Smart Call Home の設定

この章では、Cisco NX-OS デバイスの Smart Call Home 機能を設定する方法について説明します。

この章は、次の内容で構成されています。

- Smart Call Home の概要, on page 1
- Smart Call Home - 概念 （2 ページ）
- Smart Call Home の前提条件, on page 9
- Smart Call Home の注意事項および制約事項 （9 ページ）
- Smart Call Home のデフォルト設定, on page 9
- Smart Call Home の設定 （10 ページ）
- Smart Call Home 設定の確認, on page 26
- Smart Call Home の設定例 （26 ページ）
- その他の参考資料 （28 ページ）

Smart Call Home の概要

Smart Call Home により、重要なシステム ポリシーについて電子メールベースの通知が提供されます。豊富なメッセージ フォーマットから選択できるので、ポケットベル サービス、標準 E メール、または XML ベースの自動解析アプリケーションとの最適な互換性が得られます。この機能を使用して、ネットワーク サポート エンジニアやネットワーク オペレーション センターを呼び出せます。また、Cisco Smart Call Home サービスを使用して、TAC でケースを自動的に生成することもできます。

Smart Call Home には、次の機能があります。

- 関連する CLI コマンド出力の実行および添付が自動化されます。
- 次のような、複数のメッセージ フォーマット オプションがあります。
  - ショート テキスト：ポケットベルまたは印刷形式のレポートに最適。
  - フルテキスト：人間が判読しやすいように完全にフォーマットされたメッセージ情報です。

- XML：Extensible Markup Language（XML）および Adaptive Messaging Language（AML）XML Schema Definition（XSD）を使用する、調和の取れた判読可能なフォーマット。AML XSD は Cisco.com の Web サイトで公開されています。XML 形式は、Technical Assistance Center とのやり取りの中でも使用されます。

- 複数のメッセージ宛先への同時配信が可能。各宛先プロファイルには最大 50 件の電子メール宛先アドレスを設定できます。

Smart Call Home - 概念

このセクションでは、Smart Call Home に関連するいくつかの概念について説明します。

宛先プロファイル

宛先プロファイルには、次の情報が含まれます。

- 1 つ以上のアラート グループ：アラートの発生時に、特定の Smart Call Home メッセージを送信するアラートのグループ。
- 1 つまたは複数の電子メール宛先：この宛先プロファイルに割り当てられたアラート グループによって生成された Smart Call Home メッセージの受信者リスト。
- メッセージ フォーマット：Smart Call Home メッセージのフォーマット（ショート テキスト、フル テキスト、または XML）。
- メッセージ重大度：Cisco NX-OS が宛先プロファイル内のすべての電子メール アドレスに対して Smart Call Home メッセージを生成するまで、アラートが満たす必要がある Smart Call Home 重大度。アラートの Smart Call Home 重大度が、宛先プロファイルに設定されたメッセージ重大度よりも低い場合、Cisco NX-OS はアラートを生成しません。

定期メッセージを日別、週別、月別で送信するコンポーネント アラート グループを使用して、定期的なコンポーネント アップデート メッセージを許可するよう宛先プロファイルを設定することもできます。

Cisco NX-OS は、次の定義済み宛先プロファイルをサポートします。

- CiscoTAC-1：XML メッセージ フォーマットの Cisco-TAC アラート グループをサポートします。このプロファイルは、callhome@cisco.com という E メール コンタクト、最大メッセージサイズ、およびメッセージ重大度 0 で設定済みです。このプロファイルのデフォルト情報はどれも変更できません。
- full-text-destination：フル テキスト メッセージ フォーマットをサポートします。
- short-text-destination：ショート テキスト メッセージ フォーマットをサポートします。

Smart Call Home アラート グループ

アラート グループは、すべての Cisco Nexus デバイスでサポートされる Smart Call Home アラートの定義済みサブセットです。アラート グループを使用すると、定義済みまたはカスタム宛先プロファイルに送信する一連の Smart Call Home アラートを選択できます。Smart Call Home アラートが宛先プロファイルにアソシエートされたいずれかのアラート グループに属する場合、およびアラートで、Smart Call Home メッセージ重大度が宛先プロファイルに設定されているメッセージ重大度と同じか、それ以上である場合のみ、デバイスは Smart Call Home アラートを宛先プロファイルの電子メールの宛先に送信します。

次の表に、サポートされるアラート グループと、アラート グループ用に生成された Smart Call Home メッセージに含まれるデフォルトの CLI コマンド出力を示します。

Table 1: アラート グループおよび実行されるコマンド

アラート グループ	説明	実行されるコマンド
Cisco-TAC	Smart Call Home 宛ての、他のアラート グループからのすべてのクリティカル アラート。	アラートを発信するアラート グループに基づいてコマンドを実行します。
設定	設定に関連した定期的なイベント。	**show module** **show version**
診断	診断によって生成されたイベント。	**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**

アラート グループ	説明	実行されるコマンド
EEM	EEMによって生成されるイベント	**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**
環境	電源、ファン、および温度アラームなどの環境検知要素に関連するイベント。	**show environment** **show logging last 200** **show module** **show version**
インベントリ	装置がコールドブートした場合、またはFRUの取り付けまたは取り外しを行った場合に示されるコンポーネントステータス。このアラートは重要でないイベントであり、情報はステータスおよび使用権に使用されます。	**show inventory** **show license usage** **show module** **show sprom all** **show system uptime** **show version**
ライセンス	ライセンスおよびライセンス違反に関連するイベント	**show logging last 200**

