **\*\* created earlier \*\***

*Note: Since this is a NEW Switch Policy Group, just leave the other listed Policies blank. You will return to this Switch Policy Group in a later Lab.*

- Click **SUBMIT**



# CREATE SPINE SWITCH POLICY GROUP

Specify the Policy Group Properties

Name:	fab-spine-policyGroup
Description:	ACI Bootcamp Lab Task 1.2.1 CREATE SPINE SWITCH POLICY GROUP
Monitoring Policy:	select or type to pre-provision
TechSupport Export Policy:	select or type to pre-provision
Core Export Policy:	select or type to pre-provision
Inventory Policy:	fab-callhome-inventory
Power Redundancy Policy:	select or type to pre-provision

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Spine Switch Policy Group. You can use this APIC Example and use POSTMAN REST Client to create the Spine Switch Policy Group.

## API EXAMPLE

### Task 1.2.1 CREATE SPINE SWITCH POLICY GROUP

method: [POST](#)

url:

<https://192.168.1.141/api/node/mo/uni/fabric/funcprof/spnodepgrp-fab-spine-policyGroup.json>

payload

```
{"fabricSpNodePGrp":{"attributes":{"dn":"uni/fabric/funcprof/spnodepgrp-fab-spine-policyGroup","name":"fab-spine-policyGroup","descr":"ACI Bootcamp Lab Task 1.2.1 CREATE SPINE SWITCH POLICY GROUP","rn":"spnodepgrp-fab-spine-policyGroup","status":"created"},"children":[{"fabricRsCallhomeInvPol":{"attributes":{"tnC allhomeInvPName":"fab-callhome-inventory","status":"created,modified"},"children":[]}}]}
```

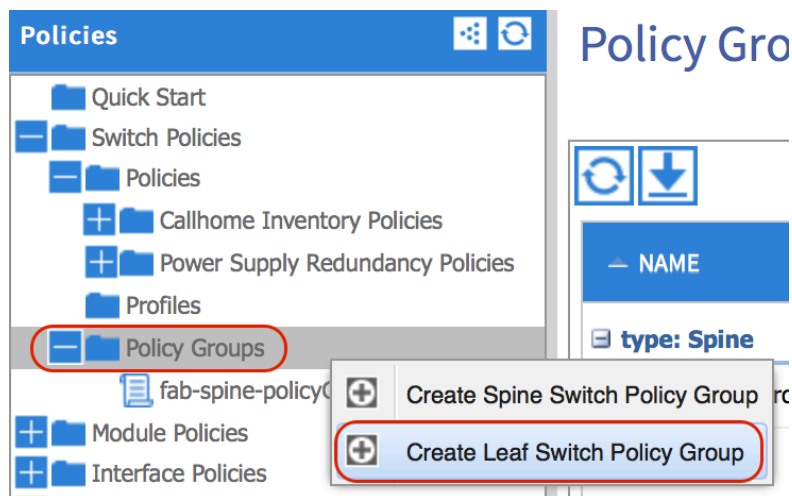
## Task 1.2.2 CREATE LEAF SWITCH POLICY GROUP

On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **SWITCH POLICIES**. Expand **POLICIES**. Select **POLICY GROUPS**. Right-click and click **CREATE LEAF SWITCH POLICY GROUP**. In the Create Leaf Switch Policy Group dialog box, perform the following actions:

- Enter **Name** (`fab-leaf-policyGroup`)
- Select **Inventory Policy** (`fab-callhome-inventory`) *\*\* created earlier \*\**

*Note: Since this is a NEW Switch Policy Group, just leave the other listed Policies blank. You will return to this Switch Policy Group in a later Lab.*

- Click **SUBMIT**



## CREATE LEAF SWITCH POLICY GROUP

### Specify the Policy Group Properties

Name:	<input type="text" value="fab-leaf-policyGroup"/>
Description:	<input type="text" value="ACI Bootcamp Lab Task 1.2.2 CREATE LEAF SWITCH POLICY GROUP"/>
Monitoring Policy:	<input type="text" value="select or type to pre-provision"/>
TechSupport Export Policy:	<input type="text" value="select or type to pre-provision"/>
Core Export Policy:	<input type="text" value="select or type to pre-provision"/>
Inventory Policy:	<input type="text" value="fab-callhome-inventory"/>
Power Redundancy Policy:	<input type="text" value="select or type to pre-provision"/>

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Leaf Switch Policy Group. You can use this APIC Example and use POSTMAN REST Client to create the Leaf Switch Policy Group.

## **API EXAMPLE**

### **Task 1.2.2 CREATE LEAF SWITCH POLICY GROUP**

method: **POST**

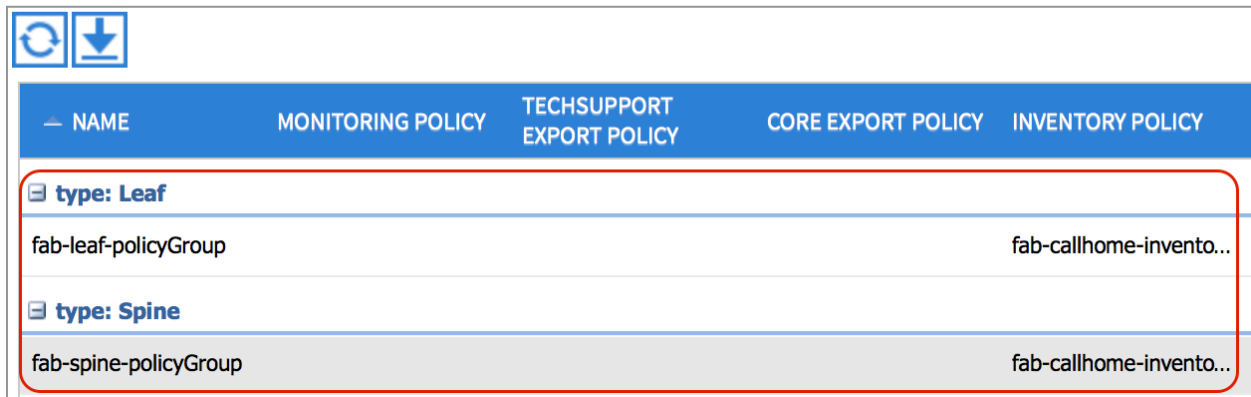
url:

<https://192.168.1.141/api/node/mo/uni/fabric/funcprof/lenodepgrp-fab-leaf-policyGroup.json>

payload

```
{ "fabricLeNodePGrp": { "attributes": { "dn": "uni/fabric/funcprof/lenodepgrp-fab-leaf-policyGroup", "name": "fab-leaf-policyGroup", "descr": "ACI Bootcamp Lab Task 1.2.2 CREATE LEAF SWITCH POLICY GROUP", "rn": "lenodepgrp-fab-leaf-policyGroup", "status": "created" }, "children": [ { "fabricRsCallhomeInvPol": { "attributes": { "tnC allhomeInvPName": "fab-callhome-inventory", "status": "created,modified" }, "children": [] } } ] }
```

## Policy Groups



NAME	MONITORING POLICY	TECHSUPPORT EXPORT POLICY	CORE EXPORT POLICY	INVENTORY POLICY
fab-leaf-policyGroup				fab-callhome-invento...
fab-spine-policyGroup				fab-callhome-invento...

### Task 1.3 CREATE SWITCH POLICY PROFILES

Like Task 1.2, This Task may not be necessary in an “existing” ACI Fabric with previously configured switch policies. A Switch Policy Profile for Leaf and Spine Switches may already be in use. If so, all you would need to do is VERIFY the existing Policy Group that you added the Callhome Inventory Policy to is “associated with the required Switches” in the Switch Profiles. For this Lab, there should not be an existing Switch Profile for Leaf and Spine Switches. So you will need to create Switch Profile for Leaf and Spine Switches for your designated ACI Fabric.

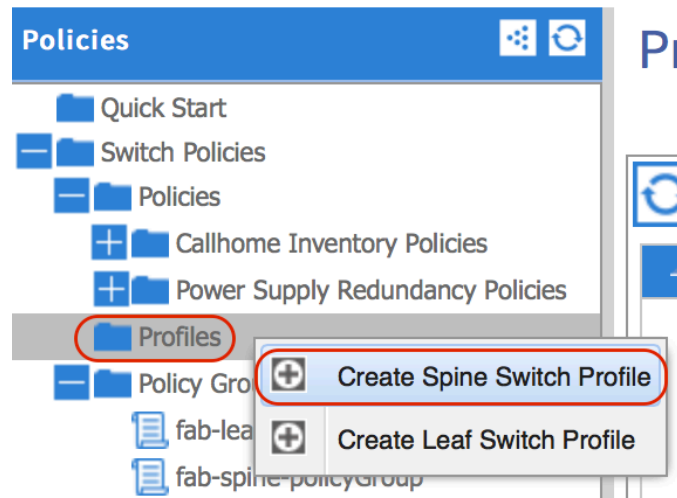
This lab section will:

- **Create Spine Switch Profile.**
- **Create Leaf Switch Profile.**

#### Task 1.3.1 CREATE SPINE SWITCH POLICY PROFILE

On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **SWITCH POLICIES**. Expand **POLICIES**. Select **PROFILES**. Right-click and click **CREATE SPINE SWITCH PROFILE**. In the Create Spine Switch Profile dialog box, perform the following actions:

- Enter **Name** (*fab-spine-swProfile*)
- Click on the " + " to **SWITCH ASSOCIATIONS**. In the Switch Associations Panel, perform the following actions:
  - Enter **Name** (*fab-spines-swassoc*)
  - Select **Blocks** (*201-202*) *\*\* selected from Block drop down list \*\**
  - Select **Blocks** (*fab-spine-policyGroup*) *\*\* created earlier \*\**
  - Click **UPDATE**
- Click **SUBMIT**



## CREATE SPINE SWITCH PROFILE

Specify the association profile Identity

Name:

Description:

Switch Associations:

Name	Blocks	Policy Group
fab-spines-swassoc	201,202	fab-spine-policyGroup

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Spine Switch Policy Profile. You can use this APIC Example and use POSTMAN REST Client to create the Spine Switch Policy Profile.

### API EXAMPLE

#### Task 1.3.1 CREATE SPINE SWITCH POLICY PROFILE

method: [POST](#)

url:

<https://192.168.1.141/api/node/mo/uni/fabric/spprof-fab-spine-swProfile.json>

payload

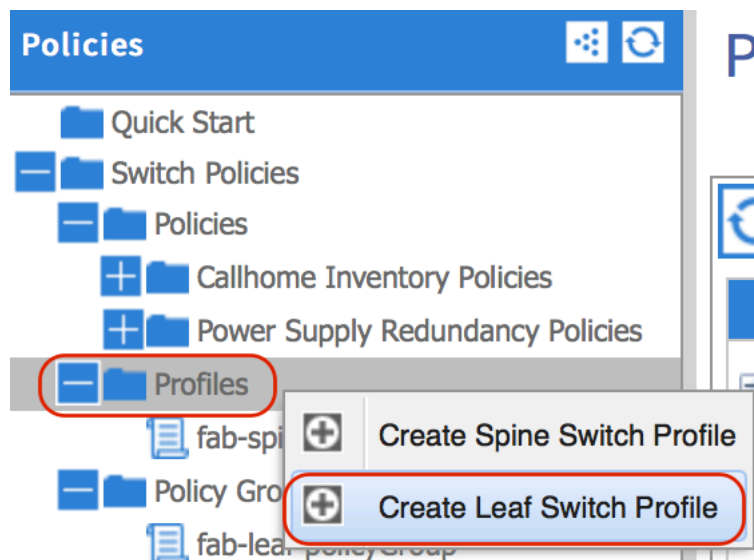
```
{
  "fabricSpineP": {
    "attributes": {
      "dn": "uni/fabric/spprof-fab-spine-swProfile",
      "name": "fab-spine-swProfile",
      "descr": "ACI Bootcamp Lab Task 1.3.1 CREATE SPINE SWITCH POLICY PROFILE",
      "rn": "spprof-fab-spine-swProfile",
      "status": "created,modified"
    },
    "children": [
      {
        "fabricSpineS": {
          "attributes": {
            "dn": "uni/fabric/spprof-fab-spine-swProfile/spines-fab-spines-swassoc-typrange",
            "type": "range",
            "name": "fab-spines-swassoc",
            "rn": "spines-fab-spines-swassoc-typrange",
            "status": "created,modified"
          },
          "children": [
            {
              "fabricNodeBlk": {
                "attributes": {
                  "dn": "uni/fabric/spprof-fab-spine-swProfile/spines-fab-spines-swassoc-typrange/nodeblk-b44c4439ea8ad834",
                  "from_": "201",
                  "to_": "202",
                  "name": "b44c4439ea8ad834",
                  "rn": "nodeblk-b44c4439ea8ad834",
                  "status": "created,modified"
                },
                "children": []
              },
              "fabricRsSpNodePGrp": {
                "attributes": {
                  "tDn": "uni/fabric/funcprof/spnodepgrp-fab-spine-policyGroup",
                  "status": "created"
                },
                "children": []
              }
            }
          ]
        }
      }
    ]
  }
}
```



### Task 1.3.2 CREATE LEAF SWITCH POLICY PROFILE

On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **SWITCH POLICIES**. Expand **POLICIES**. Select **PROFILES**. Right-click and click **CREATE LEAF SWITCH PROFILE**. In the Create Leaf Switch Profile dialog box, perform the following actions:

- Enter **Name** (`fab-leaf-swProfile`)
- Click on the " + " to **SWITCH ASSOCIATIONS**. In the Switch Associations Panel, perform the following actions:
  - Enter **Name** (`fab-leafs-swassoc`)
  - Select **Blocks** (`101-104`) *\*\* selected from Block drop down list \*\**
  - Select **Blocks** (`fab-leaf-policyGroup`) *\*\* created earlier \*\**
  - Click **UPDATE**
- Click **SUBMIT**



## CREATE LEAF SWITCH PROFILE

Specify the profile Identity

Name:

Description:

Switch Associations:

Name	Blocks	Policy Group
fab-leafs-swassoc	101,102,103,104	fab-leaf-policyGroup

Using the **APIC API Inspector**, this API Example was captured from the POST request to Create Leaf Switch Policy Profile. You can use this APIC Example and use POSTMAN REST Client to create the Leaf Switch Policy Profile.

## API EXAMPLE

### Task 1.3.2 CREATE LEAF SWITCH POLICY PROFILE

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/leprof-fab-leaf-swProfile.json>

payload

```
{
  "fabricLeafP": {
    "attributes": {
      "dn": "uni/fabric/leprof-fab-leaf-swProfile",
      "name": "fab-leaf-swProfile",
      "descr": "ACI Bootcamp Lab Task 1.3.2 CREATE LEAF SWITCH POLICY PROFILE",
      "rn": "leprof-fab-leaf-swProfile",
      "status": "created,modified"
    },
    "children": [
      {
        "fabricLeafS": {
          "attributes": {
            "dn": "uni/fabric/leprof-fab-leaf-swProfile/leaves-fab-leafs-swassoc-typ-range",
            "type": "range",
            "name": "fab-leafs-swassoc",
            "rn": "leaves-fab-leafs-swassoc-typ-range",
            "status": "created,modified"
          },
          "children": [
            {
              "fabricNodeBlk": {
                "attributes": {
                  "dn": "uni/fabric/leprof-fab-leaf-swProfile/leaves-fab-leafs-swassoc-typ-range/nodeblk-cd0b1ab33dcce538",
                  "from_": "101",
                  "to_": "104",
                  "name": "cd0b1ab33dcce538",
                  "rn": "nodeblk-cd0b1ab33dcce538",
                  "status": "created,modified"
                },
                "children": []
              },
              "fabricRsLeNodePGrp": {
                "attributes": {
                  "tDn": "uni/fabric/funcprof/lenodepgrp-fab-leaf-policyGroup",
                  "status": "created"
                },
                "children": []
              }
            }
          ]
        }
      }
    ]
  }
}
```

## Profiles



NAME	NODE	DESCRIPTION
<b>type: Spine</b>		
fab-spine-swProfile	201-202	ACI Bootcamp Lab Task 1.3.1 CREATE SPINE SWITCH P...
<b>type: leaf</b>		
fab-leaf-swProfile	101-104	ACI Bootcamp Lab Task 1.3.2 CREATE LEAF SWITCH PO...

## STEP 2: Configure FABRIC > FABRIC POLICIES to send CALLHOME messages to Callhome Destinations.

Fabric policies govern the operation of internal fabric interfaces. The system provides default fabric policies. Fabric policies enable configuring various functions or protocols. Administrators who have fabric administrator privileges can create new fabric policies according to their requirements. The APIC enables administrators to select the pods, leaf switches, and interfaces to which they will apply access policies.

Fabric policies configure interfaces that connect spine and leaf switches. Fabric policies can enable features such as monitoring (statistics collection and statistics export), troubleshooting (on-demand diagnostics and SPAN), or NTP.

