



## C コマンド

---

この章では、コマンド名が C で始まる Cisco NX-OS システム管理コマンドについて説明します。

# callhome

CLI Call home コンフィギュレーション モードを開始するには、**callhome** コマンドを使用します。

## callhome

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

ディセーブル

### コマンド モード

グローバル コンフィギュレーション

### サポートされるユーザ ロール

network-admin  
vdc-admin

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、CLI Call home コンフィギュレーション モードを開始する例を示します。

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

### 関連コマンド

<b>snmp-server contact</b>	Call home の SNMP サーバの連絡先の名前を指定します。
<b>email-contact</b>	デバイスの担当者の E メール アドレスを指定します。
<b>phone-contact</b>	デバイスの担当者の電話番号を指定します。
<b>streetaddress</b>	デバイスの担当者の所在地住所を指定します。
<b>contract-id</b>	このデバイスのサービス契約の連絡先番号を指定します。
<b>customer-id</b>	このデバイスのサービス契約のカスタマー番号を指定します。
<b>site-id</b>	このデバイスのサイト ID 番号を指定します。
<b>switch-priority</b>	このデバイスのプライオリティ番号を指定します。
<b>destination-profile</b>	Call home 宛先プロファイルを作成し、設定します。
<b>enable</b>	Call home をイネーブルにします。デフォルトでは、Call home はディセーブルです。
<b>callhome test</b>	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。

---

<b>callhome send</b>	設定されたすべての宛先に指定の Call home テスト メッセージを送信します。
<b>show callhome</b>	Call home コンフィギュレーションを表示します。

---

# callhome send

設定されたすべての宛先に Call home メッセージを送信するには、**callhome send** コマンドを使用します。

**callhome send [configuration | diagnostic]**

## 構文の説明

<b>configuration</b>	(任意) 設定メッセージを送信します。
<b>diagnostic</b>	(任意) 診断メッセージを送信します。

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、Call home コンフィギュレーション メッセージを送信する例を示します。

```
switch(config)# callhome send configuration
trying to send configuration callhome message
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>callhome</b>	Call home コンフィギュレーション モードを開始します。
<b>callhome test</b>	設定されたすべての Call home の宛先にテストメッセージを送信します。
<b>show callhome</b>	Call home コンフィギュレーションを表示します。

# callhome test

設定されたすべての Call home の宛先にテスト メッセージを送信するには、**callhome test** コマンドを使用します。

**callhome test {inventory}**

構文の説明	<b>inventory</b>	設定されたすべての Call home の宛先に、ダミーの Call home インベントリを送信します。
-------	------------------	---

デフォルト	なし
-------	----

コマンド モード	任意のコマンド モード
----------	-------------

サポートされるユーザ ロール	network-admin network-operator vdc-admin vdc-operator
----------------	--

コマンド履歴	リリース	変更箇所
	4.0(1)	このコマンドが追加されました。

使用上のガイドライン	このコマンドには、ライセンスは必要ありません。
------------	-------------------------

例	次に、Call home テスト メッセージを送信する例を示します。
---	------------------------------------

```
switch(config)# callhome test
trying to send test callhome message
switch(config)#
```

関連コマンド	コマンド	説明
	<b>callhome</b>	Call home コンフィギュレーション モードを開始します。
	<b>callhome send</b>	設定されたすべての Call home の宛先に、設定メッセージまたは診断メッセージを送信します。
	<b>show callhome</b>	Call home コンフィギュレーションを表示します。

# cdp advertise

デバイスによってサポートされる Cisco Discovery Protocol (CDP) バージョンを設定するには、**cdp advertise** コマンドを使用します。CDP の設定を削除するには、このコマンドの **no** 形式を使用します。

**cdp advertise {v1 | v2}**

**no cdp advertise [v1 | v2]**

## 構文の説明

<b>v1</b>	CDP バージョン 1 を指定します。
<b>v2</b>	CDP バージョン 2 を指定します。

## デフォルト

なし

## コマンドモード

グローバル コンフィギュレーション (config)  
 if-ethernet-all コンフィギュレーション (config-if-ethernet-all)  
 if-gig-ether コンフィギュレーション (config-if-gig-ether)  
 if-eth-base (config-if-eth-base)  
 if-mgmt-ether (config-if-mgmt-ether)

## サポートされるユーザロール

network-admin  
 vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、CDP バージョンを設定する例を示します。

```
switch(config)# cdp advertise v2
switch(config)
```

次に、CDP 設定を削除する例を示します。

```
switch(config)# no cdp advertise v2
switch(config)
```

## 関連コマンド

コマンド	説明
<b>cdp enable</b>	特定のインターフェイス上で CDP をイネーブルにします。

# cdp enable

インターフェイスで Cisco Discovery Protocol (CDP) をイネーブルにするには、**cdp enable** コマンドを使用します。CDP をディセーブルにするには、このコマンドの **no** 形式を使用します。

**cdp enable**

**no cdp enable**

## 構文の説明

このコマンドには、引数またはキーワードはありません。

## デフォルト

なし

## コマンドモード

グローバル コンフィギュレーション (config)

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、インターフェイスで CDP をイネーブルにする例を示します。

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

次に、インターフェイスで CDP をディセーブルにする例を示します。

```
switch(config)# no cdp enable
```

## 関連コマンド

コマンド	説明
<b>cdp advertise</b>	デバイスによってサポートされる CDP バージョンを設定します。

# cdp format device-id

Cisco Discovery Protocol (CDP) のデバイス ID 形式を設定するには、**cdp format device-id** コマンドを使用します。デバイス ID 形式を削除するには、このコマンドの **no** 形式を使用します。

**cdp format device-id {mac-address | serial-number | system-name}**

**no cdp format device-id {mac-address | serial-number | system-name}**

## 構文の説明

<b>mac-address</b>	シャーシの MAC アドレスを指定します。
<b>serial-number</b>	シャーシのシリアル番号または Organizationally Unique Identifier (OUI; 組織固有識別子) を指定します。
<b>system-name</b>	システム名を指定します。デフォルトでは、完全修飾ドメイン名です。

## デフォルト

なし

## コマンドモード

グローバル コンフィギュレーション (config)

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、CDP のデバイス ID 形式を設定する例を示します。

```
switch(config)# cdp format device-id mac-address
switch(config)#
```

次に、デバイス ID 形式を削除する例を示します。

```
switch(config)# no cdp format device-id mac-address
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>cdp enable</b>	CDP のデバイス ID 形式を設定します。

# cdp holdtime

Cisco Discovery Protocol (CDP) が、ネイバー情報をリフレッシュするまでに、それを保持する時間を設定するには、**cdp holdtime** コマンドを使用します。CDP 保持時間を削除するには、このコマンドの **no** 形式を使用します。

**cdp holdtime** *seconds*

**no cdp holdtime** *seconds*

## 構文の説明

*seconds* 秒単位の保持時間。指定できる範囲は 10 ~ 255 です。

## デフォルト

なし

## コマンドモード

グローバル コンフィギュレーション (config)

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、CDP がネイバー情報を保持する時間を設定する例を示します。

```
switch(config)# cdp holdtime 30
switch(config)#
```

次に、CDP 保持時間を削除する例を示します。

```
switch(config)# no cdp holdtime 30
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>cdp timer</b>	CDP のリフレッシュ時間の間隔を設定します。

# cdp timer

シスコ検出プロトコル（CDP）のリフレッシュ時間の間隔を設定するには、**cdp timer** コマンドを使用します。CDP のリフレッシュ時間の間隔を削除するには、このコマンドの **no** 形式を使用します。

**cdp timer seconds**

**no cdp timer seconds**

## 構文の説明

*seconds* 秒単位の時間の間隔。範囲は 5 ～ 254 です。

## デフォルト

なし

## コマンドモード

グローバル コンフィギュレーション (config)

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、CDP のリフレッシュ時間の間隔を設定する例を示します。

```
switch(config)# cdp timer 45
switch(config)#
```

次に、CDP のリフレッシュ時間の間隔を削除する例を示します。

