

Smart Call Home

Smart Call Home provides an email-based notification for critical system policies. A range of message formats are available for compatibility with pager services, standard email, 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 automatically generate a case with the Technical Assistance Center.

- Overview, on page 1
- Guidelines and Limitations, on page 8
- Configuration, on page 9
- Configure Contact Information, on page 12
- Create Destination Profile, on page 13
- Modify Destination Profile, on page 15
- Associate Alert Group with Destination Profile, on page 17
- Add Show Commands to Alert Group, on page 18
- Configure Email Server, on page 19
- Configure VRFs to Send Messages Using HTTP, on page 20
- Configure HTTP Proxy Server, on page 22
- Configure Periodic Inventory Notifications, on page 23
- Disable Duplicate Message Throttling, on page 24
- Enable or Disable Smart Call Home, on page 25
- Configure SMTP-AUTH for Call Home Mail Transfer, on page 26
- Configure Source Interface, on page 28
- Test Smart Call Home Configuration, on page 29
- Additional Reference Information, on page 30

Overview

- Smart Call Home automatically executes relevant CLI commands and attaches their output.
- It provides multiple message format options, including Short Text, Full Text, and XML.
- It supports multiple concurrent message destinations, allowing configuration of up to 50 email addresses per destination profile.

Features

If you have a service contract directly with Cisco, you can register for the Smart Call Home service. Smart Call Home analyzes Smart Call Home messages and provides background information and recommendations. For known issues, particularly online diagnostics failures, Automatic Service Requests are generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostic alerts.
- Analysis of Smart Call Home messages and, if needed, Automatic Service Request generation, routed to the correct TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device, through an HTTP proxy server, or a downloadable Transport Gateway (TG). You can use a TG aggregation point to support multiple devices or in cases where security dictates that your devices may not be connected directly to the Internet.
- Web-based access to Smart Call Home messages and recommendations, inventory, and configuration information for all Smart Call Home devices. This feature provides access to associated field notices, security advisories, and end-of-life information.
- Automatic execution and attachment of relevant CLI command output.
- Multiple message format options:
  - Short Text—Suitable for pagers or printed reports.
  - Full Text—Fully formatted message information suitable for human reading.
  - 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. The XML format enables communication with the Technical Assistance Center.
- Multiple concurrent message destinations. You can configure up to 50 email destination addresses for each destination profile.
- High Availability - Both stateful and stateless restarts are supported for Smart Call Home.
- Virtualization Support - One instance of Smart Call Home is supported. You can register your contact information at the Smart Call Home web site at the following URL: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome. You can test Smart Call Home using the **callhome send** and **callhome test** commands. Smart Call Home is virtual routing and forwarding (VRF) aware. You can configure Smart Call Home to use a particular VRF to reach the Smart Call Home SMTP server.

You need the following information to register:

- The SMARTnet contract number for your device
- Your email address
- Your Cisco.com ID

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Smart Call Home message if the alert occurs.
- One or more email destinations—The list of recipients for the Smart Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Smart Call Home message (short text, full text, or XML).
- Message severity level—The Smart Call Home severity level that the alert must meet before Cisco NX-OS generates a Smart Call Home message to all email addresses in the destination profile. Cisco NX-OS does not generate an alert if the Smart Call Home severity level of the alert is lower than the message severity level set for the destination profile.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group that will send out periodic messages daily, weekly, or monthly.

Cisco NX-OS supports the following predefined destination profiles:

- CiscoTAC-1—Supports the Cisco-TAC alert group in XML message format. This profile is preconfigured with the callhome@cisco.com email contact, maximum message size, and message severity level 0. You can change any of the default information for this profile.
- full-text-destination—Supports the full text message format.
- short-text-destination—Supports the short text message format.

Default Settings

This table lists the default settings for Smart Call Home parameters.

Table 1: Default Smart Call Home Parameters

Parameters	Default
Destination message size for a message sent in full text format	2,500,000
Destination message size for a message sent in XML format	2,500,000
Destination message size for a message sent in short text format	4000
SMTP server port number if no port is specified	25
SMTP server priority if no priority is specified	50
Alert group association with profile	All for full-text-destination and short-text-destination profiles. The cisco-tac alert group for the CiscoTAC-1 destination profile.
Format type	XML
Smart Call Home message level	0 (zero)
HTTP proxy server use	Disabled and no proxy server configured

Parameters	Default
Periodic inventory notification (**periodic-inventory notification**)	Enabled by default

Alert Groups

An alert group is a predefined subset of Smart Call Home alerts that are supported in all Cisco Nexus devices. Alert groups allow you to select the set of Smart Call Home alerts that you want to send to a predefined or custom destination profile. The device sends Smart Call Home alerts to email destinations in a destination profile only if that Smart Call Home alert belongs to one of the alert groups associated with that destination profile and if the alert has a Smart Call Home message severity at or above the message severity set in the destination profile.

The following table lists the supported alert groups and the default CLI command output included in Smart Call Home messages generated for the alert group.