The **call home source profile**, which determines the information to deliver and the urgency level for sending email-based alert notifications to a call home destination. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to generate a case with the Technical Assistance Center.

Fabric **Callhome** Sources need to be configured in the **DEFAULT** and **COMMON** monitoring policies configured in the **Fabric Policies** configuration. Use the GUI to configure the **DEFAULT** and **COMMON** Callhome monitoring policies. Use the API Inspector to capture the API POST information from this configuration.

**Task 2.1** Use the GUI to a configure the "**DEFAULT**" monitoring policy for Callhome. For this task, use the admin user "admin" and the password "Aci123bc".

- On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **MONITORING POLICIES**.
- Expand **default**
- Select "**Callhome/SNMP/Syslog**"
- In the "Callhome/SNMP/Syslog" Work Pane, **Select** the **SOURCE TYPE** "**CALLHOME**" from the Source Type drop down list.
- Click on the " + " sign to **CREATE CALLHOME SOURCE**. In the Create Callhome Source dialog box, perform the following actions:
  - Enter **Name** (*fab-callhome-source*)
  - For **Include** (*Check boxes for Events, Audit logs, and Faults*)
  - Select **Level** (*critical*)
  - Select **Destination Group** (*fab-callhome-destGrp*) **\*\* created earlier \*\***
  - Select **Query Group** (*fab-queryGroup*) **\*\* created earlier \*\***
  - Click **SUBMIT**

## Callhome/SNMP/Syslog

Monitoring Object: ALL

Source Type: Callhome

NAME	INCLUDE	LEVEL
No items have been found		

Define Callhome Source

Name: fab-callhome-source

Include:  Events  
 Audit logs  
 Faults

Level: information

Destination Group: fab-callhome-destGrp

Query Group: fab-queryGroup

Using the **APIC API Inspector**, this API Example was captured from the POST request to create Callhome Source for the **"DEFAULT"** monitoring policy for Callhome. You can use this APIC Example and use POSTMAN REST Client to create Callhome Source for the "DEFAULT" monitoring policy.

### API EXAMPLE

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/monfab-default/chsrc-fab-callhome-source.json>

payload

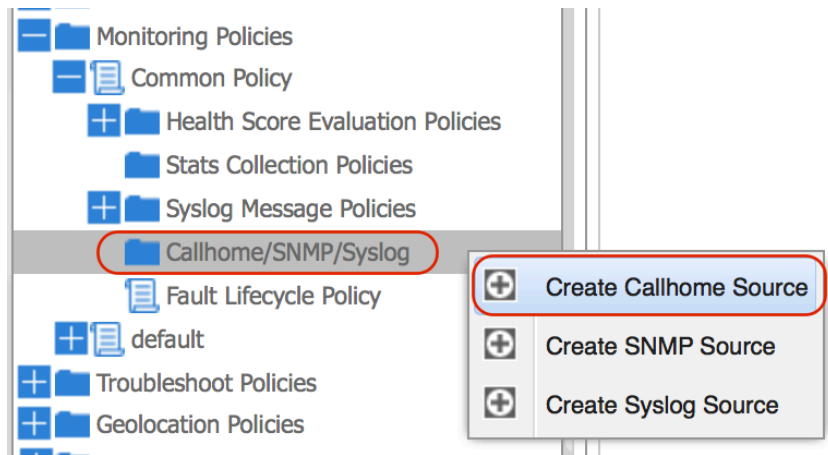
```
{"callhomeSrc":{"attributes":{"dn":"uni/fabric/monfab-default/chsrc-fab-callhome-source","name":"fab-callhome-source","incl":"events,audit,faults","urgency":"info","rn":"chsrc-fab-callhome-source","status":"created"},"children":[{"callhomeRsDestGroup":{"attributes":{"tDn":"uni/fabric/chgroup-fab-callhome-destGrp","status":"created"},"children":[]},"callhomeRsQueryGroupRel":{"attributes":{"tDn":"uni/fabric/chquerygroup-fab-queryGroup","status":"created"},"children":[]}}]}
```

# Callhome/SNMP/Syslog

Monitoring Object: ALL Source Type: Callhome

NAME	INCLUDE	LEVEL	DESTINATION GROUP	QUERY GROUP
fab-callhome-source	All Audit logs Events Faults	information	fab-callhome-destGrp	fab-queryGroup

- On the menu bar, choose **FABRIC > FABRIC POLICIES**. In the Navigation pane, expand **MONITORING POLICIES**.
- Expand **Common Policy**
- Select **"Callhome/SNMP/Syslog"**
- **Right Click** on "Callhome/SNMP/Syslog" and Select the **CREATE CALLHOME SOURCE**. In the Create Callhome Source dialog box, perform the following actions:
  - Enter **Name** ([fab-callhome-source](#))
  - For **Include** ([Check boxes for Events, Audit logs, and Faults](#))
  - Select **Level** ([critical](#))
  - Select **Destination Group** ([fab-callhome-destGrp](#)) *\*\* created earlier \*\**
  - Select **Query Group** ([fab-queryGroup](#)) *\*\* created earlier \*\**
  - Click **SUBMIT**



# CREATE CALLHOME SOURCE



Define Callhome Source

Name: fab-callhome-source

Include:  Events  
 Audit logs  
 Faults

Level: information

Destination Group: fab-callhome-destGrp

Query Group: fab-queryGroup

## Callhome/SNMP/Syslog

NAME	INCLUDE	LEVEL	DESTINATION GROUP	QUERY GROUP
fab-callhome-source	All Audit logs Events Faults	information	fab-callhome-destGrp	fab-queryGroup

Using the **APIC API Inspector**, this API Example was captured from the POST request to create Callhome Source for the "**COMMON**" monitoring policy for Callhome. You can use this APIC Example and use POSTMAN REST Client to create Callhome Source for the "COMMON" monitoring policy.

