



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。この章の内容は、次のとおりです。

- [Smart Call Home の概要, 1 ページ](#)
- [Smart Call Home のライセンス要件, 9 ページ](#)
- [Smart Call Home の前提条件, 9 ページ](#)
- [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, 10 ページ](#)
- [Smart Call Home の設定, 11 ページ](#)
- [Smart Call Home 設定の確認, 29 ページ](#)
- [Smart Call Home の設定例, 30 ページ](#)
- [その他の参考資料, 31 ページ](#)

Smart Call Home の概要

Smart Call Home では、重要なシステム ポリシーに対し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 エンジニアにポケットベルで連絡したり、ネットワーク オペレーション センターに電子メールを送信したりできます。また、Cisco Smart Call Home サービスを使用して TAC のケースを自動的に生成できます。

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 フォーマットでは、TAC との通信が可能になります。
- 複数のメッセージ宛先への同時配信が可能。それぞれの宛先プロファイルには、最大 50 個の電子メール宛先アドレスを設定できます。

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 つ以上の電子メール宛先：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の重大度に満たない場合、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。

表 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 宛での、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 (Configuration)	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
組み込みイベントマネージャ (EEM)	EEMによって生成されるイベント	show diagnostic result module all detail show diagnostic result modulenumdetail show module show tech-support gold show tech-support ha show tech-support platform
Environmental	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
Test	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 CLI **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。



(注) 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 レベルを一覧表示します。

表 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が、ご使用のデバイスから直接、HTTP プロキシサーバを経由して転送されるか、またはダウンロード可能な転送ゲートウェイ（TG）から転送されます。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、関連する現場の注意事項、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
- Cisco.com ID

Smart Call Home の詳細については、次の Smart Call Home のページを参照してください。 https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

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電子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、同じ名前を使用できません。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。

ハイ アベイラビリティ

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、仮想ルーティングおよびフォワーディング (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 のライセンス要件

製品	ライセンス要件
Cisco NX-OS	Smart Call Home にはライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS ライセンス方式の詳細については、『 <i>Cisco NX-OS Licensing Guide</i> 』を参照してください。

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 E 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転送 (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 に対して最大 5 つの SMTP サーバを設定できます。

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。

表 3: デフォルトの *Smart Call Home* パラメータ

パラメータ (Parameters)	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 グループの関連付け	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 メッセージ レベル	0 (ゼロ)
HTTP プロキシ サーバの使用	ディセーブルであり、プロキシサーバは設定されていない

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ります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関連付け](#), (18 ページ)
- 4 (任意) [アラートグループへの show コマンドの追加](#), (19 ページ)
- 5 [Smart Call Home のイネーブル化またはディセーブル化](#), (27 ページ)
- 6 (任意) [Smart Call Home 設定のテスト](#), (28 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。

手順の概要

1. **configure terminal**
2. **snmp-server contact***sys-contact*
3. **callhome**
4. **email-contact***email-address*
5. **phone-contact***international-phone-number*
6. **streetaddress***address*
7. (任意) **contract-id***contract-number*
8. (任意) **customer-id***customer-number*
9. (任意) **site-id***site-number*
10. (任意) **switch-priority***number*
11. **commit**
12. (任意) **show callhome**
13. (任意) **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： <pre>switch# configure terminal switch(config)#</pre>	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 <i>sys-contact</i> 例： <pre>switch(config)# snmp-server contact personname@companyname.com</pre>	SNMP sysContact を設定します。
ステップ 3	callhome 例： <pre>switch(config)# callhome switch(config-callhome)#</pre>	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 <i>email-address</i> 例： <pre>switch(config-callhome)# email-contact admin@Mycompany.com</pre>	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。 <i>email-address</i> 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。 (注) 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 <i>international-phone-number</i> 例： <pre>switch(config-callhome)# phone-contact +1-800-123-4567</pre>	デバイスの主要責任者の電話番号を国際電話フォーマットで設定します。 <i>international-phone-number</i> は、最大 17 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。 (注) 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 <i>address</i> 例： <pre>switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere</pre>	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。 <i>address</i> 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	contract-id <i>contract-number</i> 例： <pre>switch(config-callhome)# contract-id Contract5678</pre>	(任意) 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。 <i>contract-number</i> 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。