```
switch(config)# no cdp timer 45
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>cdp holdtime</b>	CDP が、ネイバー情報をリフレッシュするまでに、ネイバー情報を保持する時間を設定します。

# cfs distribute

デバイスの Cisco Fabric Services (CFS) 配信をグローバルにイネーブルにするには、**cfs distribute** コマンドを使用します。CFS 配信をディセーブルにするには、コマンドの **no** 形式を使用します。CFS 設定を削除するには、このコマンドの **no** 形式を使用します。

**cfs distribute**

**no cfs distribute**

## 構文の説明

このコマンドには、引数またはキーワードはありません。

## デフォルト

イネーブル

## コマンドモード

グローバル コンフィギュレーション

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

構成情報を配信するためには、デバイスとアプリケーションの両方で CFS 配信がイネーブルにされている必要があります。

デバイスについては、CFS はデフォルトでイネーブルになっています。ファブリック内のすべてのデバイスで CFS をイネーブルに設定しないと配信は受信されません。

アプリケーションで CFS 配信がディセーブルにされている場合、そのアプリケーションは設定を配信せず、またファブリック内の他のデバイスからの配信も受け入れません。

## 例

次に、CFS 配信をイネーブルにする例を示します。

```
switch(config)# cfs distribute
```

## 関連コマンド

<b>show cfs status</b>	CFS 配信ステータスを表示します。
<b>role distribute</b>	CFS をイネーブルにしてロールのコンフィギュレーションを配信します。
<b>show application_name status</b>	アプリケーションで CFS 配信をイネーブルにするかどうかを含め、指定されたアプリケーションのステータスを表示します。
<b>cfs region</b>	CFS 配信スコープを限定するリージョンを指定します。

# cfs eth

Cisco Fabric Services (CFS) がイネーブルのすべてのアプリケーションに変更を配信するために、イーサネットを使用するよう、デバイスをグローバルに設定するには、**cfs eth** コマンドを使用します。CFS 設定を削除するには、このコマンドの **no** 形式を使用します。

**cfs eth {distribute}**

**no cfs eth distribute**

## 構文の説明

**distribute**      イーサネットを介した CFS 配信をイネーブルにします。

## デフォルト

なし

## コマンド モード

グローバル コンフィギュレーション (config)

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、CFS がイネーブルのすべてのアプリケーションに変更を配信するためにイーサネットを使用するよう、デバイスを設定する例を示します。

```
switch(config)# cfs eth distribute
switch(config)#
```

次に、CFS 設定を削除する例を示します。

```
switch(config)# no cfs eth distribute
This will prevent CFS from distributing over Ethernet network.
Are you sure? (y/n) n
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>cfs distribute</b>	デバイスについて、CFS 配信をグローバルにイネーブルにします。
<b>show cfs status</b>	CFS 配信ステータスを表示します。

# cfs ipv4

Cisco Fabric Services (CFS) がイネーブルのすべてのアプリケーションに変更を配信するために、IPv4 を使用するよう、デバイスをグローバルに設定するには、**cfs ipv4** コマンドを使用します。CFS 設定を削除するには、このコマンドの **no** 形式を使用します。

**cfs ipv4 [mcast | distribute]**

**no cfs ipv4 [mcast | distribute]**

## 構文の説明

<b>mcast</b>	(任意) 設定の変更が配信される IPv4 マルチキャスト アドレスを設定します。
<b>distribute</b>	(任意) CFS がイネーブルのアプリケーションで変更を配信するために、IPv4 を使用するよう、デバイスを設定します。

## デフォルト

デフォルトの IPv4 マルチキャスト アドレスは 239.255.70.83 です。

## コマンドモード

グローバル コンフィギュレーション

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

CFS は、同じデバイスから IPv4 と IPv6 両方を介して配信できません。

構成情報を配信するためには、デバイスとアプリケーションの両方で CFS 配信がイネーブルにされている必要があります。

デバイスについては、CFS はデフォルトでイネーブルになっています。ファブリック内のすべてのデバイスで CFS をイネーブルに設定しないと配信は受信されません。

アプリケーションで CFS 配信がディセーブルにされている場合、そのアプリケーションは設定を配信せず、またファブリック内の他のデバイスからの配信も受け入れません。

マルチキャスト アドレスを変更する前に、IP を介した CFS をディセーブルにする必要があります。

## 例

次に、IPv4 を介した CFS 配信を、まずディセーブルにし、設定の変更が配信される IPv4 マルチキャスト アドレスを設定する例を示します。

```
switch(config)# no cfs ipv4 distribute
This will prevent CFS from distributing over IPv4 network.

Are you sure? (y/n) [n] y
```

```

switch(config)# cfs ipv4 mcast-address 239.255.1.1

Distribution over this IP type will be affected

Change multicast address for CFS-IP ?

Are you sure? (y/n) [n] y

```

---

**関連コマンド**

<b>cfs distribute</b>	デバイスについて、CFS 配信をグローバルにイネーブルにします。
<b>cfs</b>	CFS 配信モードを指定します。
<b>show cfs status</b>	CFS 配信ステータスを表示します。
<i>application_name</i> <b>distribute</b>	RADIUS などの、指定されたアプリケーションへの配信をイネーブルにします。
<b>show application_name status</b>	アプリケーションで CFS 配信をイネーブルにするかどうかを含め、RADIUS などの、指定されたアプリケーションのステータスを表示します。

# cfs region

アプリケーションの配信スコープを限定する、Cisco Fabric Services (CFS) リージョンを作成するには、**cfs region** コマンドを使用します。リージョンまたはアプリケーションを削除するには、このコマンドの **no** 形式を使用します。

**cfs region** *region\_id*

*application\_name*

**no cfs region**

**no** *application\_name*

## 構文の説明

<i>region_id</i>	番号 0 ~ 200 によって識別される CFS リージョン。リージョン 0 はデフォルトリージョンで、別のリージョンに割り当てられていないファブリックの各デバイスが含まれています。ユーザは、リージョン番号 1 ~ 200 を設定できます。
<i>application_name</i>	CFS 配信で指定されたリージョンに割り当てるアプリケーション。

## デフォルト

デフォルトのリージョン ID は 0 です。

## コマンドモード

グローバル コンフィギュレーション  
CFS リージョン設定

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

## 使用上のガイドライン

機能を移動する、つまり、機能を新規のリージョンに割り当てた場合は、範囲はこの新規のリージョンに制限され、配信または結合のためのリージョンはすべて無視されます。

複数のアプリケーションの設定を配信するように CFS リージョンを設定できます。ただし、特定のスイッチでは、一度に特定のアプリケーション設定を配信するように設定できる CFS リージョンは 1 つだけです。

アプリケーションを CFS リージョンに割り当てた場合、この設定を別の CFS リージョン内に配信できません。

リージョンからアプリケーションを削除する場合で、異なるリージョンに割り当てない場合、デフォルトリージョンのリージョン 0 に追加されます。

1つのアプリケーションを複数回同じリージョンに追加しようとした場合、次のメッセージが表示されます。

```
Application already present in the same region.
```

構成情報を配信するためには、デバイスとアプリケーションの両方で CFS 配信がイネーブルにされている必要があります。

デバイスについては、CFS はデフォルトでイネーブルになっています。ファブリック内のすべてのデバイスで CFS をイネーブルに設定しないと配信は受信されません。

アプリケーションで CFS 配信がディセーブルにされている場合、そのアプリケーションは設定を配信せず、またファブリック内の他のデバイスからの配信も受け入れません。

このコマンドには、ライセンスは必要ありません。

## 例

次に、リージョン 4 を作成し、それに NTP アプリケーションを追加する例を示します。ユーザがリージョンを作成するときに、CLI によって、ユーザがそのリージョンのコンフィギュレーション モードに置かれ、そこで、ユーザは、アプリケーションを追加できます。

```
switch(config)# cfs region 4
switch(config-cfs-region)# callhome
switch(config-cfs-region)# show cfs region brief
```