Table 2: Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Cisco-TAC	All critical alerts from the other alert groups destined for Smart Call Home.	Execute commands based on the alert group that originates the alert.
Configuration	Periodic events related to configuration.	**show module** **show version**
Diagnostic	Events generated by diagnostics.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**

Alert Group	Description	Executed Commands
EEM	Events generated by EEM.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show module** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	**show environment** **show logging last 200** **show module** **show version**
Inventory	Inventory status that is provided whenever a unit is cold booted or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	**show inventory** **show license usage** **show module** **show sprom all** **show system uptime** **show version**
License	Events related to licensing and license violations.	**show logging last 200**

Alert Group	Description	Executed Commands
Linecard hardware	Events related to standard or intelligent switching modules.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Supervisor hardware	Events related to supervisor modules.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Syslog port group	Events generated by the syslog PORT facility.	**show license usage** **show logging last 200**

Alert Group	Description	Executed Commands
System	Events generated by failure of a software system that is critical to unit operation.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Test	User-generated test message.	**show module** **show version**

Smart Call Home maps the syslog severity level to the corresponding Smart Call Home severity level for syslog port group messages.

You can customize predefined alert groups to execute additional CLI **show** commands when specific events occur and send that **show** output with the Smart Call Home message.

You can add **show** commands only to full text and XML destination profiles. Short text destination profiles do not support additional **show** commands because they only allow 128 bytes of text.

Message Levels

Smart Call Home allows you to filter messages based on their level of urgency. You can associate each predefined or user-defined destination profile with a Smart Call Home threshold from 0 (least urgent) to 9 (most urgent). The default is 0 (all messages are sent).

Syslog severity levels are mapped to the Smart Call Home message level.



Note Smart Call Home and Syslogs use different severity levels (see the following table). Smart Call Home does not change the syslog message level in the message text.

The following table lists each Smart Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 3: Severity and Syslog Level Mapping

Smart Call Home Level	Keyword	Syslog Level	Description
9	Catastrophic	N/A	Network-wide catastrophic failure.

Smart Call Home Level	Keyword	Syslog Level	Description
8	Disaster	N/A	Significant network impact.
7	Fatal	Emergency (0)	System is unusable.
6	Critical	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	Major	Critical (2)	Major conditions.
4	Minor	Error (3)	Minor conditions.
3	Warning	Warning (4)	Warning conditions.
2	Notification	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	Information (6)	Normal event signifying return to normal state.
0	Debugging	Debug (7)	Debugging messages.

Guidelines and Limitations

Smart Call Home has the following configuration guidelines and limitations:

- If there is no IP connectivity or if the interface in the virtual routing and forwarding (VRF) instance to the profile destination is down, the device cannot send Smart Call Home messages.
- Smart Call Home operates with any SMTP server.
- You can configure up to five SMTP servers for Smart Call Home.
- Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.
- When configuring Smart Call Home commands such as street address, customer ID, and site ID, you must configure each one of these commands as individual command instead of grouping them with semi-colon separator.
- Callhome does not support specifying a source interface using the **ip http source-interface** command.
- Callhome does not support specifying a source interface using the **ip http source-interface** command.

Prerequisites

Smart Call Home has the following prerequisites:

- To send messages to an email address, you must first configure an email server. To send messages using HTTP, you must have access to an HTTPS server and have a valid certificate installed on the Cisco Nexus device.
- Your device must have IP connectivity to an email server or HTTPS server.

- You must first configure the contact name (SNMP server contact), phone, and street address information. This step is required to determine the origin of messages received.
- If you use Smart Call Home, you need an active service contract for the device that you are configuring.

Release Specific Limitations

Beginning with Cisco NX-OS Release 10.2(3)F

- SMTP-AUTH is supported for secure call home mail transfer on Cisco Nexus 9000 Series platform switches.

Beginning with Cisco NX-OS Release 10.3(2)F

- can configure the **source-interface** command under callhome to specify a source interface.

Database Merge Guidelines

When you merge two Smart Call Home databases, the following guidelines apply:

- The merged database contains the following information:
  - A superset of all the destination profiles from the merging devices.
  - The destination profile email addresses and alert groups.
  - Other configuration information (for example, message throttling, or periodic inventory) present in the managing device.
- Destination profile names cannot be duplicated within the merging devices—even though the configurations are different, the names cannot be duplicated. If a profile name is duplicated, one of the duplicate profiles must first be deleted or the merger fails.

Configuration

The configuration process involves completing the Smart Call Home configuration procedures in a recommended sequence.

- Follow the recommended order for Smart Call Home configuration steps.
- Some steps are optional and may be performed as needed.
- Cisco NX-OS commands may differ from Cisco IOS commands.

Configuration Sequence

We recommend that you complete the Smart Call Home configuration procedures in the following sequence:

1. Configure Contact Information, on page 12
2. Create Destination Profile, on page 13
3. Associate Alert Group with Destination Profile, on page 17

4. Add Show Commands to Alert Group, on page 18 (Optional)
5. Enable or Disable Smart Call Home, on page 25
6. Configure Source Interface, on page 28 (Optional)
7. Test Smart Call Home Configuration, on page 29 (Optional)

Note Be aware that the Cisco NX-OS commands may differ from the Cisco IOS commands.

Verify Configuration

To display Smart Call Home configuration information, perform one of the following tasks:

Command	Purpose
show callhome	Displays the Smart Call Home configuration.
show callhome destination-profile *name*	Displays one or more Smart Call Home destination profiles.
show callhome transport	Displays the transport-related configuration for Smart Call Home.
show callhome user-def-cmds	Displays CLI commands added to any alert groups.
show running-config callhome [**all**]	Displays the running configuration for Smart Call Home.
show startup-config callhome	Displays the startup configuration for Smart Call Home.
show tech-support callhome	Displays the technical support output for Smart Call Home.