アラート グループ	説明	実行されるコマンド
ラインカード ハードウェア	標準またはインテリジェント スイッチング モジュールに関連するイベント。	**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**
スーパーバイザ ハードウェア	スーパーバイザ モジュールに関連するイベント。	**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	syslog PORT ファシリティによって生成されるイベント	**show license usage** **show logging last 200**

アラート グループ	説明	実行されるコマンド
システム	装置の動作に必要なソフトウェア システムの障害によって生成されたイベント。	**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**
テスト	ユーザが作成したテスト メッセージ	**show module** **show version**

Smart Call Home は、syslog の重大度を、syslog ポート グループ メッセージの対応する Smart Call Home の重大度に対応させます。

特定のイベントが発生し、Smart Call Home メッセージを含む **show** 出力を送信した場合に、追加の **show** コマンドを実行するために、定義済みのアラート グループをカスタマイズできます。

show コマンドは、フル テキストおよび XML 宛先プロファイルにのみ追加できます。ショート テキスト宛先プロファイルは、128 バイトのテキストに制限されているため、追加の **show** コマンドをサポートしていません。

Smart Call Home のメッセージ レベル

Smart Call Home を使用すると、緊急度に基づいてメッセージをフィルタリングできます。各定義済みまたはユーザ定義宛先プロファイルを、0（最小緊急度）～9（最大緊急度）までの Smart Call Home しきい値と関連付けることができます。デフォルトは 0（全メッセージを送信）です。

syslog 重大度は、Smart Call Home メッセージ レベルにマッピングされています。

Note Smart Call Home と Syslog は異なる重大度を使用します（次の表を参照）。Smart Call Home は、メッセージ テキストで syslog メッセージ レベルを変更しません。

次の表に、各 Smart Call Home メッセージ レベルのキーワードと、syslog ポート アラート グループの対応する syslog レベルを示します。

Table 2: 重大度と *syslog* レベルのマッピング

Smart Call Home レベル	キーワード	Syslog レベル	説明
9	Catastrophic	該当なし	ネットワーク全体に壊滅的な障害が発生しています。
8	Disaster	該当なし	ネットワークに重大な影響が及びます。
7	Fatal	緊急（0）	システムが使用不可能な状態。
6	Critical	アラート（1）	クリティカルな状況で、すぐに対応する必要があります。
5	Major	重要（2）	重大な状態。
4	Minor	エラー（3）	軽微な状態。
3	警告	警告（4）	警告状態。
2	通知	通知（5）	基本的な通知および情報メッセージです。他と関係しない、重要性の低い障害です。
1	標準	情報（6）	標準状態に戻ることを示す標準イベントです。
0	Debugging	デバッグ（7）	デバッグ メッセージ。

Smart Call Home の取得

シスコと直接サービス契約を結んでいる場合は、Smart Call Home サービスに登録できます。Smart Call Home は、Smart Call Home メッセージを分析し、背景説明と推奨措置を提供します。既知の問題、特にオンライン診断障害については、TAC に Automatic Service Request が作成されます。

Smart Call Home には、次の機能があります。

- 継続的なデバイス ヘルス モニタリングとリアルタイムの診断アラート。
- Smart Call Home メッセージの分析。必要に応じて、自動サービス要求（詳細な診断情報が含まれる）が作成され、該当する TAC チームにルーティングされるため、問題解決を高速化できます。
- セキュアなメッセージ転送が、ご使用のデバイスから直接、または HTTP プロキシサーバやダウンロード可能な転送ゲートウェイ（TG）を経由して行われます。TG 集約ポイントは、複数のデバイスをサポートする場合またはセキュリティ要件によって、デバイスをインターネットに直接接続できない場合に使用できます。

- あらゆる Smart Call Home デバイスの Smart Call Home メッセージおよび推奨事項、インベントリ情報、設定情報への Web アクセス。この機能によって、関連するフィールドの注意事項、セキュリティ勧告、および廃止情報にアクセスできます。

登録には次の情報が必要です。

- デバイスの SMARTnet 契約番号
- 電子メール アドレス
- お使いの Cisco.com ID

データベース マージの注意事項

2 つの Smart Call Home データベースをマージする場合は、次の注意事項に従ってください。

- マージされるデータベースには、次の情報が含まれます。
  - マージ側デバイスからの全宛先プロファイルのスーパーセット。
  - 宛先プロファイルの E メール アドレスとアラート グループ。
  - マージ側デバイスにあるその他の設定情報（メッセージスロットリング、定期的なインベントリなど）。
- 宛先プロファイル名は、マージするデバイス内で重複しないようにしてください。コンフィギュレーションが異なっても、同じ名前は使用できません。プロファイル名が重複している場合、重複するプロファイルの 1 つを削除する必要があります。そうしなければマージ処理が失敗します。