```
-----
Region          Application  Enabled
-----
4                ntp         no
4                callhome   no
6                igmp       yes
6                radius     no
```

```
switch(config-cfs-region)#
```

## 関連コマンド

<b>show cfs region</b>	デバイスに設定された CFS 配信リージョンを表示します。
<b>show cfs status</b>	CFS 配信ステータスを表示します。
<b>application_name distribute</b>	NTP などの、指定されたアプリケーションへの配信をイネーブルにします。
<b>show application_name status</b>	アプリケーションで CFS 配信をイネーブルにするかどうかを含め、NTP などの、指定されたアプリケーションのステータスを表示します。

# check logflash

compactFlash メモリ カードをチェックするには、**check logflash** コマンドを使用します。

**check logflash [bad-blocks]**

構文の説明	<b>bad-blocks</b> (任意) compactFlash メモリ カードの不良ブロックを検出します。				
デフォルト	なし				
コマンド モード	任意のコマンド モード				
サポートされるユーザ ロール	network-admin network-operator vdc-admin vdc-operator				
コマンド履歴	<table border="1"> <thead> <tr> <th>リリース</th> <th>変更箇所</th> </tr> </thead> <tbody> <tr> <td>4.0(3)</td> <td>このコマンドが追加されました。</td> </tr> </tbody> </table>	リリース	変更箇所	4.0(3)	このコマンドが追加されました。
リリース	変更箇所				
4.0(3)	このコマンドが追加されました。				
使用上のガイドライン	このコマンドには、ライセンスは必要ありません。				
例	次に、compactFlash メモリ カードをチェックする例を示します。 switch# <b>check logflash</b>				

# checkpoint

ロールバック チェックポイントを設定するには、**checkpoint** コマンドを使用します。チェックポイントを削除するには、このコマンドの **no** 形式を使用します。

**checkpoint** {*name* | **description** *description* | **file** *name*}

**no checkpoint** *name*

## 構文の説明

<i>name</i>	(任意) チェックポイント データベースで使用されるチェックポイント名。名前は、最大 80 文字の英数字のストリングで指定できますが、空白文字を含めることはできません。
<b>description</b> <i>description</i>	(任意) 指定されたチェックポイントのチェックポイント説明を指定します。 <b>description</b> には、スペースも含めて最大 80 文字の英数字を指定できます。
<b>file</b> <i>name</i>	(任意) チェックポイントの保存に使用されるファイル名を指定します。

## デフォルト

なし

## コマンド モード

任意のコマンド モード

## サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

チェックポイント ファイル名を指定せずに **checkpoint** コマンドを実行すると、**auto-x** (x は無名のチェックポイント ファイルを作成するたびに増分される 10 進数) という名前のファイルが作成されません。

このコマンドには、ライセンスは必要ありません。

## 例

次に、ロールバック チェックポイントを設定する例を示します。

```
switch# checkpoint stable
switch#
```

次に、チェックポイント ファイルを削除する例を示します。

```
switch# no checkpoint stable
switch#
```

---

**関連コマンド**

コマンド	説明
<b>clear checkpoint database</b>	チェックポイント ファイルの内容を表示します。

# clear callhome session

Call home Cisco Fabric Services (CFS) 配信セッションをクリアするには、**clear callhome session** コマンドを使用します。

## clear callhome session

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、Call home CFS 配信セッションをクリアする例を示します。

```
switch(config)# clear callhome session
```

### 関連コマンド

コマンド	説明
<b>callhome</b>	Call home コンフィギュレーションモードを開始します。
<b>callhome send</b>	設定されたすべての Call home の宛先に、設定メッセージまたは診断メッセージを送信します。
<b>show callhome</b>	Call home コンフィギュレーションを表示します。

# clear cdp

インターフェイスで Cisco Discovery Protocol (CDP) の統計情報をクリアするには、**clear cdp** コマンドを使用します。

```
clear cdp {counters [interface interface] | table [interface interface]}
```

## 構文の説明

<b>counters</b>	すべてのインターフェイスで CDP カウンタをクリアします。
<b>interface</b> <i>interface</i>	(任意) インターフェイスの CDP カウンタをクリアします。
<b>table</b>	すべてのインターフェイスで CDP キャッシュをクリアします。

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、インターフェイスで CDP 統計情報をクリアする例を示します。

```
switch(config)# clear cdp counters  
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>enable cdp</b>	特定のインターフェイス上で CDP をイネーブルにします。

# clear checkpoint database

データベース内のチェックポイント ファイルをすべて削除するには、**clear checkpoint database** コマンドを使用します。

## clear checkpoint database

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンド モード

任意のコマンド モード

### サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、チェックポイント データベースをクリアする例を示します。

```
switch# clear checkpoint database
Processing the Request... Please Wait
..... Done
switch#
```

### 関連コマンド

コマンド	説明
<b>show checkpoint</b>	チェックポイント ファイルの内容を表示します。

# clear cores

コア ファイルをクリアするには、**clear cores** コマンドを使用します。

## clear cores [archive]

構文の説明	<b>archive</b> (任意) logflash ファイル システム上のコア ファイルをクリアします。
-------	---

デフォルト	なし
-------	----

コマンド モード	任意のコマンド モード
----------	-------------

サポートされるユーザ ロール	network-admin network-operator vdc-admin vdc-operator
----------------	--

コマンド履歴	リリース	変更箇所
	4.0(1)	このコマンドが追加されました。

使用上のガイドライン	コア ファイルに関する情報を表示するには、 <b>show system cores</b> コマンドを使用します。 このコマンドには、ライセンスは必要ありません。
------------	---

例	次に、コア ファイルをクリアする例を示します。
---	-------------------------

```
switch# clear cores
```

次に、logflash ファイル システム上にあるコア ファイルをクリアする例を示します。

```
switch# clear cores archive
```

関連コマンド	コマンド	説明
	<b>show system cores</b>	コア ファイル名を表示します。
	<b>system cores</b>	コア ファイル名を設定します。

# clear flow exporter

Flexible NetFlow フロー エクスポートの統計情報をクリアするには、**clear flow exporter** コマンドを使用します。

```
clear flow exporter {name exporter-name | exporter-name}
```

## 構文の説明

<b>name</b>	フロー エクスポートの名前を指定します。
<i>exporter-name</i>	既存のフロー エクスポートの名前

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**clear flow exporter** コマンドを使用するには、事前に、エクスポートを使用して Flexible NetFlow によるトラフィック モニタリングをイネーブリングしておく必要があります。

このコマンドには、ライセンスは必要ありません。

## 例

次に、NFC-DC-PHOENIX という名前のフロー エクスポートの統計情報をクリアする例を示します。

```
switch# clear flow exporter name NFC-DC-PHOENIX
switch#
```

## 関連コマンド

コマンド	説明
<b>clear flow exporter</b>	エクスポートの統計情報をクリアします。
<b>flow exporter</b>	フロー エクスポートを作成します。
<b>show flow exporter</b>	フロー エクスポートのステータスと統計情報を表示します。

# clear flow monitor

Flexible NetFlow フロー モニタ、フロー モニタ キャッシュ、またはフロー モニタ統計情報をクリアし、フロー モニタ キャッシュ内のデータを強制的にエクスポートするには、**clear flow monitor** コマンドを使用します。

```
clear flow monitor {name monitor-name | monitor-name} [cache [force-export] | statistics]
```

## 構文の説明

<b>name</b>	フロー モニタの名前を指定します。
<i>monitor-name</i>	既存のフロー モニタの名前
<b>cache</b>	(任意) フロー モニタ キャッシュ情報をクリアします。
<b>force-export</b>	(任意) フロー モニタ キャッシュ統計情報を強制的にエクスポートします。
<b>statistics</b>	(任意) フロー モニタの統計情報をクリアします。

## デフォルト

なし

## コマンド モード

任意のコマンド モード

## サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**clear flow monitor** コマンドを使用するには、事前に Flexible NetFlow モニタをイネーブルにしておく必要があります。