Configuration Examples

This example shows how to create a destination profile called Noc101, associate the Configuration alert group to that profile, configure contact and email information, and specify the VRF used to send Smart Call Home messages over HTTP:

```
configure terminal
snmp-server contact person@company.com
callhome
distribute
email-contact admin@Mycompany.com
phone-contact +1-800-123-4567
streetaddress 123 Anystreet st. Anytown,AnyWhere
destination-profile Noc101 format full-txt
destination-profile full-text-destination email-addr person@company.com
destination-profile full-text-destination message-level 5
destination-profile Noc101 alert-group Configuration
alert-group Configuration user-def-cmd show ip route
transport email mail-server 192.0.2.10 priority 1
transport http use-vrf Blue
enable
commit
```

This example shows how to configure multiple SMTP servers for Smart Call Home messages:

```
configure terminal
callhome
transport email mail-server 192.0.2.10 priority 4
transport email mail-server 172.21.34.193
transport email smtp-server 10.1.1.174
transport email mail-server 64.72.101.213 priority 60
transport email from person@company.com
transport email reply-to person@company.com
commit
```

Note Configuration of multiple smtp-servers for authentication purpose using the **callhome email mail-server** command is not supported.

Based on the configuration above, the SMTP servers would be tried in this order:

10.1.1.174 (priority 0)

192.0.2.10 (priority 4)

172.21.34.193 (priority 50, which is the default)

64.72.101.213 (priority 60)

Note The **transport email smtp-server** command has a priority of 0, which is the highest. The server specified by this command is tried first followed by the servers specified by the **transport email mail-server** commands in order of priority.

This example shows how to configure Smart Call Home to send HTTP messages through an HTTP proxy server:

```
configure terminal
callhome
transport http proxy server 10.10.10.1 port 4
transport http proxy enable
commit
```

This example shows how to configure SMTP-AUTH servers for call home mail transfer:

```
callhome
email-contact admin@Mycompany.com
destination-profile testProfile-1
destination-profile testProfile-1 format XML
destination-profile testProfile-1 index 1 email-addr person@company.com
destination-profile testProfile-1 alert-group all
destination-profile full_txt alert-group test
transport email from callhome_person@company.com
transport email smtp-server 10.1.1.174 port 587 use-vrf management
transport email username user1 passwd Y2FsbGhvbWUK encrypted
transport http use-vrf management
enable
```

Configure Contact Information

You must configure the email, phone, and street address information for Smart Call Home. You can optionally configure the contract ID, customer ID, site ID, and switch priority information.

You must configure each one of these Smart Call Home commands as individual command instead of grouping them with semi-colon separator.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the SNMP sysContact using the command **snmp-server contact** *sys-contact*

Example:

```
switch(config)# snmp-server contact personname@companyname.com
```

Step 3 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 4 Configure the email address for the person primarily responsible for the device using the command **email-contact** *email-address*

Example:

```
switch(config-callhome)# email-contact admin@Mycompany.com
```

The *email-address* can be up to 255 alphanumeric characters in email address format.



Note You can use any valid email address. The address cannot contain spaces.

Step 5 Configure the phone number in international phone number format for the person primarily responsible for the device using the command **phone-contact** *international-phone-number*

Example:

```
switch(config-callhome)# phone-contact +1-800-123-4567
```

The *international-phone-number* can be up to 17 alphanumeric characters and must be in international phone number format.



Note The phone number cannot contain spaces. Use the plus (+) prefix before the number.

Step 6 Configure the street address as an alphanumeric string with white spaces for the person primarily responsible for the device using the command **streetaddress** *address*

Example:

```
switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere
```

The *address* can be up to 255 alphanumeric characters. Spaces are accepted.

Step 7 Configure the contract number for this device from the service agreement using the command **contract-id** *contract-number* and configure the customer number for this device from the service agreement using the command **customer-id** *customer-number*

Example:

```
switch(config-callhome)# contract-id Contract5678
```

Example:

```
switch(config-callhome)# customer-id Customer123456
```

This is an optional step.

The *contract-number* can be up to 255 alphanumeric characters in free format. The *customer-number* can be up to 255 alphanumeric characters in free format.

Step 8 Configure the site number for this device using the command **site-id** *site-number* and configure the switch priority for this device using the command **switch-priority** *number* .

Example:

```
switch(config-callhome)# site-id Site1
```

Example:

```
switch(config-callhome)# switch-priority 3
```

This is an optional step.

The *site-number* can be up to 255 alphanumeric characters in free format. The range is from 0 to 7, with 0 being the highest priority and 7 the lowest. The default is 7.

Step 9 Commit the Smart Call Home configuration commands using the command **commit** and display a summary of the Smart Call Home configuration using the command **show callhome**

Example:

```
switch(config-callhome)# commit
```

Example:

```
switch(config-callhome)# show callhome
```

What to do next

Copy the running configuration to the startup configuration using the command **copy running-config startup-config** and create a destination profile.