	コマンドまたはアクション	目的
ステップ 8	customer-id <i>customer-number</i> 例： switch(config-callhome)# customer-id Customer123456	(任意) サービス契約からこのデバイスの顧客番号を設定します。 <i>customer-number</i> 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	site-id <i>site-number</i> 例： switch(config-callhome)# site-id Site1	(任意) このデバイスのサイト番号を設定します。 <i>site-number</i> 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	switch-priority <i>number</i> 例： switch(config-callhome)# switch-priority 3	(任意) 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 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 12	show callhome 例： switch(config-callhome)# show callhome	(任意) 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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **destination-profilename**
4. **destination-profilenameformat** {XML | full-txt | short-txt}
5. **commit**
6. (任意) **show callhome destination-profile** [profilename]
7. (任意) **copy running-config startup-config**

手順の詳細

	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-profilename 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-profilenameformat {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。
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	show callhome destination-profile [profilename] 例： switch(config-callhome)# show callhome destination-profile profile Noc101	(任意) 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。

	コマンドまたはアクション	目的
ステップ 7	copy running-config startup-config 例： <pre>switch(config)# copy running-config startup-config</pre>	(任意) 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。

次の作業

宛先プロフィールに 1 つまたは複数のアラート グループを関連付けます。

宛先プロフィールの変更

定義済みまたはユーザ定義の宛先プロフィールの次の属性を変更できます。

- 宛先電子メールアドレス：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：電子メールまたは 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：この宛先プロフィールの Smart Call Home メッセージの重大度
- メッセージサイズ：この宛先プロフィールの電子メールアドレスに送信された Smart Call Home メッセージの許容長さ。

手順の概要

1. **configure terminal**
2. **callhome**
3. **destination-profile** {*name* | CiscoTAC-1 | full-txt-destination | short-txt-destination} **email-addr***address*
4. **destination-profile** {*name* | CiscoTAC-1 | full-txt-destination | short-txt-destination} **http***address*
5. **destination-profile** {*name* | CiscoTAC-1 | full-txt-destination | short-txt-destination} **transport-method** {*email* | *http*}
6. **destination-profile** {*name* | CiscoTAC-1 | full-txt-destination | short-txt-destination} **message-level***number*
7. **destination-profile** {*name* | CiscoTAC-1 | full-txt-destination | short-txt-destination} **message-size***number*
8. **commit**
9. (任意) **show callhome destination-profile** [*profile**name*]
10. (任意) **copy running-config startup-config**

手順の詳細

	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 { <i>name</i> CiscoTAC-1 full-txt-destination short-txt-destination} email-addr <i>address</i> 例： 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電子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 { <i>name</i> CiscoTAC-1 full-txt-destination short-txt-destination} http <i>address</i> 例： switch(config-callhome)# destination-profile CiscoTAC-1 http http://site.com/service/callhome	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。

	コマンドまたはアクション	目的
ステップ 5	destination-profile { <i>name</i> CiscoTAC-1 full-txt-destination short-txt-destination } transport-method { email http } 例： <pre>switch(config-callhome)# destination-profile CiscoTAC-1 transport-method http</pre>	ユーザ定義または定義済み宛先プロフィー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 { <i>name</i> CiscoTAC-1 full-txt-destination short-txt-destination } message-level <i>number</i> 例： <pre>switch(config-callhome)# destination-profile full-txt-destination message-level 5</pre>	この宛先プロフィールの Smart Call Home メッセージの重大度を設定します。Cisco NX-OS がこのプロフィールの宛先に送信するのは、Smart Call Home の重大度が同じか、それ以上のアラートだけです。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 { <i>name</i> CiscoTAC-1 full-txt-destination short-txt-destination } message-size <i>number</i> 例： <pre>switch(config-callhome)# destination-profile full-txt-destination message-size 100000</pre>	この宛先プロフィー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。
ステップ 8	commit 例： <pre>switch(config-callhome)# commit</pre>	Smart Call Home 設定コマンドをコミットします。
ステップ 9	show callhome destination-profile [profile <i>filename</i>] 例： <pre>switch(config-callhome)# show callhome destination-profile profile full-text-destination</pre>	(任意) 1 つまたは複数の宛先プロフィールに関する情報を表示します。
ステップ 10	copy running-config startup-config 例： <pre>switch(config)# copy running-config startup-config</pre>	(任意) 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。