フロー モニタ キャッシュからすべてのエントリを削除するには、**clear flow monitor monitor-name cache** コマンドを使用します。キャッシュ内のエントリはエクスポートされ、キャッシュ内に保存されていたデータは失われます。

クリアされたキャッシュ エントリの統計情報は保持されます。

フロー モニタ キャッシュからすべてのエントリを削除し、フロー モニタに割り当てられているすべてのフロー エクスポートにそれをエクスポートするには、**clear flow monitor monitor-name force-export** コマンドを使用します。この処理により、CPU の使用率が一時的に増大します。



### 注意

**clear flow monitor monitor-name force-export** コマンドを使用すると、CPU 使用率が短期間増加する可能性があるため、このコマンドを使用する場合は注意してください。

このフロー モニタの統計情報とキャッシュ エントリをクリアするには、**clear flow monitor monitor-name statistics** コマンドを使用します。

この統計情報では、キャッシュにあるエントリ数が示されるため、[Current entries] 統計情報はクリアされません。

このコマンドには、ライセンスは必要ありません。

## 例

次に、NFC-DC-PHOENIX という名前のフロー モニタの統計情報とキャッシュ エントリをクリアする例を示します。

```
switch# clear flow monitor name NFC-DC-PHOENIX
switch#
```

次に、NFC-DC-PHOENIX という名前のフロー モニタの統計情報とキャッシュ エントリをクリアして、強制的にエクスポートする例を示します。

```
switch# clear flow monitor NFC-DC-PHOENIX force-export
switch#
```

次に、NFC-DC-PHOENIX という名前のフロー モニタのキャッシュをクリアして、強制的にエクスポートする例を示します。

```
switch# clear flow monitor NFC-DC-PHOENIX cache force-export
switch#
```

次に、NFC-DC-PHOENIX という名前のフロー モニタの統計情報をクリアする例を示します。

```
switch# clear flow monitor NFC-DC-PHOENIX statistics
switch#
```

## 関連コマンド

コマンド	説明
<b>clear flow monitor</b>	フロー モニタをクリアします。
<b>flow monitor</b>	フロー モニタを作成します。
<b>show flow sw-monitor</b>	フロー モニタのステータスと統計情報を表示します。

# clear hardware flow ip

NetFlow ハードウェア IP フローをクリアするには、**clear hardware flow ip** コマンドを使用します。

```
clear hardware flow ip [{vdc vdc_id} | {monitor name} | {profile profile-id} | {vlan
vlan-id} | {interface if-type if-number}] [instance inst] [force-export] [module num]
```

## 構文の説明

<b>vdc</b> <i>vdc_id</i>	VDC を指定します。指定できる範囲は 1 ~ 16 です。
<b>monitor</b> <i>name</i>	NetFlow フロー モニタの名前を指定します。モニタ名は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
<b>profile</b> <i>profile-id</i>	フロー プロファイルの名前を指定します。指定できる範囲は 1 ~ 31 です。
<b>vlan</b> <i>vlan-id</i>	VLAN を設定します。有効な範囲は 1 ~ 4094 です。
<b>interface</b>	インターフェイスを指定します。
<i>if-type</i>	インターフェイス タイプ。詳細については、疑問符 (?) オンライン ヘルプ機能を使用します。
<i>if-number</i>	インターフェイスまたはサブインターフェイスの番号です。ネットワーキング デバイスに対する番号付け構文の詳細については、疑問符 (?) のオンライン ヘルプ機能を使用してください。
<b>instance</b> <i>inst</i>	(任意) EARL インスタンスを指定します。EARL インスタンスには 32 文字以内の英数字のストリングを指定します。
<b>force-export</b>	(任意) クリア操作の前に、データをコレクタに強制的にエクスポートします。
<b>module</b> <i>num</i>	(任意) モジュールを指定します。

## デフォルト

なし

## コマンド モード

任意のコマンド モード

## サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、NetFlow ハードウェア IP フローをクリアする例を示します。

## ■ clear hardware flow ip

```
switch(config)# clear hardware flow ip module 8 ethernet 2/1  
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>show hardware flow {ip   ipv6}</b>	NetFlow ハードウェア IP/IPV6 フローに関する情報を表示します。

# clear logging ip access-list cache

Optimized ACL Logging (OAL) キャッシュからすべてのエントリをクリアし、Syslog に送信するには、**clear logging ip access-list cache** コマンドを使用します。

## clear logging ip access-list cache

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、OAL キャッシュからすべてのエントリをクリアし、Syslog に送信する例を示します。

```
switch# clear logging ip access-list cache  
switch#
```

### 関連コマンド

コマンド	説明
<b>show logging ip access-list</b>	IP アクセス リストのロギング ステータスを表示します。

# clear logging logfile

ロギング ファイルからのメッセージをクリアするには、**clear logging logfile** コマンドを使用します。

## clear logging logfile

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンド モード

任意のコマンド モード

### サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、ロギング ファイルからメッセージをクリアする例を示します。

```
switch# clear logging logfile
switch#
```

### 関連コマンド

コマンド	説明
<b>show logging logfile</b>	ローカル ログ ファイル内のログを表示します。

# clear logging nvram

NVRAM ログをクリアするには、**clear logging nvram** コマンドを使用します。

## clear logging nvram

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、NVRAM ログをクリアする例を示します。

```
switch# clear logging nvram
switch#
```

### 関連コマンド

コマンド	説明
show logging nvram	NVRAM ログを表示します。

# clear logging onboard

永続ログの Onboard Failure Logging (OBFL; オンボード障害ロギング) エントリをクリアするには、**clear logging onboard** コマンドを使用します。

**clear logging onboard** [**counter-stats**] [**environmental-history**] [**error-stats**]  
[**exception-log**] [**interrupt-stats**] [**module num**] [**obfl-log**] [**stack-trace**]

## 構文の説明

<b>counter-stats</b>	(任意) OBFL カウンタ統計情報をクリアします。
<b>environmental-history</b>	(任意) OBFL 環境履歴をクリアします。
<b>error-stats</b>	(任意) OBFL エラー統計情報をクリアします。
<b>exception-log</b>	(任意) OBFL 例外ログのエントリをクリアします。
<b>interrupt-stats</b>	(任意) OBFL 割り込み統計情報をクリアします。
<b>module num</b>	(任意) 特定のモジュールの OBFL 情報をクリアします。
<b>obfl-log</b>	(任意) OBFL (boot-uptime/device-version/obfl-history) をクリアします。
<b>stack-trace</b>	(任意) OBFL スタック トレースのエントリをクリアします。

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。
4.0(2)	<b>counter-stats</b> キーワードが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、OBFL 環境履歴のエントリをクリアする例を示します。

```
switch# clear logging onboard environmental-history
switch#
```

次に、OBFL エラー統計情報をクリアする例を示します。

```
switch# clear logging onboard error-stats
switch#
```

次に、OBFL 例外ログのエントリをクリアする例を示します。

```
switch# clear logging onboard exception-log  
switch#
```

次に、OBFL 割り込み統計情報をクリアする例を示します。

```
switch# clear logging onboard interrupt-stats  
switch#
```

次に、特定のモジュールの OBFL 情報をクリアする例を示します。

```
switch# clear logging onboard module 2  
switch#
```

次に、OBFL (boot-uptime/device-version/obfl-history) のエントリをクリアする例を示します。

```
switch# clear logging onboard obfl-log  
switch#
```

次に、OBFL スタック トレースのエントリをクリアする例を示します。

```
switch# clear logging onboard stack-trace  
switch#
```

## 関連コマンド

コマンド	説明
<b>hw-module logging onboard</b>	エラー タイプに基づいて OBFL をイネーブルにします。
<b>show logging onboard</b>	オンボード失敗ログを表示します。

# clear logging session

現在のロギングセッションをクリアするには、**clear logging session** コマンドを使用します。

## clear logging session

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、現在のロギングセッションをクリアする例を示します。