高可用性

ステートフルおよびステートレスの両方のリスタートが、Smart Call Home でサポートされます。

仮想化のサポート

Smart Call Home のインスタンスが 1 つサポートされます。次の URL から、Smart Call Home の Web サイトでお客様の連絡先を登録できます。https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

callhome send および **callhome test** コマンドを使用して Smart Call Home をテストできます。

Smart Call Home は Virtual Routing and Forwarding（VRF）を認識します。特定の VRF を使用して Smart Call Home SMTP サーバに接続するように Smart Call Home を設定できます。

Smart Call Home の前提条件

Smart Call Home には、次の前提条件があります。

- 電子メール アドレスにメッセージを送信するには、まず電子メール サーバを設定する必要があります。HTTP を使用してメッセージを送信するには、HTTPS サーバにアクセスでき、Cisco Nexus デバイスに有効な証明書がインストールされている必要があります。
- デバイスは電子メール サーバまたは HTTPS サーバと IP 接続している必要があります。
- まず、コンタクト名（SNMP サーバのコンタクト）、電話番号、および住所情報を設定する必要があります。この手順は、受信メッセージの送信元を判別するために必要です。
- Smart Call Home サービスを使用する場合、設定中のデバイスに対応している現在のサービス契約が必要です。

Smart Call Home の注意事項および制約事項

Smart Call Home には、次の注意事項および制限事項があります。

- IP 接続がない場合、またはプロファイル宛先への仮想ルーティングおよびフォワーディング（VRF）インスタンス内のインターフェイスがダウンしている場合、デバイスは Smart Call Home メッセージを送信できません。
- Smart Call Home はあらゆる SMTP サーバで動作します。
- Smart Call Home には最大 5 個までの SMTP サーバを設定できます。
- Link up/down syslog メッセージは、Smart Call Home メッセージまたはアラート通知をトリガーしません。
- 住所、顧客 ID、サイト ID などの Smart Call Home コマンドを設定する場合は、これらのコマンドをセミコロン区切りでグループ化するのではなく、個別のコマンドとして設定する必要があります。
- Callhome は、**ip http source-interface** コマンドを使用した送信元インターフェイスの指定をサポートしていません。

Smart Call Home のデフォルト設定

このテーブルは、Smart Call Home パラメータのデフォルト設定を示します。

Table 3: デフォルトの *Smart Call Home* パラメータ

パラメータ	デフォルト
フルテキスト フォーマットで送信するメッセージの宛先メッセージ サイズ	2,500,000
XML フォーマットで送信するメッセージの宛先メッセージ サイズ	2,500,000
ショートテキスト フォーマットで送信するメッセージの宛先メッセージ サイズ	4000
ポートを指定しなかった場合の SMTP サーバ ポート	25
プライオリティを指定しなかった場合の SMTP サーバのプライオリティ	50
プロファイルとアラート グループのアソシエート	フルテキスト宛先プロファイルおよびショートテキスト宛先プロファイルの場合はすべて。CiscoTAC-1 宛先プロファイルの場合は cisco-tac アラート グループ
フォーマット タイプ	XML
Smart Call Home のメッセージ レベル	0（ゼロ）
HTTP プロキシ サーバの使用	無効であり、プロキシ サーバは設定されていません。

Smart Call Home の設定

（注） Cisco NX-OS コマンドは Cisco IOS コマンドと異なる場合があるので注意してください。

次の順序で Smart Call Home 設定を行うことを推奨します。

1. 連絡先情報の設定 （11 ページ）
2. 宛先プロファイルの作成 （13 ページ）
3. アラート グループと宛先プロファイルのアソシエート （16 ページ）
4. （オプション）
   アラート グループへの show コマンドの追加 （17 ページ）

5. Smart Call Home のイネーブル化またはディセーブル化 （24 ページ）
6. （オプション）

   Smart Call Home 設定のテスト （25 ページ）

連絡先情報の設定

Smart Call Home には、電子メール、電話番号、住所の各情報を指定する必要があります。契約 ID、カスタマー ID、サイト ID、およびスイッチ プライオリティ情報を任意で指定できます。

これらの Smart Call Home コマンドは、セミコロン区切りでグループ化するのではなく、個別のコマンドとして設定する必要があります。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server contact** *sys-contact* **Example:** `switch(config)# snmp-server contact personname@companyname.com`	SNMP sysContact を設定します。

ステップ 3	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 4	**email-contact** *email-address* **Example:** `switch(config-callhome)# email-contact admin@Mycompany.com`	デバイスの主要責任者の電子メールアドレスを設定します。 *email-address* には、電子メール アドレスの形式で、最大 255 の英数字を使用できます。 **Note** 任意の有効な電子メールアドレスを使用できます。アドレスには、空白を含めることはできません。
ステップ 5	**phone-contact** *international-phone-number* **Example:** `switch(config-callhome)# phone-contact +1-800-123-4567`	デバイスの担当者の電話番号を国際電話フォーマットで設定します。 *international-phone-number* は、最大 17