### API EXAMPLE

method: **POST**

url:

<https://192.168.1.141/api/node/mo/uni/fabric/moncommon/chsrc-fab-callhome-source.json>

payload

```
{ "callhomeSrc": { "attributes": { "dn": "uni/fabric/moncommon/chsrc-fab-callhome-source", "name": "fab-callhome-source", "incl": "events,audit,faults", "urgency": "info", "rn": "chsrc-fab-callhome-source", "status": "created" }, "children": [ { "callhomeRsDestGroup": { "attributes": { "tDn": "uni/fabric/chgroup-fab-callhome-destGrp", "status": "created" }, "children": [] }, { "callhomeRsQueryGroupRel": { "attributes": { "tDn": "uni/fabric/chquerygroup-fab-queryGroup", "status": "created" }, "children": [] } } ] } }
```

### STEP 3: Configure FABRIC > ACCESS POLICIES to send CALLHOME messages to Callhome Destinations.

Access policies govern the operation of interfaces that provide external access to the fabric. The system provides default access policies. Access policies enable configuring various functions or protocols. Administrators who have fabric administrator privileges can create new access policies according to their requirements. The APIC enables administrators to select the pods, leaf switches, and interfaces to which they will apply access policies.

The **call home source profile**, which determines the information to deliver and the urgency level for sending email-based alert notifications to a call home destination. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to generate a case with the Technical Assistance Center.

Fabric Callhome Sources need to be configured in the **DEFAULT** monitoring policies configured in the **Access Policies** configuration. Use the GUI to configure the **DEFAULT** Callhome monitoring policies. Use the API Inspector to capture the API POST information from this configuration.

Task 3.1 Use the GUI to a configure the "**DEFAULT**" monitoring policy for Callhome. For this task, use the admin user "admin" and the password "Aci123bc".

- On the menu bar, choose **FABRIC > ACCESS POLICIES**. In the Navigation pane, expand **MONITORING POLICIES**.
- Expand **default**
- Select "**Callhome/SNMP/Syslog**"
- In the "Callhome/SNMP/Syslog" Work Pane, **Select the SOURCE TYPE "CALLHOME"** from the Source Type drop down list.
- Click on the " + " sign to **CREATE CALLHOME SOURCE**. In the Create Callhome Source dialog box, perform the following actions:
  - Enter **Name** (*fab-callhome-source*)
  - For **Include** (*Check boxes for Events, Audit logs, and Faults*)
  - Select **Level** (*critical*)
  - Select **Destination Group** (*fab-callhome-destGrp*) **\*\* created earlier \*\***
  - Select **Query Group** (*fab-queryGroup*) **\*\* created earlier \*\***
  - Click **SUBMIT**

**Policies**

- Quick Start
- + Switch Policies
- + Module Policies
- + Interface Policies
- + Global Policies
- Monitoring Policies
  - default
  - Stats Collection Policies
  - Stats Export Policies
  - Diagnostics Policies
  - Callhome/SNMP/Syslog

## Callhome/SNMP/Syslog

Monitoring Object: ALL

Source Type: Callhome

NAME	INCLUDE	LEVEL

## CREATE CALLHOME SOURCE i

Define Callhome Source

Name: fab-callhome-source

Include:  Events  
 Audit logs  
 Faults





Level: information

Destination Group: fab-callhome-destGrp

Query Group: fab-queryGroup



# Callhome/SNMP/Syslog

 <b>Monitoring Object:</b> ALL 	<b>Source Type:</b> Callhome 			
 				
NAME	INCLUDE	LEVEL	DESTINATION GROUP	QUERY GROUP
fab-callhome-source	All Audit logs Events Faults	information	fab-callhome-destGrp	fab-queryGroup

Using the **APIC API** Inspector, this API Example was captured from the POST request to create Callhome Source for the "DEFAULT" monitoring policy for Callhome. You can use this APIC Example and use POSTMAN REST Client to create Callhome Source for the "DEFAULT" monitoring policy.

## API EXAMPLE

method: [POST](#)

url:

<https://192.168.1.141/api/node/mo/uni/infra/moninfra-default/chsrc-fab-callhome-source.json>

payload

```
{ "callhomeSrc": { "attributes": { "dn": "uni/infra/moninfra-default/chsrc-fab-callhome-source", "name": "fab-callhome-source", "incl": "events,audit,faults", "urgency": "info", "rn": "chsrc-fab-callhome-source", "status": "created" }, "children": [ { "callhomeRsDestGroup": { "attributes": { "tDn": "uni/fabric/chgroup-fab-callhome-destGrp", "status": "created" }, "children": [] }, { "callhomeRsQueryGroupRel": { "attributes": { "tDn": "uni/fabric/chquerygroup-fab-queryGroup", "status": "created" }, "children": [] } } ] } }
```

## 6 Verify Configuration of Call Home Feature for the ACI Fabric

For this lab section, you will verify the configuration of Callhome for Controllers and LeafSpine Switches in your designated ACI fabric. This section will provide references that may be helpful in verifying the configuration and application of the Callhome policies for Controllers and LeafSpine Switches in your designated ACI fabric.

This lab section you will verify Callhome configuration by:

- Check the various Callhome destinations configured in your designated ACI Fabric and verify that the configured destination email addresses are receiving Inventory Policy Callhome messages.
- Check the various Callhome destinations configured in your designated ACI Fabric and verify that the configured destination email addresses are receiving Fabric & Access Policy Callhome messages.