```
switch# clear logging session
switch#
```

### 関連コマンド

コマンド	説明
<b>show logging session</b>	ロギングセッションのステータスを表示します。

# clear ntp session

Network Time Protocol (NTP; ネットワーク タイム プロトコル) セッションをクリアするには、**clear ntp session** コマンドを使用します。

## clear ntp session

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンド モード

任意のコマンド モード

### サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.2(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、NTP セッションをクリアする例を示します。

```
switch(config)# clear ntp session
```

# clear ntp statistics

ネットワーク タイム プロトコル (NTP) 統計情報をクリアするには、**clear ntp statistics** コマンドを使用します。

**clear ntp statistics {all-peers | io | local | memory}**

## 構文の説明

<b>all-peers</b>	すべての NTP ピアの統計情報をクリアします。
<b>io</b>	IO 統計情報をクリアします。
<b>local</b>	ローカル統計情報をクリアします。
<b>memory</b>	メモリ統計情報をクリアします。

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、すべての NTP ピアの統計情報をクリアする例を示します。

```
switch(config)# clear ntp statistics all-peers
```

## 関連コマンド

コマンド	説明
<b>show ntp peers</b>	NTP ピアに関する情報を表示します。

# clear nvram

NVRAM をクリアするには、**clear nvram** コマンドを使用します。

## clear nvram

### 構文の説明

このコマンドには、キーワードと引数はありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、NVRAM をクリアする例を示します。

```
switch(config)# clear nvram
```

# clear platform flow ip

NetFlow ハードウェア IPv4 エントリをクリアするには、**clear platform flow ip** コマンドを使用します。

**clear platform flow ip** [*type*] [**force-export**] [**module** *mod-num*]

## 構文の説明

<i>type</i>	(任意) クリアするエントリのタイプ。有効な値については、「使用上のガイドライン」の項を参照してください。
<b>force-export</b>	(任意) クリアしたデータをコレクタに強制的にエクスポートするように指定します。
<b>module</b> <i>mod-num</i>	(任意) モジュールを指定します。モジュール番号の範囲は、使用しているシャーシによって異なります。

## デフォルト

エントリのタイプを省略すると、すべてのタイプのエントリがクリアされます。

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、クリアしたデータをコレクタに強制的にエクスポートするように指示する例を示します。

```
switch# clear platform flow ip forced-export
switch#
```

次に、モジュールの NetFlow 統計情報をクリアする例を示します。

```
switch# clear platform flow ip module 2
switch#
```

## 関連コマンド

コマンド	説明
<b>flow exporter</b>	フロー エクスポートを作成します。
<b>clear flow monitor</b>	フロー モニタをクリアします。
<b>flow monitor</b>	フロー モニタを作成します。
<b>show flow sw-monitor</b>	フロー モニタのステータスと統計情報を表示します。

# clear processes log archive

ログ フラッシュのログ ファイルを削除するには、**clear processes log archive** コマンドを使用します。

**clear processes log archive** [*file file-number*]

構文の説明	<b>file file-number</b> (任意) 削除するログ フラッシュのログ ファイルを指定します。
-------	--

デフォルト	なし
-------	----

コマンド モード	任意のコマンド モード
----------	-------------

サポートされるユーザ ロール	network-admin network-operator vdc-admin vdc-operator
----------------	--

コマンド履歴	リリース	変更箇所
	4.0(1)	このコマンドが追加されました。

使用上のガイドライン	このコマンドには、ライセンスは必要ありません。
------------	-------------------------

例	次に、ログ フラッシュのログ ファイルを削除する例を示します。
---	---------------------------------

```
switch(config)# clear processes log archive
switch(config)#
```

関連コマンド	コマンド	説明
	<b>show processes log</b>	プロセス ログの内容を表示します。

# clear rmon

簡易ネットワーク管理プロトコル (SNMP) 通知から Remote Network Monitoring (RMON) テーブルを削除するには、**clear rmon** コマンドを使用します。

**clear rmon {alarms | events | hcalarms | all-alarms}**

## 構文の説明

<b>alarms</b>	すべての 32 ビット アラームをクリアします。
<b>events</b>	RMON ログをクリアし、RMON イベント テーブルもクリアします。
<b>hcalarms</b>	すべての 64 ビット RMON アラームをクリアします。
<b>all-alarms</b>	すべての 32 ビット RMON アラームと 64 ビット RMON アラームをクリアします。

## デフォルト

なし

## コマンド モード

任意のコマンド モード

## サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、RMON テーブルを削除する例を示します。

```
switch(config)# clear rmon alarms
switch(config)#
```

## 関連コマンド

コマンド	説明
<b>clear snmp counters</b>	SNMP カウンタを削除します。

# clear session state name

セッションの状態情報をクリアするには、**clear session state name** コマンドを使用します。

**clear session state name** *name*

## 構文の説明

<i>name</i>	セッションの名前。63 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
-------------	---

## デフォルト

なし

## コマンドモード

任意のコマンドモード

## サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、コンフィギュレーションセッションの内部状態をクリアする例を示します。

```
switch# clear session state name myACLs
```

## 関連コマンド

コマンド	説明
<b>show configuration session</b>	コンフィギュレーションセッションに関する情報を表示します。

# clear snmp counters

Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) カウンタを削除するには、**clear snmp counters** コマンドを使用します。

## clear snmp counters

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、SNMP カウンタを削除する例を示します。

```
switch(config)# clear snmp counters  
switch(config)#
```

### 関連コマンド

コマンド	説明
show snmp sessions	SNMP セッションに関する情報を表示します。

# clear snmp hostconfig

簡易ネットワーク管理プロトコル（SNMP）ホスト設定を削除するには、**clear snmp hostconfig** コマンドを使用します。

## clear snmp hostconfig

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、SNMP ホスト設定を削除する例を示します。

```
switch(config)# clear snmp hostconfig
switch(config)#
```

### 関連コマンド

コマンド	説明
<b>show snmp sessions</b>	SNMP セッションを表示します。
<b>clear snmp counters</b>	SNMP カウンタを削除します。

# clear system reset-reason

デバイスのリセット理由履歴をクリアするには、**clear system reset-reason** コマンドを使用します。

## clear system reset-reason

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

任意のコマンドモード

### サポートされるユーザロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

### 例

次に、デバイスのリセット理由履歴をクリアする例を示します。

```
switch# clear system reset-reason
```

### 関連コマンド

コマンド	説明
show system reset-reason	デバイスのリセット理由履歴を表示します。

# collect counter

フローの確認されたバイト数またはパケット数を非キー フィールドとして設定し、Flexible NetFlow フロー レコードのカウンタ値（バイト数またはパケット数）を収集するには、**collect counter** コマンドを使用します。フロー内のバイト数またはパケット数を Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

**collect counter {bytes [long] | packets [long]}**

**no collect counter {bytes [long] | packets [long]}**

## 構文の説明

<b>bytes</b>	フローの確認されたバイト数を非キー フィールドとして設定し、フローの合計バイト数を収集します。
<b>long</b>	(任意) 64 ビットのカウンタを使用してフロー合計バイト数を収集します。
<b>packets</b>	フローの確認されたバイト数を非キー フィールドとして設定し、フローの合計パケット数を収集します。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンド モード

フロー レコード コンフィギュレーション

## サポートされるユーザ ロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。**nonkey** フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されません。**nonkey** フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

フロー内パケットが確認されるたびに増分される 32 ビットのカウンタを設定するには、**collect counter packets** コマンドを使用します。非常に長いフローの場合は、カウンタが上限値（約 40 億パケット）を超えてラップする可能性があります。フロー モニタによって、ラップが起こりそうなシナリオが検出されると、通常キャッシュ タイプを使用したフロー モニタでは、フローがエクスポートされて、新しいフローが開始されます。

フロー内パケットが確認されるたびに増分される 64 ビットのカウンタを設定するには、**collect counter packets long** コマンドを使用します。64 ビットのカウンタがラップすることはまず考えられません。

このコマンドには、ライセンスは必要ありません。

## 例

次に、フローの合計バイト数を非キー フィールドとして収集するように設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect counter bytes
```