	Command or Action	Purpose
		文字の英数字で、国際電話フォーマットにする必要があります。 **Note** 電話番号には、空白を含めることはできません。番号の前にプラス（+）プレフィックスを使用します。
ステップ 6	**streetaddress** *address* **Example:** `switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere`	デバイスの主要責任者の住所を空白の含まれる英数字ストリングとして設定します。 *address* には、最大 255 の英数字を使用できます。スペースを使用できます。
ステップ 7	(Optional) **contract-id** *contract-number* **Example:** `switch(config-callhome)# contract-id Contract5678`	サービス契約からこのデバイスの契約番号を設定します。 契約番号は、最大 255 文字の英数字を自由なフォーマットで指定できます。
ステップ 8	(Optional) **customer-id** *customer-number* **Example:** `switch(config-callhome)# customer-id Customer123456`	サービス契約からこのデバイスのカスタマー番号を設定します。 カスタマー番号は、最大 255 文字の英数字を自由なフォーマットで指定できます。
ステップ 9	(Optional) **site-id** *site-number* **Example:** `switch(config-callhome)# site-id Site1`	このデバイスのサイト番号を設定します。 *site-number* は、最大 255 文字の英数字を自由なフォーマットで指定できます。
ステップ 10	(Optional) **switch-priority** *number* **Example:** `switch(config-callhome)# switch-priority 3`	このデバイスのスイッチ プライオリティを設定します。 指定できる範囲は 0 ～ 7 です。0 は最高のプライオリティを、7 は最低のプライオリティを示します。デフォルト値は 7 です。
ステップ 11	**commit** **Example:** `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。

	Command or Action	Purpose
ステップ 12	(Optional) **show callhome** **Example:** `switch(config-callhome)# show callhome`	Smart Call Home コンフィギュレーションの概要を表示します。
ステップ 13	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

宛先プロファイルを作成します。

宛先プロファイルの作成

ユーザ定義宛先プロファイルを作成し、メッセージ フォーマットを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**destination-profile** *name* 例： `switch(config-callhome)# destination-profile Noc101`	新しい宛先プロファイルを作成します。名前は、最大 31 文字の英数字で指定できます。
ステップ 4	**destination-profile** *name* **format** {**XML** | **full-txt** | **short-txt**} 例： `switch(config-callhome)# destination-profile Noc101 format full-txt`	プロファイルのメッセージ フォーマットを設定します。名前は、最大 31 文字の英数字で指定できます。

	コマンドまたはアクション	目的
ステップ 5	**commit** 例： `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 6	（任意） **show callhome destination-profile** [**profile** *name*] 例： `switch(config-callhome)# show callhome destination-profile profile Noc101`	1 つまたは複数の宛先プロファイルに関する情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

次のタスク

1 つの宛先プロファイルに 1 つまたは複数のアラート グループを関連付けます。

宛先プロファイルの変更

定義済みまたはユーザ定義の宛先プロファイルの次の属性を変更できます。

- 宛先メール アドレス：アラートの送信先となる実際のアドレス（トランスポート メカニズムに関係します）。
- 宛先 URL：アラートの送信先となる HTTP または HTTPS URL。
- 転送方式：E メールまたは HTTP 転送によって、使用される宛先アドレスのタイプが決まります。
- メッセージ フォーマット：アラート送信に使用されるメッセージ フォーマット（フル テキスト、ショート テキスト、または XML）。
- メッセージ レベル：この宛先プロファイルの Smart Call Home メッセージの重大度。
- メッセージ サイズ：この宛先プロファイルの E メール アドレスに送信された Smart Call Home メッセージの長さ。

Procedure

	Command or Action	Purpose
ステップ 1	**configure terminal** **Example:**	グローバル コンフィギュレーション モードを開始します

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

ステップ 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**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`	ユーザ定義または定義済みの宛先プロファイルに E メール アドレスを設定します。宛先プロファイルには、最大 50 個の電子メール アドレスを設定できます。
ステップ 4	**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`	ユーザ定義または定義済み宛先プロファイルの HTTP または HTTPS URL を設定します。URL の最大文字数は 255 文字です。
ステップ 5	**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`	ユーザ定義または定義済み宛先プロファイルに対応する電子メールまたは HTTP 転送方式を設定します。選択する転送方式のタイプによって、そのタイプに設定された宛先アドレスが決まります。
ステップ 6	**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`	この宛先プロファイルの Smart Call Home メッセージの重大度を設定します。Cisco NX-OS では、Smart Call Home 重大度が一致する、またはそれ以上であるアラートのみが、このプロファイルの宛先に送信されます。指定できる範囲は 0 ～ 9 です。9 は最大の重大度を示します。
ステップ 7	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-size** *number* **Example:**	この宛先プロファイルの最大メッセージ サイズを設定します。範囲は 0 ～ 5000000 です。デフォルト値は 2500000 です。

	Command or Action	Purpose
	`switch(config-callhome)# destination-profile full-txt-destination message-size 100000`	
ステップ 8	**commit** **Example:** `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 9	(Optional) **show callhome destination-profile** [**profile** *name*] **Example:** `switch(config-callhome)# show callhome destination-profile profile full-text-destination`	1 つまたは複数の宛先プロファイルに関する情報を表示します。
ステップ 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

1 つの宛先プロファイルに 1 つまたは複数のアラート グループを関連付けます。

アラート グループと宛先プロファイルのアソシエート

Procedure

	Command or Action	Purpose
ステップ 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	グローバル コンフィギュレーション モードを開始します
ステップ 2	**callhome** **Example:** `switch(config)# callhome switch(config-callhome)#`	Smart Call Home コンフィギュレーション モードを開始します。
ステップ 3	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **alert-group** {**All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** |	アラート グループをこの宛先プロファイルにアソシエートします。キーワード **All** を使用して、すべてのアラート グループをこの宛先プロファイルにアソシエートします。