Create Destination Profile

You can create a user-defined destination profile and configure its message format.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Create a new destination profile using the command **destination-profile** *name*

Example:

```
switch(config-callhome)# destination-profile Noc101
```

The name can be any alphanumeric string up to 31 characters.

Step 4 Set the message format for the profile using the command **destination-profile** *name* **format** { **XML** | **full-txt** | **short-txt** }

Example:

```
switch(config-callhome)# destination-profile Noc101 format full-txt
```

The name can be any alphanumeric string up to 31 characters.

Step 5 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 6 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*]

Example:

```
switch(config-callhome)# show callhome destination-profile profile Noc101
```

This is an optional step.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Associate one or more alert groups with a destination profile.

Modify Destination Profile

You can modify the following attributes for a predefined or user-defined destination profile:

- Destination email address—The actual address, pertinent to the transport mechanism, to which the alert should be sent.
- Destination URL—The HTTP or HTTPS URL that defines where alerts should be sent.
- Transport method—The email or HTTP transport that determines which type of destination addresses are used.
- Message formatting—The message format used for sending the alert (full text, short text, or XML).
- Message level—The Smart Call Home message severity level for this destination profile.
- Message size—The allowed length of a Smart Call Home message sent to the email addresses in this destination profile.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure an email address for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **email-addr** *address*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination email-addr person@place.com
```

You can configure up to 50 email addresses in a destination profile.

Step 4 Configure an HTTP or HTTPS URL for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **http** *address*

Example:

```
switch(config-callhome)# destination-profile CiscoTAC-1 http
https://tools.cisco.com/its/service/oddce/services/DDCEService
```

The URL can be up to 255 characters.

Step 5 Configure an email or HTTP transport method for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **transport-method** { **email** | **http** }

Example:

```
switch(config-callhome)# destination-profile CiscoTAC-1 transport-method http
```

The type of transport method that you choose determines the configured destination addresses of that type.

Step 6 Configure the Smart Call Home message severity level for this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **message-level** *number*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination message-level 5
```

Cisco NX-OS sends only alerts that have a matching or higher Smart Call Home severity level to destinations in this profile. The range is from 0 to 9, where 9 is the highest severity level.

Step 7 Configure the maximum message size for this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **message-size** *number*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination message-size 100000
```

The range is from 0 to 5000000. The default is 2500000.

Step 8 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 9 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*] and copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-callhome)# show callhome destination-profile profile full-text-destination
```

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Associate one or more alert groups with a destination profile.

Associate Alert Group with Destination Profile

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Associate an alert group with this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **alert-group** { **All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test** }

Example:

```
switch(config-callhome)# destination-profile Noc101 alert-group All
```

Use the **All** keyword to associate all alert groups with the destination profile.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*]

Example:

```
switch(config-callhome)# show callhome destination-profile profile Noc101
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally add **show** commands to an alert group and then configure the SMTP email server.

Add Show Commands to Alert Group

You can assign a maximum of five user-defined CLI **show** commands to an alert group.

Note You cannot add user-defined CLI **show** commands to the CiscoTAC-1 destination profile.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Add the **show** command output to any Smart Call Home messages sent for this alert group using the command **alert-group** { **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test** } **user-def-cmd** *show-cmd*

Example:

```
switch(config-callhome)# alert-group Configuration user-def-cmd show ip route
```

Only valid **show** commands are accepted.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about all user-defined **show** commands added to alert groups using the command **show callhome user-def-cmds**

Example:

```
switch(config-callhome)# show callhome user-def-cmds
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Configure Smart Call Home to connect to the SMTP email server.

Configure Email Server

You must configure the SMTP server address for the Smart Call Home functionality to work. You can also configure the from and reply-to email addresses.

You can configure up to five SMTP servers for Smart Call Home. The servers are tried based on their priority. The highest priority server is tried first. If the message fails to be sent, the next server in the list is tried until the limit is exhausted. If two servers have equal priority, the one that was configured earlier is tried first.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Configure the SMTP server as the domain name server (DNS) name, IPv4 address, or IPv6 address using the command **transport email mail-server** *ip-address* [**port** *number*] [**priority** *number*] [**use-vrf** *vrf-name*]

Example:

```
switch(config-callhome)# transport email mail-server 192.0.2.1 use-vrf Red
```

Optionally configures the port number. The port range is from 1 to 65535. The default port number is 25.

Also optionally configures the priority of the SMTP server. The priority range is from 1 to 100, with 1 being the highest priority and 100 the lowest. If you do not specify a priority, the default value of 50 is used.

Also optionally configures the VRF to use when communicating with this SMTP server. The VRF specified is not used to send messages using HTTP.

Step 4 Configure the email from field for Smart Call Home messages using the command **transport email from** *email-address*

Example:

```
switch(config-callhome)# transport email from person@company.com
```

This is an optional step.

Step 5 Configure the email reply-to field for Smart Call Home messages using the command **transport email reply-to** *email-address*

Example:

```
switch(config-callhome)# transport email reply-to person@company.com
```

This is an optional step.