### Sample Emails for Each configured Callhome Destination

#### AML Format

```
Node 104 is inactive and not reachable.
aci-p1-server@aci.bootcamp.local
Sent: Saturday, January 10, 2015 at 2:15 AM
To: aci-bc-p5(mailer list)


<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope>
<xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session>
<xmlns:aml-session="http://www.cisco.com/2004/01/aml-session" soap-env:mus
envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-se
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>10D0::54B0D17D:
</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block>
<xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome</aml-block:Type>
<aml-block:CreationDate>Sat Jan 10 02:15:09 2015
```

## Short-Text Format

 **Node 104 is inactive and not reachable.**  
aci-p1-server@aci.bootcamp.local  
Sent: Saturday, January 10, 2015 at 2:15 AM  
To: aci-bc-p3(mailer list)

Created: 1420873987104  
Description: Node 104 is inactive and not reachable. topology/pod-1/node-104/fault-F1543  
Status: soaking  
Severity: critical

## XML Format

 **Node 104 is inactive and not reachable.**  
aci-p1-server@aci.bootcamp.local  
Sent: Saturday, January 10, 2015 at 2:15 AM  
To: aci-bc-p1(mailer list)

```
<?xml version="1.0" encoding="UTF-8"?>
<msgEnvelope>
<cnwRsActiveIf childAction="" dn="topology/pod-1/node-2/sys/caggr-[pol]/rsacti
10T03:13:06.361-04:00" parentSKey="unspecified" rType="mo" state="formed" state
1/node-2/sys/cphys-[eth2/1]" tSKey="eth2/1" tType="mo"/>
<cnwRsMbrIfs childAction="" dn="topology/pod-1/node-2/sys/caggr-[pol]/rsmbrIfs-
lcOwn="local" modTs="2015-01-10T03:13:06.361-04:00" parentSKey="unspecified" r
tCl="cnwPhysIf" tDn="topology/pod-1/node-2/sys/cphys-[eth2/2]" tSKey="eth2/2" t
<cnwRsMbrIfs childAction="" dn="topology/pod-1/node-2/sys/caggr-[pol]/rsmbrIfs-
lcOwn="local" modTs="2015-01-10T03:13:06.361-04:00" parentSKey="unspecified" r
tCl="cnwPhysIf" tDn="topology/pod-1/node-2/sys/cphys-[eth2/1]" tSKey="eth2/1" t
<cnwAggrIf adminSt="up" autoNeg="on" bw="0" childAction="" delay="1" descr="" c
dot1qEtherType="0x8100" id="pol" inhBw="unspecified" layer="Layer3" lcOwn="loc
medium="broadcast" modTs="2015-01-08T11:36:03.469-04:00" mode="trunk" monPolDn
operSt="up" portT="unknown" routerMac="90:E2:BA:4E:0A:38" snmpTrapSt="enable" s
switchingSt="disabled" trunkLog="default" usage="discovery"/>
<cnwRsActiveIf childAction="" dn="topology/pod-1/node-2/sys/caggr-[pol.1]/rsact
10T03:13:06.361-04:00" parentSKey="unspecified" rType="mo" state="formed" state
1/node-2/sys/cphys-[eth1/1]" tSKey="eth1/1" tType="mo"/>
<cnwRsMbrIfs childAction="" dn="topology/pod-1/node-2/sys/caggr-[pol.1]/rsmbrIf
forceResolve="no" lcOwn="local" modTs="2015-01-10T03:13:06.361-04:00" parentSKe
```

## Sample Callhome Inventory Message

```
Inventory message
aci-p1-server@aci.bootcamp.local
Sent: Friday, January 9, 2015 at 9:02 PM
To: aci-bc-p1(mailer list)
<leqptLooseNode childAction="" id="10.122.254.82" lcOwn="local" modTs="2015-01-07T13:38:50.086-04:0
Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2014, Cisco Systems, Inc. All rights reserved." sysName="tsi-rtp-infra-n5k-1" ui
<leqptLooseNode childAction="" id="10.122.254.55" lcOwn="local" modTs="2015-01-08T16:38:39.622-04:0
Operating System (NX-OS) Software 6.0(2)N2(5)
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2014, Cisco Systems, Inc. All rights reserved." sysName="aci-n5k-55" uid="0"/>
<leqptLooseNode childAction="" id="10.122.254.218" lcOwn="local" modTs="2015-01-07T13:38:50.146-04:0
Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2014, Cisco Systems, Inc. All rights reserved." sysName="tsi-bootcamp-A" uid="0"
<leqptLooseNode childAction="" id="172.18.217.32" lcOwn="local" modTs="2015-01-07T13:38:50.106-04:0
n7000, Software (n7000-s1-dk9), Version 6.2(8a), RELEASE SOFTWARE Copyright (c) 2002-2013 by Cisco
<leqptLooseNode childAction="" id="10.122.254.45" lcOwn="local" modTs="2015-01-08T16:37:36.083-04:0
Operating System (NX-OS) Software 6.0(2)N2(1)
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2013, Cisco Systems, Inc. All rights reserved." sysName="aci-n5k-45" uid="0"/>
<leqptLooseNode childAction="" id="10.122.254.64" lcOwn="local" modTs="2015-01-08T16:39:18.509-04:0
Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2012, Cisco Systems, Inc. All rights reserved." sysName="aci-n5k-64" uid="0"/>
```

- Additional resources that can be used to verify Callhome Policy Configuration.