次の作業

宛先プロフィールに 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関連付け

手順の概要

1. **configure terminal**
2. **callhome**
3. **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}
4. **commit**
5. (任意) **show callhome destination-profile** [*profilename*]
6. (任意) **copy running-config startup-config**

手順の詳細

	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 { <i>name</i> 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} 例： switch(config-callhome)# destination-profile Noc101 alert-group All	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。
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	show callhome destination-profile [<i>profilename</i>] 例： switch(config-callhome)# show callhome destination-profile profile Noc101	(任意) 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。

	コマンドまたはアクション	目的
ステップ 6	copy running-config startup-config 例： <pre>switch(config)# copy running-config startup-config</pre>	(任意) 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。

次の作業

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ユーザ定義の CLI **show** コマンドを5つまで割り当てることができます。



(注) 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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **alert-group {Configuration | Diagnostic | EEM | Environmental | Inventory | License | Supervisor-Hardware | Syslog-group-port | System | Test} user-def-cmdshow-cmd**
4. **commit**
5. (任意) **show callhome user-def-cmds**
6. (任意) **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： <pre>switch# configure terminal switch(config)#</pre>	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
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	alert-group {Configuration Diagnostic EEM Environmental Inventory License Supervisor-Hardware Syslog-group-port System Test} user-def-cmd <i>show-cmd</i> 例： 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せん。
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	show callhome user-def-cmds 例： switch(config-callhome)# show callhome user-def-cmds	(任意) 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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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電子メールアドレスも設定できます。

Smart Call Home に対して最大 5 つの 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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **transport email mail-serverip-address [portnumber] [prioritynumber] [use-vrfvrf-name]**
4. (任意) **transport email fromemail-address**
5. (任意) **transport email reply-toemail-address**
6. **commit**
7. (任意) **show callhome transport**
8. (任意) **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	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 : <pre>switch(config)# callhome switch(config-callhome)#</pre>	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-serverip-address [portnumber] [prioritynumber] [use-vrfvrf-name] 例 : <pre>switch(config-callhome)# transport email mail-server 192.0.2.1 use-vrf Red</pre>	<p>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。デフォルトのポート番号は25です。</p> <p>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。</p> <p>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。</p>
ステップ 4	transport email fromemail-address 例 : <pre>switch(config-callhome)# transport email from person@company.com</pre>	(任意) 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。

	コマンドまたはアクション	目的
ステップ 5	transport email reply-toemail-address 例： switch(config-callhome)# transport email reply-to person@company.com	(任意) 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 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 7	show callhome transport 例： switch(config-callhome)# show callhome transport	(任意) 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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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 を使用して、HTTP で Smart 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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **transport http use-vrfvrf-name**
4. **commit**
5. (任意) **show callhome**
6. (任意) **copy running-config startup-config**

手順の詳細

	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。
ステップ 5	show callhome 例： switch(config-callhome)# show callhome	(任意) Smart Call Home に関する情報を表示します。
ステップ 6	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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

手順の概要

1. **configure terminal**
2. **callhome**
3. **transport http proxy serverip-address [portnumber]**
4. **transport http proxy enable**
5. **commit**
6. (任意) **show callhome transport**
7. (任意) **copy running-config startup-config**

手順の詳細

	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 serverip-address [portnumber] 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 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) 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。