Step 6 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 7 Display the transport-related configuration for Smart Call Home using the command **show callhome transport**

Example:

```
switch(config-callhome)# show callhome transport
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally use VRFs to send Smart Call Home messages over HTTP.

Configure VRFs to Send Messages Using HTTP

You can use VRFs to send Smart Call Home messages over HTTP. If HTTP VRFs are not configured, the default VRF is used to transport messages over HTTP.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure the VRF used to send email and other Smart Call Home messages over HTTP using the command **transport http use-vrf** *vrf-name*

Example:

```
switch(config-callhome)# transport http use-vrf Blue
```

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about Smart Call Home using the command **show callhome**

Example:

```
switch(config-callhome)# show callhome
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally configure Smart Call Home to send HTTP messages through an HTTP proxy server.

Configure HTTP Proxy Server

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure the HTTP proxy server domain name server (DNS) name, IPv4 address, or IPv6 address using the command **transport http proxy server** *ip-address* [**port** *number*]

Example:

```
switch(config-callhome)# transport http proxy server 192.0.2.1
```

Optionally configures the port number. The port range is from 1 to 65535. The default port number is 8080.

Step 4 Enable Smart Call Home to send all HTTP messages through the HTTP proxy server using the command **transport http proxy enable**

Example:

```
switch(config-callhome)# transport http proxy enable
```

Note You can execute this command only after the proxy server address has been configured.

Note The VRF used for transporting messages through the proxy server is the same as that configured using the **transport http use-vrf** command.

Step 5 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 6 Display the transport-related configuration for Smart Call Home using the command **show callhome transport**

Example:

```
switch(config-callhome)# show callhome transport
```

This is an optional step.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally configure your device to periodically send inventory notifications.

Configure Periodic Inventory Notifications

You can configure the device to periodically send a message with an inventory of all software services currently enabled and running on the device along with hardware inventory information. The device generates two Smart Call Home notifications: periodic configuration messages and periodic inventory messages.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Configure periodic inventory messages using the command **periodic-inventory notification** [**interval** *days*] [**timeofday** *time*]

Example:

```
switch(config-callhome)# periodic-inventory notification interval 20
```

The interval range is from 1 to 30 days, and the default is 7 days. The *time* argument is in HH:MM format. It defines at what time of the day every *X* days an update is sent (where *X* is the update interval).

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about Smart Call Home using the command **show callhome**

Example:

```
switch(config-callhome)# show callhome
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally disable duplicate message throttling.

Disable Duplicate Message Throttling

You can limit the number of duplicate messages received for the same event. By default, the device limits the number of duplicate messages received for the same event. If the number of duplicate messages sent exceeds 30 messages within a 2-hour time frame, the device discards further messages for that alert type.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Disable duplicate message throttling for Smart Call Home using the command **no duplicate-message throttle**

Example:

```
switch(config-callhome)# no duplicate-message throttle
```

Duplicate message throttling is enabled by default.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Enable Smart Call Home.

Enable or Disable Smart Call Home

Once you have configured the contact information, you can enable the Smart Call Home function.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Enable or disable Smart Call Home using the command [**no**] **enable**

Example:

```
switch(config-callhome)# enable
```

Smart Call Home is disabled by default.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally generate a test message.

Configure SMTP-AUTH for Call Home Mail Transfer

You can use the SMTP-AUTH feature for call home mail transfer to share mails in a secure way using standard SMTP-AUTH TCP port 587 or 465, or any other user-defined port, instead of clear text over port 25. This feature is supported from Cisco NX-OS Release 10.2(3)F.

Before you begin

- SMTP-AUTH server certificate should be installed on the switch.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch# configure terminal
switch(config)#
```

Example:

```
switch(config)# callhome
switch(config-callhome)#
```

Step 2 Configure the email address for the person primarily responsible for the device using the command **email-contact** *email-address*

Example:

```
switch(config-callhome)# email-contact admin@Mycompany.com
```

The *email-address* can be up to 255 alphanumeric characters in email address format.

Note You can use any valid email address. The address cannot contain spaces.

Step 3 Create a new destination profile using the command **destination-profile** *name* and set the message format for the profile using the command **destination-profile** *name* **format** { **XML** | **full-txt** | **short-txt** }

Example:

```
switch(config-callhome)# destination-profile testProfile-1
```

Example:

```
switch(config-callhome)# destination-profile testProfile-1 format XML
```

The name can be any alphanumeric string up to 31 characters.

Step 4 Configure an email address to which the secure mail must be delivered using the command **destination-profile** *name* **email-address** *email-address* and associate all the alert groups with the destination profile using the command **destination-profile** *name* **alert-group all**

Example:

```
switch(config-callhome)# destination-profile testProfile-1
index 1 email address person@company.com
```

Example:

```
switch(config-callhome)# destination-profile testProfile-1
alert-group all
```

You can configure up to 50 email addresses in a destination profile.

Step 5 Configure the email from **callhome** field for Smart Call Home messages using the command **transport email from callhome_** *email-address*

Example:

```
switch(config)# transport email from callhome_person@company.com
```

Step 6 Enable SMTP-AUTH mail transfer method using the command **transport email smtp-server** *hostname/ip-address* **port 465 use-vrf** *vrf-name*

transport email smtp-server *hostname/ip-address* **port 587 use-vrf** *vrf-name*

Example:

```
switch(config)# transport email smtp-server 10.1.1.174 port 465
use-vrf management
```

```
switch(config)# transport email smtp-server 10.1.1.174 port 587
use-vrf management
```

STARTTLS-based SMTP-AUTH is over the standard TCP ports, that is, 465 and 587 ports.