### CLI Commands (For APIC)

- `show external-data-collectors query-groups`
- `cat /aci/admin/external-data-collectors/callhome-query-groups/fab-queryGroup/summary`
- `show external-data-collectors monitoring-destinations`
- `cat /aci/admin/external-data-collectors/monitoring-destinations/callhome/summary`
- `show fabric policies callhome-inventory`

### Visore

- `callhomeGroup`
- `callhomeDest`
- `callhomeQueryGroup`
- `callhomeQuery`
- `callhomeRtQueryGroupRel`
- `callhomeInvP`
- `callhomeRsDestGroupRel`
- `callhomeRsInvScheduler`
- `callhomeRtCallhomeInvPol`

## REST API

- [/api/node/class/callhomeGroup.xml?](#)
- [/api/node/class/callhomeDest.xml?](#)
- [/api/node/class/callhomeQueryGroup.xml?](#)
- [/api/node/mo/uni/fabric/chquerygroup-fab-queryGroup.xml?query-target=children](#)
- [/api/node/class/callhomeInvP.xml?](#)
- [/api/node/mo/uni/fabric/chinvp-fab-callhome-inventory.xml?query-target=children](#)

## Logs

/var/log/dme/log

- [svc\\_ifc\\_eventmgr.bin.log](#)
- [svc\\_ifc\\_policymgr.bin.log](#)







## Reference Material:







- **Cisco APIC Faults, Events, and System Messages Management Guide**  
[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b APIC Faults Errors.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors.html)
- **Cisco ACI System Messages Reference Guide**  
[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci\\_syslog/About.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci_syslog/About.html)
- **ACI System Messages**  
[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci\\_syslog/ACI\\_SysMsg.pdf](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci_syslog/ACI_SysMsg.pdf)

## VISORE Screenshots from Lab Configuration

<u>callhomeGroup</u>	
childAction	
descr	ACI Bootcamp Lab for CALLHOME
dn	<a href="#">uni/fabric/chgroup-fab-callhome-destGrp</a> < >     ! H
lcOwn	local
modTs	2015-01-09T17:16:51.255-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >     ! H
name	fab-callhome-destGrp
status	
uid	15374

<u>callhomeDest</u>	
adminState	enabled
childAction	
descr	
dn	<a href="#">uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P1-Server</a> < >     ! H
email	aci-bc-p1@cisco.com
format	xml
lcOwn	local
maxSize	1000000
modTs	2015-01-09T17:16:51.255-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >     ! H
name	ACI-P1-Server
status	
uid	15374
urgency	info

<a href="#">callhomeDest</a>	
adminState	enabled
childAction	
descr	
dn	<a href="#">uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P5-Server</a> < >   
email	aci-bc-p5@cisco.com
format	aml
lcOwn	local
maxSize	1000000
modTs	2015-01-09T17:16:51.255-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >   
name	ACI-P5-Server
status	
uid	15374
urgency	info

<a href="#">callhomeDest</a>	
adminState	enabled
childAction	
descr	
dn	<a href="#">uni/fabric/chgroup-fab-callhome-destGrp/dest-ACI-P3-Server</a> < >   
email	aci-bc-p3@cisco.com
format	short-txt
lcOwn	local
maxSize	1000000
modTs	2015-01-09T17:16:51.255-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >   
name	ACI-P3-Server
status	
uid	15374
urgency	info

<b><u>callhomeQueryGroup</u></b>	
childAction	
dn	<a href="#">uni/fabric/chquerygroup-fab-queryGroup</a> < >     ! H
lcOwn	local
modTs	2015-01-09T18:13:08.346-04:00
name	fab-queryGroup
status	
uid	15374

<b><u>callhomeQuery</u></b>	
childAction	
dn	<a href="#">uni/fabric/chquerygroup-fab-queryGroup/chquery-fab-query-1</a> < >     ! H
entity	
lcOwn	local
modTs	2015-01-09T18:13:08.346-04:00
name	fab-query-1
rspSubtree	full
rspSubtreeInclude	add-mo-list,audit-logs,config-only,count,deployment,event-logs,fault-count,fault-records,faults,health,health-records,local-prefix,no-scoped,port-deployment,record-subtree,relations,relations-with-parent,required,state,stats,tasks
status	
target	subtree
type	dn
uid	15374



callhomeRtQueryGroupRel	
childAction	
dn	<a href="#">uni/fabric/chquerygroup-fab-queryGroup/rtqueryGroupRel-[uni/fabric/monfab-default/chsrc-fab-callhome-source]</a> < >   
lcOwn	local
modTs	2015-01-09T22:38:22.875-04:00
status	
tCl	callhomeSrc
tDn	<a href="#">uni/fabric/monfab-default/chsrc-fab-callhome-source</a> < >   
callhomeRtQueryGroupRel	
childAction	
dn	<a href="#">uni/fabric/chquerygroup-fab-queryGroup/rtqueryGroupRel-[uni/fabric/moncommon/chsrc-fab-callhome-source]</a> < >   
lcOwn	local
modTs	2015-01-09T22:54:30.127-04:00
status	
tCl	callhomeSrc
tDn	<a href="#">uni/fabric/moncommon/chsrc-fab-callhome-source</a> < >   
callhomeRtQueryGroupRel	
childAction	
dn	<a href="#">uni/fabric/chquerygroup-fab-queryGroup/rtqueryGroupRel-[uni/infra/moninfra-default/chsrc-fab-callhome-source]</a> < >   
lcOwn	local
modTs	2015-01-09T23:05:27.680-04:00
status	
tCl	callhomeSrc
tDn	<a href="#">uni/infra/moninfra-default/chsrc-fab-callhome-source</a> < >   

<a href="#">callhomeInvP</a>	
adminSt	untriggered
adminState	enabled
childAction	
descr	
dn	<a href="#">uni/fabric/chinvp-fab-callhome-inventory</a> < >     ! H
lastAttempt	2015-01-10T02:00:28.015-04:00
lastSuccess	2015-01-10T02:00:28.045-04:00
lcOwn	local
maximumRetryCount	1
modTs	2015-01-10T01:59:10.595-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >     ! H
name	fab-callhome-inventory
retryCount	0
seqNum	0
status	
triggerTime	1969-12-31T20:00:00.000-04:00
uid	15374

