コマンドライン インターフェイスの概要

この章の内容は、次のとおりです。

- 機能情報の確認，2 ページ
- CLI プロンプトの概要，2 ページ
- コマンド モード，3 ページ
- 特殊文字，7 ページ
- キーストローク ショートカット，8 ページ
- コマンドの省略形，11 ページ
- 部分的なコマンド名の補完，11 ページ
- コマンド階層での場所の特定，12 ページ
- コマンドの no 形式の使用方法 ，12 ページ
- CLI 変数の設定，13 ページ
- コマンド エイリアス，16 ページ
- コマンド スクリプト，18 ページ
- 状況依存ヘルプ ，20 ページ
- 正規表現について，22 ページ
- show コマンド出力の検索とフィルタリング，23 ページ
- --More-- プロンプトからの検索およびフィルタリング，29 ページ
- コマンド履歴の使用方法，30 ページ
- CLI 確認プロンプトのイネーブルまたはディセーブル，32 ページ
- CLI 画面の色の設定，32 ページ
- モジュールへのコマンドの送信，33 ページ
- BIOS ローダー プロンプト，34 ページ

- CLI の使用例，34 ページ
- CLI に関する追加情報，36 ページ

機能情報の確認

ご使用のソフトウェア リリースで、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の警告および機能情報については、https://tools.cisco.com/bugsearch/ の Bug Search Tool およびご使用のソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、「新機能および変更された機能に関する情報」の章を参照してください。

CLI プロンプトの概要

デバイスに正常にアクセスすると、コンソール ポートのターミナル ウィンドウまたはリモート ワークステーションに、次の例のような CLI プロンプトが表示されます。

```
User Access Verification
login: admin
Password:<password>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2009, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch#
```

デフォルトのデバイス ホスト名を変更できます。

CLI プロンプトから、次の方法を実行できます。

- 機能を設定するための CLI コマンドを使用する
- コマンド履歴にアクセスする
- コマンド解析機能を使用する

（注） 通常の動作では、ユーザ名の大文字と小文字が区別されます。ただし、コンソール ポートを介してデバイスに接続する場合、ユーザ名がどのように定義されているかに関係なく、すべて大文字でログイン ユーザ名を入力できます。正しいパスワードを入力すれば、デバイスにログインできます。

コマンド モード

ここでは、Cisco NX-OS CLI でのコマンド モードについて説明します。

EXEC コマンド モード

初めてログインしたときに、Cisco NX-OS ソフトウェアでは EXEC モードが開始されます。EXEC モードで使用可能なコマンドには、デバイスの状態および構成情報を表示する **show** コマンド、**clear** コマンド、ユーザがデバイス コンフィギュレーションに保存しない処理を実行するその他のコマンドがあります。

グローバル コンフィギュレーション コマンド モード

グローバルコンフィギュレーションモードでは、広範なコマンドにアクセスできます。この用語は、デバイスに全体的な影響を与える特性または特徴を示しています。グローバル コンフィギュレーション モードでコマンドを入力すると、デバイスをグローバルに設定することができます。また、さらに特定のコンフィギュレーション モードを開始して、インターフェイスやプロトコルなどの特定の要素を設定することもできます。

手順の概要

1. **configure terminal**

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。
		（注） CLI プロンプトが変わり、グローバル コンフィギュレーション モードに入ったことが示されます。

インターフェイス コンフィギュレーション コマンド モード

インターフェイス コンフィギュレーション モードは、グローバル コンフィギュレーション モードから開始する、特定のコンフィギュレーション モードの 1 例です。デバイスのインターフェイスを設定するには、インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始する必要があります。

インターフェイスごとに多くの機能をイネーブルにする必要があります。インターフェイス コンフィギュレーションコマンドを使用すると、イーサネットインターフェイスや管理インターフェイス（mgmt 0）などの、デバイス上のインターフェイスの動作が変更されます。

インターフェイスの設定の詳細については、ご使用のデバイスの『Cisco Nexus Interfaces Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **interface** *type number*

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	**interface** *type number* 例： switch(config)# interface ethernet 2/2 switch(config-if)#	設定するインターフェイスを指定します。 CLI により、指定したインターフェイスのインターフェイス コンフィギュレーション モードになります。 （注） CLI プロンプトが変わり、インターフェイス コンフィギュレーション モードに入ったことが示されます。

サブインターフェイス コンフィギュレーション コマンド モード

グローバル コンフィギュレーション モードから、サブインターフェイスと呼ばれる VLAN インターフェイスを設定するためのコンフィギュレーション サブモードにアクセスできます。サブインターフェイス コンフィギュレーション モードでは、1 つの物理インターフェイスに複数の仮想インターフェイスを設定できます。サブインターフェイスは、別個の物理インターフェイスとしてプロトコルに認識されます。

また、サブインターフェイスは、プロトコルによる単一インターフェイスでの複数のカプセル化を可能にします。たとえば、IEEE 802.1Q カプセル化を設定して、サブインターフェイスを VLAN に関連付けることができます。

サブインターフェイスの設定の詳細については、ご使用のデバイスの『Cisco Nexus Interfaces Guide』を参照してください。サブインターフェイス コマンドの詳細については、ご使用のデバイスの『Command Reference Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **interface***typenumber***.***subint*

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	**interface***typenumber***.***subint*
	
	例：
	`switch(config)# interface ethernet 2/2.1`
	`switch(config-subif)#`
		設定する VLAN インターフェイスを指定します。
		CLI は、指定した VLAN インターフェイスに対するサブインターフェイス コンフィギュレーション モードを開始します。
		（注） CLI プロンプトが変わり、グローバル コンフィギュレーション モードに入ったことが示されます。