Step 7 Accept username and password and passes these details for SMTP-AUTH authentication using the command **transport email username** *username* **passwd** *password* { **cleartext** | **encrypted** }

Example:

```
switch(config)# transport email username user1 passwd Y2FsbGhvbWUK encrypted
```

The username should be alphanumeric and must be less than 256 bytes. Password option can be entered in cleartext or encrypted format (if the user already has the encrypted password). The password length must be less than 64 bytes for the cleartext option and less than 256 bytes for the encrypted option.

Note SMTP-AUTH fails in the following scenarios:

- if the password in cleartext is more than 56 characters in length.
- if the password has any of the following special characters: Dollar sign - **$**, Parentheses - **(** and **)**, Ampersand - **&**, Square Brackets - **[** and **]**, Semicolon - **;** , Question mark - **?** , Vertical bar or pipe - **|** , Apostrophe - **'**, Quotation marks - **'** , **"** , **‘** , **’** , **“** , and **”** and Less-than and More-than signs - > and <

Step 8 Configure the VRF used to send email and other Smart Call Home messages over HTTP using the command **transport http use-vrf** *vrf-name*

Example:

```
switch(config)# transport http use-vrf management
```

This is an optional step.

Step 9 Enable Smart Call Home using the command [**no**] **enable**

Example:

```
switch(config)# enable
```

The no form of this command disables Smart Call Home.

Configure Source Interface

Use this procedure to identify the source-interface in a VRF through which the packet is sent.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
                        switch(config-callhome)#
```

Step 3 Define the source interface for call home http(s) transfer, and when call home is used as the transport for Smart Licensing using the command **source interface** *interface_name*

Example:

```
switch(config-callhome)# source-interface Ethernet1/1
```

Test Smart Call Home Configuration

You can generate a test message to test your Smart Call Home communications.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Send the specified Smart Call Home test message to all configured destinations using the command **callhome send** [**configuration** | **diagnostic**]

Example:

```
switch(config-callhome)# callhome send diagnostic
```

Step 4 Send a test message to all configured destinations using the command **callhome test**

Example:

```
switch(config-callhome)# callhome test
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Additional Reference Information

Table 4: The event triggers and their Smart Call Home message severity levels

Alert Group	Event Name	Description	Smart Call Home Severity Level
Configuration	PERIODIC_CONFIGURATION	Periodic configuration update message.	2
Diagnostic	DIAGNOSTIC_MAJOR_ALERT	GOLD generated a major alert.	7
Diagnostic	DIAGNOSTIC_MINOR_ALERT	GOLD generated a minor alert.	4
Diagnostic	DIAGNOSTIC_NORMAL_ALERT	Smart Call Home generated a normal diagnostic alert.	2
Environmental and CISCO_TAC	FAN_FAILURE	Cooling fan has failed.	5
Environmental and CISCO_TAC	POWER_SUPPLY_ALERT	Power supply warning has occurred.	6
Environmental and CISCO_TAC	POWER_SUPPLY_FAILURE	Power supply has failed.	6
Environmental and CISCO_TAC	POWER_SUPPLY_SHUTDOWN	Power supply has shut down.	6
Environmental and CISCO_TAC	TEMPERATURE_ALARM	Thermal sensor going bad.	6
Environmental and CISCO_TAC	TEMPERATURE_MAJOR_ALARM	Thermal sensor indicates temperature has reached operating major threshold.	6
Environmental and CISCO_TAC	TEMPERATURE_MINOR_ALARM	Thermal sensor indicates temperature has reached operating minor threshold.	4
Inventory and CISCO_TAC	COLD_BOOT	Switch is powered up and reset to a cold boot sequence.	2
Inventory and CISCO_TAC	HARDWARE_INSERTION	New piece of hardware has been inserted into the chassis.	2

Alert Group	Event Name	Description	Smart Call Home Severity Level
Inventory and CISCO_TAC	HARDWARE_REMOVAL	Hardware has been removed from the chassis.	2
Inventory and CISCO_TAC	PERIODIC_INVENTORY	Periodic inventory message has been generated.	2
License	LICENSE_VIOLATION	Feature in use is not licensed and is turned off after grace period expiration.	6
Line module Hardware and CISCO_TAC	LINEmodule_FAILURE	Module operation has failed.	7
Supervisor Hardware and CISCO_TAC	SUP_FAILURE	Supervisor module operation has failed.	7
Syslog-group-port	PORT_FAILURE	syslog message that corresponds to the port facility has been generated.	6
Syslog-group-port	SYSLOG_ALERT	syslog alert message has been generated. **Note** Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.	5
System and CISCO_TAC	SW_CRASH	Software process has failed with a stateless restart, indicating an interruption of a service. Messages are sent for process crashes on supervisor modules.	5
System and CISCO_TAC	SW_SYSTEM_INCONSISTENT	Inconsistency has been detected in software or file system.	5
Test and CISCO_TAC	TEST	User generated test has occurred.	2

Smart Call Home supports the following message formats:

- Short Text Message Format
- Common Event Message Fields
- Alert Group Message Fields
- Reactive and Proactive Event Message Fields
- Inventory Event Message Fields
- User-Generated Test Message Fields