	コマンドまたはアクション	目的
ステップ 6	show callhome transport 例： <pre>switch(config-callhome)# show callhome transport</pre>	(任意) 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	copy running-config startup-config 例： <pre>switch(config)# copy running-config startup-config</pre>	(任意) 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。

次の作業

任意で、定期的な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

デバイス上で現在イネーブル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送信するように、デバイスを設定できます。デバイスは、2 種類の Smart Call Home 通知を生成します。定期的コンフィギュレーション メッセージと定期的インベントリ メッセージです。

手順の概要

1. **configure terminal**
2. **callhome**
3. **periodic-inventory notification [intervaldays] [timeofdaytime]**
4. **commit**
5. (任意) **show callhome**
6. (任意) **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： <pre>switch# configure terminal switch(config)#</pre>	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
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	periodic-inventory notification [intervaldays] [timeofdaytime] 例： 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 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	show callhome 例： switch(config-callhome)# show callhome	(任意) Smart Call Home に関する情報を表示します。
ステップ 6	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

任意で重複メッセージ スロットリングをディセーブルにします。

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信す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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **no duplicate-message throttle**
4. **commit**
5. (任意) **copy running-config startup-config**

手順の詳細

	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	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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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 機能をイネーブルにできます。

手順の概要

1. **configure terminal**
2. **callhome**
3. **[no] enable**
4. **commit**
5. (任意) **copy running-config startup-config**

手順の詳細

	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	[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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

次の作業

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。

手順の概要

1. **configure terminal**
2. **callhome**
3. **callhome send [configuration | diagnostic]**
4. **callhome test**
5. (任意) **copy running-config startup-config**

手順の詳細

	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、指定された Smart Call Home テスト メッセージを送信します。
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	copy running-config startup-config 例： switch(config)# copy running-config startup-config	(任意) 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。

コマンド	目的
show callhome destination-profilename	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、連絡先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。

```
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、Smart Call Home メッセージに対して複数の SMTP サーバを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 トリガー

次の表に、イベント トリガーおよび 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	Event Name	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

アラートグループ	Event Name	説明	Smart Call Home 重大度
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
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

アラートグループ	Event Name	説明	Smart Call Home 重大度
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	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
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
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	<p>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、<i>type@Sid@serial</i>。</p> <ul style="list-style-type: none"> • <i>type</i> はバックプレーン IDPROM から取得した製品モデル番号です。 • <i>@</i> は区切り文字です。 • <i>Sid</i> 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。 • <i>serial</i> は、Sid フィールドによって識別される番号です。 <p>例 : N9K-C9508@C@12345678</p>	/aml/ header/deviceId

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）
Customer 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
契約 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
Server ID	<p>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、<i>type@Sid@serial</i>。</p> <ul style="list-style-type: none"> • <i>type</i> はバックプレーン IDPROM から取得した製品モデル番号です。 • <i>@</i> は区切り文字です。 • <i>Sid</i> 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。 • <i>serial</i> は、Sid フィールドによって識別される番号です。 <p>例：N9K-C9508@C@12345678</p>	/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

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）
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
Contact email	このユニットの連絡先として識別される担当者の電子メールアドレス。	/aml/body/sysContactEmail
Contact phone number	このユニットの連絡先である人物の電話番号。	/aml/body/sysContactPhoneNumber
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（製品ファミリー名に含まれる具体的なモデル）	/aml/body/chassis/name
Serial number	ユニットのシャーシのシリアル番号。	/aml/body/chassis/serialNo
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）
コマンド出力名	実行されたCLIコマンドの正確な名前	/aml/attachments/attachment/name
添付タイプ	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

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）
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 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 の名前	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
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

次の表では、フルテキストまたはXML メッセージのインベントリ 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 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 name	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
Process state	プロセスの状態（実行中、中止など）。	/aml/body/process/processState
Process exception	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-C9508C@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_GSYNCFALIED_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]

```

フルテキスト形式での syslog アラート通知の例

```

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">
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_GSYNCFATAL_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&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&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_srvinf: 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">
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">
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ードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html