	Command or Action	Purpose
	Supervisor-Hardware | **Syslog-group-port** | **System** | **Test**} **Example:** `switch(config-callhome)# destination-profile Noc101 alert-group All`	
ステップ 4	**commit** **Example:** `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 5	(Optional) **show callhome destination-profile** [**profile** *name*] **Example:** `switch(config-callhome)# show callhome destination-profile profile Noc101`	1 つまたは複数の宛先プロファイルに関する情報を表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

任意で **show** コマンドをアラート グループに追加し、SMTP 電子メール サーバを設定します。

アラート グループへの show コマンドの追加

1 つのアラート グループには、最大 5 個のユーザー定義 CLI **show** コマンドを割り当てることができます。

Note CiscoTAC-1 宛先プロファイルには、ユーザ定義の CLI **show** コマンドを追加できません。

Procedure

	Command or Action	Purpose
ステップ 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	グローバル コンフィギュレーション モードを開始します

	Command or Action	Purpose

ステップ 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**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`	**show** コマンド出力を、このアラート グループに送信された Smart Call Home メッセージに追加します。有効な **show** コマンドだけが受け入れられます。
ステップ 4	**commit** **Example:** `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 5	(Optional) **show callhome user-def-cmds** **Example:** `switch(config-callhome)# show callhome user-def-cmds`	アラート グループに追加されたすべてのユーザ定義 **show** コマンドに関する情報を表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

SMTP 電子メール サーバに接続するように Smart Call Home を設定します。

電子メール サーバの設定

Smart Call Home 機能が動作するよう SMTP サーバ アドレスを設定します。送信元および返信先 E メール アドレスも設定できます。

Smart Call Home には最大 5 個までの SMTP サーバを設定できます。サーバは、プライオリティに基づいて試行されます。最もプライオリティの高いサーバが最初に試行されます。メッセージが送信できない場合、制限に達するまでリスト内の次のサーバが試行されます。2 つのサーバのプライオリティが同じ場合は、先に設定された方が最初に試行されます。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**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`	ドメイン ネーム サーバ（DNS）名、IPv4 アドレス、または IPv6 アドレスのいずれかとして SMTP サーバを設定します。任意でポート番号を設定します。ポート範囲は 1 ～ 65535 です。デフォルト ポート番号は、25 です。 任意で、SMTP サーバのプライオリティを設定します。プライオリティの範囲は 1 ～ 100 で、1 が最高、100 が最低のプライオリティです。プライオリティを指定しない場合、デフォルト値の 50 が使用されます。 また、この SMTP サーバと通信する際に使用するよう任意で VRF を設定します。指定された VRF は、HTTP を使用したメッセージの送信には使用されません。
ステップ 4	(Optional) **transport email from** *email-address* **Example:** `switch(config-callhome)# transport email from person@company.com`	Smart Call Home メッセージの送信元電子メール フィールドを設定します。
ステップ 5	(Optional) **transport email reply-to** *email-address* **Example:** `switch(config-callhome)# transport email reply-to person@company.com`	Smart Call Home メッセージの返信先電子メール フィールドを設定します。
ステップ 6	**commit** **Example:**	Smart Call Home 設定コマンドをコミットします。

	Command or Action	Purpose
	`switch(config-callhome)# commit`	
ステップ 7	(Optional) **show callhome transport** **Example:** `switch(config-callhome)# show callhome transport`	Smart Call Home に対する転送関係のコンフィギュレーションを表示します。
ステップ 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

任意で、VRF を使用して HTTP で Smart Call Home メッセージを送信します。

HTTP を使用したメッセージ送信のための VRF 設定

VRF を使用すると、HTTP で Call Home メッセージを送信できます。HTTP VRF が設定されていない場合は、デフォルトの VRF を使用して HTTP でメッセージが転送されます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**transport http use-vrf** *vrf-name* 例： `switch(config-callhome)# transport http use-vrf Blue`	HTTP で電子メールおよび他の Smart Call Home メッセージを送信するための VRF を設定します。
ステップ 4	**commit** 例： `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。

	コマンドまたはアクション	目的
ステップ 5	（任意） **show callhome** 例： `switch(config-callhome)# show callhome`	Smart Call Home に関する情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

次のタスク

任意で、HTTP プロキシ サーバから HTTP メッセージを送信するように Smart Call Home を設定します。

HTTP プロキシ サーバの設定

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**transport http proxy server** *ip-address* [**port** *number*] 例： `switch(config-callhome)# transport http proxy server 192.0.2.1`	HTTP プロキシ サーバのドメイン ネーム サーバ（DNS）の名前、IPv4 アドレス、またはIPv6アドレスを設定します。任意でポート番号を設定します。ポート範囲は 1 ～ 65535 です。デフォルトのポート番号は 8080 です。
ステップ 4	**transport http proxy enable** 例： `switch(config-callhome)# transport http proxy enable`	Smart Call Home で、HTTP プロキシ サーバ経由ですべてのHTTP メッセージを送信できるようにします。 （注）

	コマンドまたはアクション	目的
		プロキシサーバアドレスが設定された後にだけ、このコマンドを実行できます。 （注） プロキシ サーバを経由してメッセージを転送するために使用する VRF は、**transport http use-vrf** コマンドを使用して設定したものと同じです。
ステップ 5	**commit** 例： `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 6	（任意） **show callhome transport** 例： `switch(config-callhome)# show callhome transport`	Smart Call Home に対する転送関係のコンフィギュレーションを表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