次に、フローの合計バイト数を 64 ビット カウンタを使用した非キー フィールドとして収集するように設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect counter bytes long
```

次に、フローの合計パケット数を非キー フィールドとして収集するように設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect counter packets
```

次に、フローの合計パケット数を 64 ビット カウンタを使用した非キー フィールドとして収集するように設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect counter packets long
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect interface</b>	入力および（または）出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>debug flow record</b>	フロー レコードのデバッグ出力をイネーブルにします。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス（interface フィールド）に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポート フィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# collect flow

フロー方向またはフロー サンプラ ID 番号を非キー フィールドとして設定し、任意の Flexible NetFlow フロー レコードについてその値を収集するには、**collect flow** コマンドを使用します。フロー方向またはフロー サンプラ ID 番号を Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

**collect flow** {**direction** | **sampler**}

**no collect flow** {**direction** | **sampler**}

## 構文の説明

<b>direction</b>	フロー方向を非キー フィールドとして設定し、フローがモニタされた方向を収集します。
<b>sampler</b>	フロー サンプラ ID を非キー フィールドとして設定し、フロー モニタに割り当てられたサンプラの ID を収集します。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンドモード

フロー レコード コンフィギュレーション

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。**nonkey** フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されません。**nonkey** フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

フローの方向を示すには、**collect flow direction** コマンドを使用します。このコマンドは、単一のフロー モニタが入力と出力の両フロー用に設定されているときに、入力と出力で 1 回ずつ、計 2 回モニタされたフローを検出して排除する場合に、使用します。

フローのモニタに使用されるフロー サンプラの ID を収集するには、**collect flow sampler** コマンドを使用します。このコマンドは、異なるサンプリング レートのフロー サンプラを複数使用している場合に、使用します。**option sampler-table** コマンドを使用すると、コレクタが各フローについてスケールされたカウンタを計算できるようにフロー サンプラ ID をサンプリング レートにマッピングしたうえで、オプション レコードをエクスポートできます。

このコマンドには、ライセンスは必要ありません。

## 例

次に、非キー フィールドとしてモニタされたフロー非キーの方向を設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect flow direction
```

次に、フローに割り当てられたフロー サンプラ ID を非キー フィールドとして設定し、フロー サンプラの ID を収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect flow sampler
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect interface</b>	入力および（または）出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス（interface フィールド）に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポートフィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# collect interface

入力および（または）出力インターフェイスを非キー フィールドとして設定し、任意の Flexible NetFlow フロー レコードについてその値を収集するには、**collect interface** コマンドを使用します。入力または出力インターフェイスを Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

**collect interface {input | output}**

**no collect interface {input | output}**

## 構文の説明

<b>input</b>	入力インターフェイスを非キー フィールドとして設定し、フローから入力インターフェイスを収集します。
<b>output</b>	出力インターフェイスを非キー フィールドとして設定し、フローから出力インターフェイスを収集します。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンド モード

フロー レコード コンフィギュレーション

## サポートされるユーザ ロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。**nonkey** フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されません。**nonkey** フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

このコマンドには、ライセンスは必要ありません。

## 例

次に、入力インターフェイスを非キー フィールドとして設定し、入力インターフェイス値を収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect interface input
```

次に、入力インターフェイスを非キー フィールドとして設定し、出力インターフェイス値を収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
```

```
switch(config-flow-record)# collect interface output
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect interface</b>	入力または出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス ( <b>interface</b> フィールド) に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポート フィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# collect routing

ルーティング属性を非キー フィールドとして設定し、任意の Flexible NetFlow フロー レコードについてそのフィールド値を収集するには、**collect routing** コマンドを使用します。ルーティング属性を Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

```
collect routing {{destination | source} as [peer] | traffic-index | forwarding-status |
next-hop address ipv4 [bgp]}
```

```
no collect routing {{destination | source} as [peer] | traffic-index | forwarding-status |
next-hop address ipv4 [bgp]}
```

## 構文の説明

<b>destination</b>	1 つ以上の宛先ルーティング属性フィールドを非キー フィールドとして設定し、フローからそれらの値を収集します。
<b>source</b>	1 つ以上の送信元ルーティング属性フィールドを非キー フィールドとして設定し、フローからそれらの値を収集します。
<b>as</b>	宛先 AS フィールドを非キー フィールドとして設定し、フローから AS フィールドの値を収集します。
<b>peer</b>	(任意) ピア ネットワークの宛先 AS 番号を非キー フィールドとして設定し、フローからピア ネットワークの AS 番号の値を収集します。
<b>traffic-index</b>	ボーダー ゲートウェイ プロトコル (BGP) 送信元または宛先トラフィック インデックスを非キー フィールドとして設定し、フローから BGP 宛先トラフィック インデックスの値を収集します。
<b>forwarding-status</b>	パケットのフォワーディング ステータスを収集し、アクセス コントロール リスト (ACL) エントリによって拒否されたフローの収集をトリガーします。
<b>next-hop address ipv4</b>	ネクストホップ値を非キー フィールドとして設定し、フローからネクストホップ情報を収集します。
<b>bgp</b>	(任意) ネクストホップ BGP ネットワークの IP アドレスを非キー フィールドとして設定し、フローからネクストホップ BGP ネットワークの IP アドレス値を収集します。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンド モード

フロー レコード コンフィギュレーション

## サポートされるユーザーロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。nonkey フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されます。nonkey フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

送信元 IP アドレスを使用してルータのルーティング テーブルを検索し、その結果に基づいて 16 ビットの AS 番号を収集するには、**collect routing source as [peer]** コマンドを使用します。オプションの **peer** キーワードを指定すると、送信元ネットワークではなく、その次のネットワークが出力されます。



(注)

16 ビットの AS 番号は、このルータからパケットが逆向きにルーティングされる方法に基づいており、

非対称経路では正しい結果が得られないことがあります。

宛先 IP アドレスを使用してルータのルーティング テーブルを検索し、その結果に基づいて 16 ビットの AS 番号を収集するには、**collect routing destination as [peer]** コマンドを使用します。オプションの **peer** キーワードを指定すると、宛先ネットワークではなく、その次のネットワークが出力されます。

このフローの送信元 AS に基づいてトラフィック インデックス フィールドを収集するには、**collect routing source traffic-index** コマンドを使用します。トラフィック インデックス フィールドは、BGP によって伝播する値です。

パケットが正常に転送された場合に示すフィールドを収集するには、**collect routing forwarding-status** コマンドを使用します。このフィールドは 2 つの部分で構成され、最大長は 4 バイトです。現時点では、ステータス フィールドだけが使用されます。

```

+-+-----+-----+
| S | Reason   | |
| t | codes   |
|   | a | or   |
| t | flags   |
| u |         |
| s |         |
+-+-----+-----+
0 1 2 3 4 5 6 7

```

```

Status:
00b=Unknown, 01b = Forwarded, 10b = Dropped, 11b = Consumed

```

このコマンドには、ライセンスは必要ありません。

## 例

次に、送信元 IP アドレスを非キー フィールドとしてルータのルーティング テーブルを検索した結果に基づいて 16 ビットの AS 番号を設定し、その 16 ビット AS 番号を収集する例を示します。

```

switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect routing source as

```

次に、宛先 IP アドレスを非キー フィールドとしてルータのルーティング テーブルを検索した結果に基づいて 16 ビットの AS 番号を設定し、その 16 ビット AS 番号を収集する例を示します。

```

switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect routing destination as

```

非キー フィールドとしてのフローの送信元 AS に基づいてトラフィック インデックス フィールドに値を設定し、その値をトラフィック インデックス フィールド値として収集する例を示します。

```

switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect routing source traffic-index

```

次に、フォワーディング ステートを非キー フィールドとして設定し、そのフォワーディング ステート 値を収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect routing forwarding-status
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect interface</b>	入力または出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス (interface フィールド) に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポート フィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# collect timestamp sys-uptime