Table 5: Short text formatting option for all message types

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

Table 6: First set of common event message fields for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Timestamp	Date and time stamp of event in ISO time notation: YYYY-MM-DD HH:MM:SS GMT+HH:MM.	/aml/header/time
Message name	Name of message.	/aml/header/name
Message type	Name of message type, such as reactive or proactive.	/aml/header/type
Message group	Name of alert group, such as syslog.	/aml/header/group
Severity level	Severity level of message.	/aml/header/level
Source ID	Product type for routing, such as the Cisco Nexus 9000 Series switch.	/aml/header/source

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Device ID	Unique device identifier (UDI) for the end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is **type@Sid@serial**. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/ header/deviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header/customerID
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header /contractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	/aml/ header/siteId
Server ID	If the message is generated from the device, this ID is the unique device identifier (UDI) of the device. The format is **type@Sid@serial**. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/header/serverId
Message description	Short text that describes the error.	/aml/body/msgDesc

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Device name	Node that experienced the event (hostname of the device).	/aml/body/sysName
Contact name	Name of person to contact for issues associated with the node that experienced the event.	/aml/body/sysContact
Contact email	Email address of person identified as the contact for this unit.	/aml/body/sysContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	/aml/body/sysContactPhone Number
Street address	Optional field that contains the street address for RMA part shipments associated with this unit.	/aml/body/sysStreetAddress
Model name	Model name of the device (the specific model as part of a product family name).	/aml/body/chassis/name
Serial number	Chassis serial number of the unit.	/aml/body/chassis/serialNo
Chassis part number	Top assembly number of the chassis.	/aml/body/chassis/partNo

Table 7: Fields specific to alert group messages for full text and XML

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Command output name	Exact name of the issued CLI command.	/aml/attachments/attachment/name
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed.	/aml/attachments/attachment/atdata



Note These fields may be repeated if multiple CLI commands are executed for an alert group.

Table 8: Reactive and Proactive event message format for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Affected FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
Affected FRU serial number	Serial number of the affected FRU.	/aml/body/fru/serialNo
Affected FRU part number	Part number of the affected FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU that is generating the event message.	/aml/body/fru/slot
FRU hardware version	Hardware version of the affected FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the affected FRU.	/aml/body/fru/swVersion

Table 9: Inventory event message format for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of the chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
FRU s/n	Serial number of the FRU.	/aml/body/fru/serialNo
FRU part number	Part number of the FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU.	/aml/body/fru/slot
FRU hardware version	Hardware version of the FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the FRU.	/aml/body/fru/swVersion

Table 10: User-generated test message format for full text or XML

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample Syslog Alert Notification

- This sample shows the full-text format for a syslog port alert-group notification

```
Severity Level:5
Series:Nexus9000
Switch Priority:0
```

```
Device Id:N9K-C9508@C@TXX12345678
Server Id:N9K-C9508C@TXX12345678
Time of Event:2013-05-17 16:31:33 GMT+0000 Message Name:
Message Type:syslog
System Name:dc3-test
Contact Name:Jay Tester
Contact Email:contact@example.com
Contact Phone:+91-80-1234-5678
Street Address:#1 Any Street
Event Description:SYSLOG_ALERT 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)

syslog_facility:ETHPORT
start chassis information:
Affected Chassis:N9K-C9508
Affected Chassis Serial Number:TXX12345678 Affected Chassis Hardware Version:0.405
Affected Chassis Software Version:6.1(2) Affected Chassis Part No:11-11111-11 end chassis
 information:
start attachment
   name:show logging logfile | tail -n 200
   type:text
   data:
   2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
cleared by user
   2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16:
Invalid argument: - sshd[14484]
   2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: "System Manager
 (gsync controller)" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
   2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 3504)
hasn't caught signal 9 (no core).
   2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
   2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23210)
 hasn't caught signal 9 (no core).
   2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
   2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23294)
 hasn't caught signal 9 (no core).
   2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
   2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is
becoming active.
   2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed
 - device_test
   2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336]
Unrecognized message from MRIB. Major type 1807
   2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
   2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 -
ntpd[19045]
```

```
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined -
ntpd[19045]
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client
filter recovery failed (0)
   2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client
filter recovery failed (0)
   2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
   2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
   2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
   2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
   2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
   2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
   2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
   2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
   2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present
but all AC inputs are not connected, ac-redundancy might be affected
   2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present
but all AC inputs are not connected, ac-redundancy might be affected
   2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
   2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
  2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 4820)
hasn't caught signal 9 (no core).
  2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24239)
 hasn't caught signal 9 (no core).
  2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24401)
 hasn't caught signal 9 (no core).
   2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
  2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24407)
 hasn't caught signal 9 (no core).
   2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
```

```
<more> return code <14>
   2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
  2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
 with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT:
Ethernet3/1) end attachment start attachment
   type:text
   data:

   dc3-test interfaces:
        Ethernet3/1    Ethernet3/2    Ethernet3/3
        Ethernet3/4    Ethernet3/5    Ethernet3/6
        Ethernet3/7    Ethernet3/8    Ethernet3/9
        Ethernet3/10   Ethernet3/11   Ethernet3/12
        Ethernet3/13   Ethernet3/14   Ethernet3/15
        Ethernet3/16   Ethernet3/17   Ethernet3/18
        Ethernet3/19   Ethernet3/20   Ethernet3/21
        Ethernet3/22   Ethernet3/23   Ethernet3/24
        Ethernet3/25   Ethernet3/29   Ethernet3/30
        Ethernet3/31   Ethernet3/32   Ethernet3/33
        Ethernet3/34   Ethernet3/35   Ethernet3/36
        Ethernet3/37   Ethernet3/38   Ethernet3/39
        Ethernet3/40   Ethernet3/41   Ethernet3/42
        Ethernet3/43   Ethernet3/44   Ethernet3/45
        Ethernet3/46   Ethernet3/47   Ethernet3/48
end attachment
start attachment
   type:text
   data:
end attachment
start attachment
   name:show license usage
   type:text
   data:
   Feature Ins Lic Status Expiry Date Comments
             Count
   --------------------------------------------------------------------------------
   LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
   --------------------------------------------------------------------------------
end attachment
```