次のタスク

任意で、定期的にインベントリ通知を送信するようにデバイスを設定します。

定期的なインベントリ通知の設定

デバイス上で現在有効にされて動作しているすべてのソフトウェア サービスのインベントリとともに、ハードウェア インベントリ情報を示すメッセージを定期的に送信するように、デバイスを設定できます。デバイスは 2 つの Smart Call Home 通知（定期的な設定メッセージと定期的なインベントリ メッセージ）を生成します。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

	Command or Action	Purpose

ステップ 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**periodic-inventory notification** [**interval** *days*] [**timeofday** *time*] **Example:** `switch(config-callhome)# periodic-inventory notification interval 20`	定期的なインベントリ メッセージを設定します。間隔の範囲は 1 ～ 30 日で、デフォルトは 7 です。*time* 引数は HH:MM の形式です。これは、*X* 日ごとに更新が送信される日の時間を定義します（ここで *X* は更新間隔です）。
ステップ 4	**commit** **Example:** `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 5	(Optional) **show callhome** **Example:** `switch(config-callhome)# show callhome`	Smart Call Home に関する情報を表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

What to do next

任意で重複メッセージ スロットリングを無効にします。

重複メッセージ抑制のディセーブル化

同じイベントについて受信する重複メッセージの数を制限できます。デフォルトでは、デバイスは同じイベントについて受け取る重複メッセージの数を制限します。2 時間の時間枠内で送信された重複メッセージの数が 30 メッセージを超えると、デバイスは同じアラート タイプの以降のメッセージを廃棄します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**no duplicate-message throttle** 例： `switch(config-callhome)# no duplicate-message throttle`	Smart Call Home の重複メッセージ抑制をディセーブルにします。 重複メッセージ抑制はデフォルトでイネーブルです。
ステップ 4	**commit** 例： `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

次のタスク

Smart Call Home をイネーブルにします。

Smart Call Home のイネーブル化またはディセーブル化

担当者情報を設定した場合、Smart Call Home 機能を有効にできます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	[**no**] **enable** 例： `switch(config-callhome)# enable`	Smart Call Home をイネーブルまたはディセーブルにします。 Smart Call Home は、デフォルトでディセーブルです。
ステップ 4	**commit** 例： `switch(config-callhome)# commit`	Smart Call Home 設定コマンドをコミットします。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

次のタスク

任意でテスト メッセージを生成します。

Smart Call Home 設定のテスト

テスト メッセージを生成して Smart Call Home 通信をテストできます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**callhome**
	例：
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Smart Call Home コンフィギュレーション モードを開始します。

ステップ 3	**callhome send** [**configuration** | **diagnostic**] 例： `switch(config-callhome)# callhome send diagnostic`	設定されたすべての宛先に指定の Smart Call Home テスト メッセージを送信します。

	コマンドまたはアクション	目的
ステップ 4	**callhome test** 例： `switch(config-callhome)# callhome test`	設定されたすべての宛先にテスト メッセージを送信します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

Smart Call Home 設定の確認

Smart Call Home 設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show callhome	Smart Call Home 設定を表示します。
show callhome destination-profile *name*	1 つまたは複数の Smart Call Home 宛先プロファイルを表示します。
show callhome transport	Smart Call Home に対する転送関係のコンフィギュレーションを表示します。
show callhome user-def-cmds	任意のアラート グループに追加された CLI コマンドを表示します。
show running-config callhome [**all**]	Smart Call Home の実行コンフィギュレーションを表示します。
show startup-config callhome	Smart Call Home のスタートアップ コンフィギュレーションを表示します。
show tech-support callhome	Smart Call Home のテクニカル サポート出力を表示します。

Smart Call Home の設定例

Noc101 という宛先プロファイルを作成し、コンフィギュレーションのアラート グループをこのプロファイルに関連付けて、コンタクト情報と電子メールの情報を設定した後で、HTTP を介して Smart Call Home メッセージを送信するための VRF を指定する例を示します。Noc101 という宛先プロファイルを作成し、コンフィギュレーションのアラート グループをこのプロ

ファイルに関連付けて、コンタクト情報と電子メールの情報を設定した後で、HTTP を介して Call Home メッセージを送信するための VRF を指定する例を示します。

```
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
```

次に、複数の SMTP サーバを Smart Call Home メッセージに設定する例を示します。

```
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
```

上記のコンフィギュレーションに基づいて、SMTP サーバはこの順序で試行されます。

10.1.1.174（プライオリティ 0）

192.0.2.10（プライオリティ 4）

172.21.34.193（プライオリティ 50、デフォルト）

64.72.101.213（プライオリティ 60）

（注） **transport email smtp-server** コマンドのプライオリティは、最大の 0 です。このコマンドで指定されたサーバは最初に試行され、次に、**transport email mail-server** コマンドで指定されたサーバが、プライオリティの順に試行されます。