TIMESTAMP SYS-UPTIME フィールドを非キー フィールドとして設定し、任意の Flexible NetFlow フロー レコードについてそのフィールド値を収集するには、**collect timestamp sys-uptime** コマンドを使用します。TIMESTAMP SYS-UPTIME フィールドを Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

**collect timestamp sys-uptime {first | last}**

**no collect timestamp sys-uptime {first | last}**

## 構文の説明

<b>first</b>	フローの最初のパケットが確認されたときのシステム アップタイムを非キー フィールドとして設定し、フローの最初のパケットが確認されたときのシステム アップタイムに基づいてタイムスタンプを収集します。
<b>last</b>	フローの最後のパケットが確認されたときのシステム アップタイムを非キー フィールドとして設定し、フローの最後のパケットが確認されたときのシステム アップタイムに基づいてタイムスタンプを収集します。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンド モード

フロー レコード コンフィギュレーション

## サポートされるユーザ ロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。**nonkey** フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されません。**nonkey** フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

このコマンドには、ライセンスは必要ありません。

## 例

次に、フローの最初のパケットが確認されたときのシステム アップタイムに基づくタイムスタンプを非キー フィールドとして設定し、フローの最初のパケットが確認されたときのシステム アップタイムを収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect timestamp sys-uptime first
```

## collect timestamp sys-uptime

次に、フローの最後のパケットが確認されたときのシステム アップタイムに基づくタイムスタンプを非キー フィールドとして設定し、フローの最後のパケットが確認されたときのシステム アップタイムを収集する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect timestamp sys-uptime last
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect interface</b>	入力または出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス (interface フィールド) に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポート フィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# collect transport tcp flags

TCP フィールドを非キー フィールドとして設定し、任意の Flexible NetFlow フロー レコードについてそのフィールド値を収集するには、**collect transport tcp flags** コマンドを使用します。TCP フィールドを Flexible NetFlow フロー レコードの非キー フィールドとして使用する設定をディセーブルにするには、このコマンドの **no** 形式を使用します。

**collect transport tcp flags**

**no collect transport tcp flags**

## 構文の説明

このコマンドには、引数またはキーワードはありません。

## デフォルト

このコマンドはデフォルトではディセーブルになっています。

## コマンド モード

フロー レコード コンフィギュレーション

## サポートされるユーザ ロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**collect** で始まる Flexible NetFlow コマンドは、フロー モニタ レコードの非キー フィールドを設定し、そのレコードによって作成されたフローの各フィールドに値を取り込むために使用します。**nonkey** フィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されず。**nonkey** フィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、非キー フィールドの値はフロー内の最初のパケットからのみ取り出されます。

このコマンドには、ライセンスは必要ありません。

## 例

次に、TCP フラグを非キー フィールドとして設定する例を示します。

```
switch(config)# flow record FLOW-RECORD-1
switch(config-flow-record)# collect transport tcp flags
```

## 関連コマンド

コマンド	説明
<b>collect counter</b>	カウンタを非キー フィールドとして設定し、カウンタ値を収集します。
<b>collect flow</b>	フロー識別フィールドを非キー フィールドとして設定し、その値を収集します。

コマンド	説明
<b>collect interface</b>	入力または出力インターフェイスを非キー フィールドとして設定し、その値を収集します。
<b>collect ipv4</b>	IPv4 フィールドを非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect routing</b>	ルーティング属性を非キー フィールドとして設定し、そのフィールド値を収集します。
<b>collect timestamp</b>	タイムスタンプ フィールドを非キー フィールドとして設定し、その値を収集します。
<b>collect transport</b>	トランスポート層フィールドを非キー フィールドとして設定し、その値を収集します。
<b>flow record</b>	フロー レコードを作成します。
<b>match flow</b>	1 つ以上のフロー フィールドをキー フィールドとして設定します。
<b>match interface</b>	インターフェイス ( <b>interface</b> フィールド) に対するトラフィック フローの方向をキー フィールドとして設定します。
<b>match ipv4</b>	1 つ以上の IPv4 フィールドをキー フィールドとして設定します。
<b>match routing</b>	1 つ以上のルーティング フィールドをキー フィールドとして設定します。
<b>match timestamp</b>	タイムスタンプ フィールドをキー フィールドとして設定します。
<b>match transport</b>	1 つ以上のトランスポート フィールドをキー フィールドとして設定します。
<b>show flow record</b>	フロー レコードのステータスと統計情報を表示します。

# commit (Call home)

Cisco Fabric Services (CFS) 設定を配信するには、**commit** コマンドを使用します。

## commit

### 構文の説明

このコマンドには、引数またはキーワードはありません。

### デフォルト

なし

### コマンドモード

Call home コンフィギュレーション

### サポートされるユーザロール

network-admin  
vdc-admin

### コマンド履歴

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

### 使用上のガイドライン

**commit** コマンドを使用すると、ファブリック内のすべての CFS がイネーブルなデバイスの実行コンフィギュレーションに、CFS 設定が配信されます。

**commit** コマンドは、ファブリック ロックが取得されたデバイスだけで使用できます。

まだコミットされていない設定の変更内容（まだ作業中のコピーとして保存されている）、実行コンフィギュレーションには存在せず、**show** コマンドの出力には表示されません。

すべてのデバイスで確実に同期がとられるようにする場合には、現在の設定を配信する空のコミットを使用できます。

このコマンドには、ライセンスは必要ありません。

### 例

次に、CFS 設定をコミットし、コミットが正常終了したことを確認する例を示します。

```
switch(config-callhome)# commit
switch(config-callhome)# show callhome session status
Last Action Time Stamp      : Tue Dec 23 11:15:02 2008
Last Action                  : Commit
Last Action Result           : Success
Last Action Failure Reason   : none
```

### 関連コマンド

コマンド	説明
<b>abort</b>	CFS セッションを削除します。

コマンド	説明
<b>show cfs application</b>	現在 CFS がイネーブルのアプリケーションを表示します。
<b>show application_name session status</b>	アプリケーションの CFS コンフィギュレーションセッションに関する情報を表示します。

# commit (セッション マネージャ)

セッション マネージャ コンフィギュレーション セッション内のコマンドを有効にし、適用するには、**commit** コマンドを使用します。

**commit** [**verbose**]

## 構文の説明

**verbose** (任意) **commit** コマンドの結果の詳細バージョンを表示します。

## デフォルト

なし

## コマンド モード

セッション コンフィギュレーション

## サポートされるユーザ ロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

**commit** コマンドを使用すると、セッション マネージャ コンフィギュレーション全体が有効にされ、有効な場合には、設定がデバイスに適用されます。

このコマンドには、ライセンスは必要ありません。

## 例

次に、セッション マネージャ コンフィギュレーションをコミットする例を示します。

```
switch# config session ACL_tcp_in
Config Session started, Session ID is 1
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-s)# verify
Verification Successful
switch(config-s)# commit
Commit Successful
switch#
```

## 関連コマンド

コマンド	説明
<b>abort</b>	セッションを削除し、セッション コンフィギュレーション モードを終了します。

コマンド	説明
<b>exit</b>	コマンドをコミットせずにセッション コンフィギュレーション モードを終了します。
<b>show configuration session</b>	セッション マネージャ コンフィギュレーション セッションに関する情報を表示します。

# configure session

セッション マネージャ機能でアクセス コントロール リスト (ACL) コンフィギュレーション セッションを作成または変更するには、**configure session** コマンドを使用します。

## **configure session** *name*

### 構文の説明

<i>name</i>	セッションの名前。63 文字以内の英数字のストリング (大文字と小文字を区別) で指定します。
-------------	---

### デフォルト

なし

### コマンド モード

任意のコマンド モード

### サポートされるユーザ ロール

network-admin  
network-operator  
vdc-admin  
vdc-operator

### コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

### 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

Session Manager がサポートするのは、Access Control List (ACL; アクセス コントロール リスト) 機能だけです。

### 例

次に、ACL コンフィギュレーション セッションを作成する例を示します。