- This sample shows the XML format for a syslog port alert-group notification

```
<?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:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<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>1004:TXX12345678:478F82E6</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/syslog</aml-block:Type>
<aml-block:CreationDate>2013-05-17 16:31:33 GMT+0000</aml-block:CreationDate>
<aml-block:Builder> <aml-block:Name>DC3</aml-block:Name>
<aml-block:Version>4.1</aml-block:Version>
</aml-block:Builder>
```

```
<aml-block:BlockGroup>
<aml-block:GroupId>1005:TXX12345678:478F82E6</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>5</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome" version="1.0">
<ch:EventTime>2013-05-17 16:31:33 GMT+0000</ch:EventTime>
<ch:MessageDescription>SYSLOG_ALERT 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
 Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1) </ch:MessageDescription>
<ch:Event> <ch:Type>syslog</ch:Type> <ch:SubType></ch:SubType> <ch:Brand>Cisco</ch:Brand>
 <ch:Series>Nexus9000</ch:Series> </ch:Event> <ch:CustomerData> <ch:UserData>
<ch:Email>contact@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:DeviceId>N9K-C9508@C@TXX12345678</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>dc3-test</ch:Name>
<ch:Contact>Jay Tester</ch:Contact> <ch:ContactEmail>contact@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+91-80-1234-5678</ch:ContactPhoneNumber>
<ch:StreetAddress>#1, Any Street</ch:StreetAddress> </ch:SystemInfo> </ch:CustomerData>
 <ch:Device> <rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.1">
<rme:Model>N9K-C9508</rme:Model>
<rme:HardwareVersion>0.405</rme:HardwareVersion>
<rme:SerialNumber>TXX12345678</rme:SerialNumber>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging logfile | tail -n 200</aml-block:Name> <aml-block:Data
encoding="plain">
<![CDATA[2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
 cleared by user
2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16: Invalid
 argument: - sshd[14484]
2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: \"System Manager
(gsync controller)\" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 3504)
hasn&amp;apos;t caught signal 9 (no core).
2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23210)
hasn&amp;apos;t caught signal 9 (no core).
2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
```

```
2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23294)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is becoming
 active.
2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed
- device_test
2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336] Unrecognized
 message from MRIB. Major type 1807
2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
 recovery failed (0)
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
 recovery failed (0)
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but
 all AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but
 all AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 4820)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24239)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24401)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
```

```
2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24407)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
 with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT:
Ethernet3/1) ]]> </aml-block:Data> </aml-block:Attachment> <aml-block:Attachment
type="inline"> <aml-block:Name> <aml-block:Data encoding="plain"> <![CDATA[
dc3-test interfaces:
   Ethernet3/1     Ethernet3/2     Ethernet3/3
   Ethernet3/4     Ethernet3/5     Ethernet3/6
   Ethernet3/7     Ethernet3/8     Ethernet3/9
   Ethernet3/10    Ethernet3/11    Ethernet3/12
   Ethernet3/13    Ethernet3/14    Ethernet3/15
   Ethernet3/16    Ethernet3/17    Ethernet3/18
   Ethernet3/19    Ethernet3/20    Ethernet3/21
   Ethernet3/22    Ethernet3/23    Ethernet3/24
   Ethernet3/25    Ethernet3/26    Ethernet3/27
   Ethernet3/28    Ethernet3/29    Ethernet3/30
   Ethernet3/31    Ethernet3/32    Ethernet3/33
   Ethernet3/34    Ethernet3/35    Ethernet3/36
   Ethernet3/37    Ethernet3/38    Ethernet3/39
   Ethernet3/40    Ethernet3/41    Ethernet3/42
   Ethernet3/43    Ethernet3/44    Ethernet3/45
   Ethernet3/46    Ethernet3/47    Ethernet3/48


]]>
</aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
<aml-block:Name> <aml-block:Data encoding="plain"> <!> </aml-block:Data>
</aml-block:Attachment> <aml-block:Attachment type="inline"> <aml-block:Name>show license
 usage</aml-block:Name> <aml-block:Data encoding="plain">
<![CDATA[Feature Ins Lic Status Expiry Date Comments
                    Count
--------------------------------------------------------------------------------
LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
--------------------------------------------------------------------------------
]]>
</aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

MIBs

MIBs related to Smart Call Home - To locate and download supported MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html