次に、HTTP プロキシ サーバからの HTTP メッセージを送信するように、Smart Call Home を設定する例を示します。

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

その他の参考資料

イベント トリガ

次の表に、イベント トリガおよび Smart Call Home メッセージの重大度を示します。

アラート グループ	イベント名	説明	Smart Call Home 重大度
設定（Configuration）	PERIODIC_CONFIGURATION	定期的コンフィギュレーション アップデート メッセージ	2
診断	DIAGNOSTIC_MAJOR_ALERT	GOLD が生成したメジャー アラート	7
	DIAGNOSTIC_MINOR_ALERT	GOLD が生成したマイナー アラート	4
	DIAGNOSTIC_NORMAL_ALERT	Smart Call Home が生成した通常の診断アラート	2
環境および CISCO_TAC	FAN_FAILURE	冷却ファンが障害になりました。	5
	POWER_SUPPLY_ALERT	電源モジュールに関する警告の発生	6
	POWER_SUPPLY_FAILURE	電源モジュールの故障	6
	POWER_SUPPLY_SHUTDOWN	電源モジュールのシャットダウン	6
	TEMPERATURE_ALARM	温度センサーの障害	6
	TEMPERATURE_MAJOR_ALARM	温度が動作メジャーしきい値を超えたことを示す温度センサーの表示	6
	TEMPERATURE_MINOR_ALARM	温度が動作マイナーしきい値を超えたことを示す温度センサーの表示	4

アラート グループ	イベント名	説明	Smart Call Home 重大度
インベントリおよび CISCO_TAC	COLD_BOOT	スイッチの電源が投入され、コールド ブート シーケンスにリセットされます。	2
	HARDWARE_INSERTION	シャーシへの新しいハードウェア コンポーネントの追加	2
	HARDWARE_REMOVAL	シャーシからのハードウェアの取り外し	2
	PERIODIC_INVENTORY	定期的インベントリ メッセージの作成	2
ライセンス	LICENSE_VIOLATION	使用中の機能にライセンスがなく、猶予期間を経てオフになった場合	6
Line module Hardware および CISCO_TAC	LINEmodule_FAILURE	モジュールの動作障害	7
スーパーバイザ ハードウェアおよび CISCO_TAC	SUP_FAILURE	スーパーバイザ モジュールの動作障害	7
Syslog グループ ポート	PORT_FAILURE	ポート ファシリティに対応する syslog メッセージの生成	6
	SYSLOG_ALERT	syslog アラート メッセージの生成 （注） Link up/down syslog メッセージは、Smart Call Home メッセージまたはアラート通知をトリガーしません。	5
システムおよび CISCO_TAC	SW_CRASH	ステートレス リスタートによるソフトウェア プロセス障害、つまりサービスの停止スーパーバイザ モジュールでのプロセス クラッシュに対してメッセージが送信されます。	5
	SW_SYSTEM_INCONSISTENT	ソフトウェアまたはファイル システムにおける不整合の検出	5

アラート グループ	イベント名	説明	Smart Call Home 重大度
テストおよび CISCO_TAC	TEST	ユーザが作成したテストの発生	2

メッセージ フォーマット

Smart Call Home では、次のメッセージ フォーマットがサポートされます。

ショート テキスト メッセージ フォーマット

次の表に、すべてのメッセージ タイプのショート テキスト書式設定オプションを示します。

データ項目	説明
デバイス ID	設定されたデバイス名
日時スタンプ	起動イベントのタイム スタンプ
エラー判別メッセージ	起動イベントの簡単な説明（英語）
アラームの緊急度	エラー レベル（システム メッセージに適用されるエラー レベルなど）

共通のイベント メッセージ フィールド

次の表では、フル テキストまたは XML メッセージに共通するイベント メッセージ フィールドの最初のセットについて説明します。

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
Timestamp	ISO 時刻通知でのイベントの日付/タイム スタンプ YYYY-MM-DD HH:MM:SS GMT+HH:MM	/aml/header/time
メッセージ名	メッセージの名前。	/aml/header/name
メッセージ タイプ	リアクティブまたはプロアクティブなどのメッセージ タイプの名前。	/aml/header/type
メッセージ グループ	Syslog などのアラート グループの名前。	/aml/header/group
重大度	メッセージの重大度	/aml/header/level

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
送信元 ID	ルーティング製品タイプ（Cisco Nexus 9000 シリーズ スイッチなど）。	/aml/header/source
デバイス ID	メッセージを生成したエンド デバイスの固有デバイス識別情報（UDI）。メッセージがデバイスに対して固有でない場合は、このフィールドを空にする必要があります。形式は、*type@Sid@serial* です。 • *type* は、バックプレーン IDPROM からの製品の型番です。 • @ は区切り文字です。 • *Sid* は C で、シリアル ID をシャーシ シリアル番号として特定します。 • *serial* は、Sid フィールドによって識別される番号です。 例：N9K-C9508@C@12345678	/aml/ header/deviceId
カスタマー ID	サポート サービスによって契約情報やその他の ID に使用されるオプションのユーザ設定可能なフィールド	/aml/ header/customerID
連絡先 ID	サポート サービスによって契約情報やその他の ID に使用されるオプションのユーザ設定可能なフィールド	/aml/ header /contractId
サイト ID	シスコが提供したサイト ID または別のサポート サービスにとって意味のあるその他のデータに使用されるオプションのユーザ設定可能なフィールド	/aml/ header/siteId