```
switch# configure session myACLs
switch(config-s)#
```

### 関連コマンド

コマンド	説明
<b>show configuration session</b>	セッション マネージャ コンフィギュレーション セッションに関する情報を表示します。

# contract-id

Call home でサービス契約の契約 ID を指定するには、**contract-id** コマンドを使用します。これを削除するには、このコマンドの **no** 形式を使用します。

**contract-id** *contract\_id\_number*

**no contract-id**

## 構文の説明

*contract\_id\_number* サービス契約からの、このデバイスの契約番号。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。

## デフォルト

なし

## コマンド モード

Call home コンフィギュレーション

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、Call home でサービス契約の契約 ID を設定する例を示します。

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

## 関連コマンド

コマンド	説明
<b>callhome</b>	Call home コンフィギュレーション モードを開始します。
<b>email-contact</b>	デバイスの担当者の E メールアドレスを指定します。
<b>phone-contact</b>	デバイスの担当者の電話番号を指定します。
<b>streetaddress</b>	デバイスの担当者の所在地住所を指定します。
<b>customer-id</b>	このデバイスのサービス契約のカスタマー番号を指定します。
<b>site-id</b>	このデバイスのサイト ID 番号を指定します。
<b>switch-priority</b>	このデバイスのプライオリティ番号を指定します。
<b>show callhome</b>	Call home コンフィギュレーションを表示します。

# counter

簡易ネットワーク管理プロトコル (SNMP) のポート モニタ カウンタを設定するには、**counter** コマンドを使用します。ポート モニタ カウンタを削除するには、このコマンドの **no** 形式を使用します。

```
counter {invalid-crc [poll-interval poll-interval {absolute rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id] |
delta rising-threshold rising-threshold event event-id [falling-threshold
falling-threshold event event-id]}] | invalid-words [poll-interval poll-interval
{absolute rising-threshold rising-threshold event event-id [falling-threshold
falling-threshold event event-id] | delta rising-threshold rising-threshold event
event-id [falling-threshold falling-threshold event event-id]}] | link-loss [poll-interval
poll-interval {absolute rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id] | delta rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id]}] |
protocol-error [poll-interval poll-interval {absolute rising-threshold rising-threshold
event event-id [falling-threshold falling-threshold event event-id] | delta
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id]}] | rx-performance [poll-interval poll-interval {absolute
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id] | delta rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id]}] | signal-loss [poll-interval
poll-interval {absolute rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id] | delta rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id]}] |
sync-loss [poll-interval poll-interval {absolute rising-threshold rising-threshold
event event-id [falling-threshold falling-threshold event event-id] | delta
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id]}] | tx-performance [poll-interval poll-interval {absolute
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id] | delta rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id]}]}]
```

```
no counter {invalid-crc [poll-interval poll-interval {absolute rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id] |
delta rising-threshold rising-threshold event event-id [falling-threshold
falling-threshold event event-id]}] | invalid-words [poll-interval poll-interval
{absolute rising-threshold rising-threshold event event-id [falling-threshold
falling-threshold event event-id] | delta rising-threshold rising-threshold event
event-id [falling-threshold falling-threshold event event-id]}] | link-loss [poll-interval
poll-interval {absolute rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id] | delta rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id]}] |
protocol-error [poll-interval poll-interval {absolute rising-threshold rising-threshold
event event-id [falling-threshold falling-threshold event event-id] | delta
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id]}] | rx-performance [poll-interval poll-interval {absolute
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id] | delta rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id]}] | signal-loss [poll-interval
poll-interval {absolute rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id] | delta rising-threshold
rising-threshold event event-id [falling-threshold falling-threshold event event-id]}] |
```

```

sync-loss [poll-interval poll-interval {absolute rising-threshold rising-threshold
event event-id [falling-threshold falling-threshold event event-id] | delta
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id]}] | tx-performance [poll-interval poll-interval {absolute
rising-threshold rising-threshold event event-id [falling-threshold falling-threshold
event event-id] | delta rising-threshold rising-threshold event event-id
[falling-threshold falling-threshold event event-id]}]}

```

**構文の説明**

<b>invalid-crc</b>	無効な CRC カウンタを設定します。
<i>poll-interval</i>	(任意) カウンタのポーリング間隔 (秒単位)。有効な範囲は 0 ~ 2147483647 です。
<b>absolute</b>	絶対タイプしきい値を指定します。
<b>rising-threshold</b>	上昇しきい値を設定します。
<i>rising-threshold</i>	上昇しきい値の限度。範囲は 0 ~ 18446744073709551615 です。
<b>event</b>	上昇しきい値イベントを設定します。
<i>event-id</i>	イベント設定からのイベント ID。範囲は 1 ~ 65535 です。
<b>falling-threshold</b>	(任意) 下限しきい値を指定します。
<i>falling-threshold</i>	(任意) 下限しきい値の限度。有効な範囲は -2147483648 ~ 2147483647 です。
<b>delta</b>	(任意) デルタ タイプしきい値を指定します。
<b>invalid-words</b>	無効なワード カウンタを設定します。
<b>link-loss</b>	リンク損失カウンタを設定します。
<b>protocol-error</b>	プロトコル エラー カウンタを設定します。
<b>rx-performance</b>	入力 (rx) パフォーマンス カウンタを設定します。
<b>signal-loss</b>	信号損失カウンタを設定します。
<b>sync-loss</b>	同期損失カウンタを設定します。
<b>tx-performance</b>	出力 (tx) パフォーマンス カウンタを設定します。

**デフォルト**

なし

**コマンド モード**

ポート モニタ コンフィギュレーション (config-port-monitor)

**サポートされるユーザ ロール**network-admin  
vdc-admin**コマンド履歴**

リリース	変更箇所
4.1(2)	このコマンドが追加されました。

**使用上のガイドライン**

このコマンドには、ライセンスは必要ありません。

**例**

次に、SNMP カウンタを設定する例を示します。

```
switch(config) port-monitor name PM1  
switch(config-port-monitor) # counter invalid-crc poll-interval 30 absolute  
rising-threshold 10000000 event 100  
switch(config-port-monitor) #
```

次に、SNMP カウンタ設定を削除する例を示します。

```
switch(config) # no counter invalid-crc poll-interval 30 absolute rising-threshold 10000000  
event 100  
switch(config-port-monitor) #
```

**関連コマンド**

コマンド	説明
<b>monitor counter</b>	モニタ カウンタを設定します。

# customer-id

Call home でサービス契約の顧客 ID を指定するには、**customer-id** コマンドを使用します。これを削除するには、このコマンドの **no** 形式を使用します。

**customer-id** *contract\_id\_number*

**no customer-id**

## 構文の説明

*contract\_id\_number* サービス契約からの、このデバイスの顧客番号。顧客番号は、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。

## デフォルト

なし

## コマンドモード

Call home コンフィギュレーション

## サポートされるユーザロール

network-admin  
vdc-admin

## コマンド履歴

リリース	変更箇所
4.0(1)	このコマンドが追加されました。

## 使用上のガイドライン

このコマンドには、ライセンスは必要ありません。

## 例

次に、Call home でサービス契約の顧客 ID を設定する例を示します。

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

## 関連コマンド

コマンド	説明
<b>callhome</b>	Call home コンフィギュレーション モードを開始します。
<b>email-contact</b>	デバイスの担当者の E メールアドレスを指定します。
<b>phone-contact</b>	デバイスの担当者の電話番号を指定します。
<b>streetaddress</b>	デバイスの担当者の所在地住所を指定します。
<b>contract-id</b>	このデバイスのサービス契約の連絡先番号を指定します。
<b>site-id</b>	このデバイスのサイト ID 番号を指定します。
<b>switch-priority</b>	このデバイスのプライオリティ番号を指定します。
<b>show callhome</b>	Call home コンフィギュレーションを表示します。