<a href="#">callhomeRsDestGroupRel</a>	
childAction	
dn	<a href="#">uni/fabric/chinvp-fab-callhome-inventory/rsdestGroupRel</a> < >     ! H
forceResolve	no
lcOwn	local
modTs	2015-01-09T20:30:00.137-04:00
monPolDn	<a href="#">uni/fabric/monfab-default</a> < >     ! H
rType	mo
state	formed
stateQual	none
status	
tCl	callhomeGroup
tDn	<a href="#">uni/fabric/chgroup-fab-callhome-destGrp</a> < >     ! H
tType	mo
uid	15374

## Object Mode References for Callhome Feature

### Class callhome:Group (CONCRETE)

Class ID:1700  
 Class Label: Callhome Destination Group  
 Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
 Write Access: [admin]  
 Read Access: [admin, ops]  
 Semantic Scope: Fabric  
 Semantic Scope Evaluation Rule: Parent  
 Monitoring Policy Source: Parent  
 Monitoring Flags : [ IsObservable: true, HasStats: false, HasFaults: false, HasHealth: true ]

A Call Home group, which brings together all the information needed to send Call Home messages to a set of destinations.

### Class callhome:Dest (CONCRETE)

Class ID:1699  
 Class Label: Callhome Destination  
 Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
 Write Access: [admin]  
 Read Access: [admin, ops]  
 Semantic Scope: Fabric  
 Semantic Scope Evaluation Rule: Parent  
 Monitoring Policy Source: Parent  
 Monitoring Flags : [ IsObservable: true, HasStats: false, HasFaults: false, HasHealth: true ]

The Call Home destination profile, which contains the delivery information for receiving email-based alert notifications of critical system policies. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to generate a case with the Technical Assistance Center.

## Class callhome:QueryGroup (CONCRETE)

Class ID:1709  
Class Label: Callhome Query Group  
Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
Write Access: [admin]  
Read Access: [admin, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: false, HasStats: false, HasFaults: false, HasHealth: false ]

A Call Home query group, which is a set of Call Home query messages to be sent on returned objects.

## Class callhome:Query (CONCRETE)

Class ID:1708  
Class Label: Query  
Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
Write Access: [admin]  
Read Access: [admin, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: false, HasStats: false, HasFaults: false, HasHealth: false ]

A query. This is a query object representing all information on returned objects.

## Class callhome:RtQueryGroupRel (CONCRETE)

Class ID:1698  
Class Label: Callhome Source  
Encrypted: false - Exportable: false - Persistent: true - Configurable: false  
Relationship Type: explicit  
Relationship Cardinality: n-to-1  
Relationship From: [callhome:Src](#)  
Relationship From Rel: [callhome:RsQueryGroupRel](#)  
Relationship To: [callhome:QueryGroup](#)  
Relationship To Rel: [callhome:RtQueryGroupRel](#)  
Enforceable: true  
Resolvable: true  
Write Access: [NON CONFIGURABLE]  
Read Access: [admin, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: false, HasStats: false, HasFaults: false, HasHealth: false ]

A target relation to a call home query group.

## Class callhome:InvP (CONCRETE)

Class ID:1702  
Class Label: Callhome Inventory Policy  
Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
Write Access: [admin]  
Read Access: [admin, fabric-connectivity-I1, fabric-connectivity-I2, fabric-connectivity-I3, fabric-equipment, fabric-protocol-I1, fabric-protocol-I2, fabric-protocol-I3, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: true, HasStats: false, HasFaults: true, HasHealth: true ]

The call home inventory policy, which is used to send Call Home messages with system inventory information.

## Class callhome:RsDestGroupRel (CONCRETE)

Class ID:1705  
Class Label: Relation to Callhome Destination Group  
Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
Relationship Type: explicit  
Relationship Cardinality: n-to-1  
Relationship From: [callhome:InvP](#)  
Relationship From Rel: [callhome:RsDestGroupRel](#)  
Relationship To: [callhome:Group](#)  
Relationship To Rel: [callhome:RtDestGroupRel](#)  
Enforceable: true  
Resolvable: true  
Write Access: [admin]  
Read Access: [admin, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: true, HasStats: false, HasFaults: true, HasHealth: true ]

A source relation to a call home group.

## Class callhome:RslnvScheduler (CONCRETE)

Class ID:1703  
Class Label: Relation to Callhome Inventory Scheduler  
Encrypted: false - Exportable: true - Persistent: true - Configurable: true  
Relationship Type: named  
Relationship Cardinality: 1-to-1  
Relationship From: [callhome:InvP](#)  
Relationship From Rel: [callhome:RslnvScheduler](#)  
Relationship To: [trig:SchedP](#)  
Relationship To Rel: [trig:RtlnvScheduler](#)  
Enforceable: true  
Resolvable: true  
Write Access: [admin]  
Read Access: [admin, ops]  
Semantic Scope: Fabric  
Semantic Scope Evaluation Rule: Parent  
Monitoring Policy Source: Parent  
Monitoring Flags : [ IsObservable: true, HasStats: false, HasFaults: true, HasHealth: true ]

A source relation to the scheduler policy.

## Class callhome:RtCallhomeInvPol (CONCRETE)

Class ID:924

Encrypted: false - Exportable: false - Persistent: true - Configurable: false

Relationship Type: named

Relationship Cardinality: n-to-1

Relationship From: [fabric:ANodePGrp](#)

Relationship From Rel: [fabric:RsCallhomeInvPol](#)

Relationship To: [callhome:InvP](#)

Relationship To Rel: [callhome:RtCallhomeInvPol](#)

Enforceable: true

Resolvable: true

Write Access: [NON CONFIGURABLE]

Read Access: [admin, fabric-equipment, ops]

Semantic Scope: Fabric

Semantic Scope Evaluation Rule: Parent

Monitoring Policy Source: Parent

Monitoring Flags : [ IsObservable: false, HasStats: false, HasFaults: false, HasHealth: false ]

A target relation to the call home inventory policy.

**End of Document**