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
Server ID	デバイスからメッセージが生成された場合、この ID はデバイスの Unique Device Identifier（UDI）フォーマットです。形式は、*type@Sid@serial* です。 • *type* は、バックプレーン IDPROM からの製品の型番です。 • @ は区切り文字です。 • *Sid* は C で、シリアル ID をシャーシ シリアル番号として特定します。 • *serial* は、Sid フィールドによって識別される番号です。 例：N9K-C9508@C@12345678	/aml/header/serverId
メッセージの説明	エラーを説明するショート テキスト。	/aml/body/msgDesc
デバイス名	イベントが発生したノード（デバイスのホスト名）。	/aml/body/sysName
担当者名	イベントが発生したノード関連の問題について問い合わせる担当者名。	/aml/body/sysContact
[連絡先電子メール（Contact email）]	この装置の担当者の電子メール アドレス。	/aml/body/sysContactEmail
連絡先電話番号	このユニットの連絡先である人物の電話番号	/aml/body/sysContactPhone Number
住所	この装置関連の返品許可（RMA）部品の送付先住所を保存するオプション フィールド。	/aml/body/sysStreetAddress
モデル名	デバイスのモデル名（製品ファミリ名に含まれる具体的なモデル）。	/aml/body/chassis/name
シリアル番号	ユニットのシャーシのシリアル番号	/aml/body/chassis/serialNo

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
シャーシの部品番号	シャーシの最上アセンブリ番号	/aml/body/chassis/partNo

アラート グループ メッセージ フィールド

次の表に、フル テキストおよび XML のアラート グループ メッセージに固有のフィールドについて説明します。1 つのアラート グループに対して複数の CLI コマンドが実行される場合は、これらのフィールドが繰り返されることがあります。

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
Command output name	実行された CLI コマンドの正確な名前。	/aml/attachments/attachment/name
添付ファイルの種類	特定のコマンド出力。	/aml/attachments/attachment/type
MIME タイプ	プレーン テキストまたは符号化タイプ。	/aml/attachments/attachment/mime
コマンド出力テキスト	自動的に実行されるコマンドの出力	/aml/attachments/attachment/atdata

リアクティブおよびプロアクティブ イベント メッセージのフィールド

次の表では、フルテキストまたはXMLメッセージのリアクティブおよびプロアクティブ イベント メッセージ形式について説明します。

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
シャーシのハードウェア バージョン	シャーシのハードウェア バージョン。	/aml/body/chassis/hwVersion
スーパーバイザ モジュールのソフトウェア バージョン	最上レベルのソフトウェア バージョン	/aml/body/chassis/swVersion
影響のある FRU 名	イベント メッセージを生成する関連 FRU の名前。	/aml/body/fru/name
影響のある FRU のシリアル番号	関連 FRU のシリアル番号。	/aml/body/fru/serialNo
影響のある FRU の製品番号	関連 FRU の部品番号。	/aml/body/fru/partNo
FRU スロット	イベント メッセージを生成する FRU のスロット番号。	/aml/body/fru/slot

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
FRU ハードウェア バージョン	関連 FRU のハードウェア バージョン。	/aml/body/fru/hwVersion
FRU ソフトウェアのバージョン	関連 FRU で稼働しているソフトウェア バージョン。	/aml/body/fru/swVersion

インベントリ イベント メッセージのフィールド

次の表に、フル テキストまたは XML メッセージのコンポーネント イベント メッセージ形式について説明します。

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
シャーシのハードウェア バージョン	シャーシのハードウェア バージョン。	/aml/body/chassis/hwVersion
スーパーバイザ モジュールのソフトウェア バージョン	最上レベルのソフトウェア バージョン	/aml/body/chassis/swVersion
FRU 名	イベント メッセージを生成する関連 FRU の名前。	/aml/body/fru/name
FRU s/n	FRU のシリアル番号。	/aml/body/fru/serialNo
FRU 製品番号	FRU の部品番号。	/aml/body/fru/partNo
FRU スロット	FRU のスロット番号。	/aml/body/fru/slot
FRU ハードウェア バージョン	FRU のハードウェア バージョン。	/aml/body/fru/hwVersion
FRU ソフトウェアのバージョン	FRU で稼働しているソフトウェア バージョン。	/aml/body/fru/swVersion

ユーザが作成したテスト メッセージのフィールド

次の表に、フルテキストまたはXMLのユーザが作成したテストメッセージ形式について説明します。

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
プロセス ID	固有のプロセス ID	/aml/body/process/id
プロセス状態	プロセスの状態（実行中、中止など）	/aml/body/process/processState

データ項目（プレーン テキストおよび XML）	説明（プレーン テキストおよび XML）	XML タグ（XML のみ）
プロセス例外	原因コードの例外	/aml/body/process/exception

フル テキスト形式での syslog アラート通知の例

次の例では、Syslog ポート アラート グループ通知のフル テキスト形式を示します。

```
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
```

XML 形式での syslog アラート通知の例

次の例では、Syslog ポート アラート グループ通知の XML を示します。

```
<?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'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) ]]> </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>
```

MIB

MIB	MIB のリンク
Smart Call Home に関連する MIB	サポートされている MIB を検索およびダウンロ 次の URL にアクセスしてください。 https://cisco.github.io/cisco-mibs/supportlists/nexus9 Nexus9000MIBSupportList.html

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。