コマンド モードの保存と復元

Cisco NX-OS ソフトウェアでは、現在のコマンド モードを保存しておき、機能を設定した後に、前のコマンド モードを復元することができます。**push** コマンドでコマンド モードを保存し、**pop** コマンドでコマンド モードを復元します。

次の例は、コマンド モードを保存し、復元する方法を示します。

```
switch# configure terminal
switch(config)# event manager applet test
switch(config-applet)# push
switch(config-applet)# configure terminal
switch(config)# username testuser password newtest
switch(config)# pop
switch(config-applet)#
```

コンフィギュレーション コマンド モードの終了

コンフィギュレーション コマンド モードを終了するには、次のいずれかの作業を行います。

手順の概要

1. **exit**
2. **end**
3. （任意） Ctrl+Z

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**exit**
	
	例：
	`switch(config-if)# exit`
	`switch(config)#`
		現在のコンフィギュレーション コマンド モードを終了して、以前のコンフィギュレーション コマンド モードに戻ります。

ステップ 2	**end**
	
	例：
	`switch(config-if)# end`
	`switch#`
		現在のコンフィギュレーション コマンド モードを終了して、EXEC モードに戻ります。

ステップ 3	Ctrl+Z
	
	例：
	`switch(config-if)# ^z`
	`switch#`
		（任意）
		現在のコンフィギュレーション コマンド モードを終了して、EXEC モードに戻ります。
		注意 有効なコマンドを入力してから、コマンド ラインの最後で Ctrl+Z を押すと、CLI によってそのコマンドが実行コンフィギュレーションファイルに追加されます。ほとんどの場合、**exit** または **end** コマンドを使用してコンフィギュレーション モードを終了する必要があります。

コマンド モードの概要

この表は、主なコマンド モードの概要を示しています。

表 *1*：コマンド モードの概要

モード	アクセス方式	プロンプト	終了方法
EXEC	ログインプロンプトから、ユーザ名とパスワードを入力します。	`switch#`	終了してログインプロンプトに戻るには、**exit** コマンドを使用します。
グローバル コンフィギュレーション	EXEC モードで、**configure terminal** コマンドを使用します。	`switch(config)#`	終了して EXEC モードに戻るには、**end** または **exit** コマンドを使用するか、Ctrl+Z を押します。
インターフェイス コンフィギュレーション	グローバル コンフィギュレーションモードで、**interface** コマンドを使用してインターフェイスを指定します。	`switch(config-if)#`	終了してグローバル コンフィギュレーションモードに戻るには、**exit** コマンドを使用します。 終了して EXEC モードに戻るには、**end** コマンドを使用するか、Ctrl+Z を押します。
サブインターフェイス コンフィギュレーション	グローバル コンフィギュレーションモードで、**interface** コマンドを使用してサブインターフェイスを指定します。	`switch(config-subif)#`	終了してグローバル コンフィギュレーションモードに戻るには、**exit** コマンドを使用します。 終了して EXEC モードに戻るには、**end** コマンドを使用するか、Ctrl+Z を押します。

特殊文字

次の表に、Cisco NX-OS のテキスト ストリングで特別な意味を持つ文字を示します。正規表現あるいはその他の特有なコンテキストでのみ使用します。

表 *2*：特殊文字

文字	説明
%	パーセント
#	ポンド、ハッシュ、または番号
...	省略符号
|	縦線
< >	より小さい、またはより大きい
[ ]	角カッコ
{ }	波カッコ

キーストローク ショートカット

次の表に、EXEC モードおよびコンフィギュレーション モードの両方で使用されるコマンド キーの組み合わせを示します。

表 *3*：キーストローク ショートカット

キーストローク	説明
Ctrl+A	カーソルを行の先頭に移動します。
Ctrl+B	カーソルを 1 文字左に移動します。複数行にわたってコマンドを入力するときは、左矢印キーまたは Ctrl+B キーを繰り返し押してシステム プロンプトまでスクロールバックして、コマンドエントリの先頭まで移動できます。あるいは Ctrl+A キーを押してコマンドエントリの先頭に移動します。
Ctrl+C	コマンドを取り消して、コマンドプロンプトに戻ります。
Ctrl+D	カーソル位置にある文字を削除します。
Ctrl+E	カーソルを行の末尾に移動します。
Ctrl+F	カーソルを 1 文字右に移動します。

キーストローク	説明
Ctrl+G	コマンド ストリングを削除せずに、コマンドモードを終了して以前のコマンドモードに戻ります。
Ctrl+K	カーソル位置からコマンドラインの末尾までのすべての文字を削除します。
Ctrl+L	現在のコマンド ラインを再表示します。
Ctrl+N	コマンド履歴の次のコマンドを表示します。
Ctrl+O	端末画面をクリアします。
Ctrl+P	コマンド履歴の前のコマンドを表示します。
Ctrl-R	現在のコマンド ラインを再表示します。
Ctrl+T	カーソルの下の文字を、カーソルの右にある文字と置き換えます。カーソルが 1 文字右に移動します。
Ctrl+U	カーソル位置からコマンドラインの先頭までのすべての文字を削除します。
Ctrl+V	次のキーストロークに関する特別な意味を削除します。たとえば、正規表現で疑問符（?）を入力する前に、Ctrl+V を押します。
Ctrl+W	カーソルの左にある単語を削除します。
Ctrl+X、H	入力したコマンドの履歴を表示します。 このキーの組み合わせを使用するときは、Ctrl キーと X キーを同時に押してリリースしてから、H を押します。
Ctrl+Y	バッファ内の最新のエントリを呼び出します（キーを同時に押します）。
Ctrl+Z	コンフィギュレーション セッションを終了して、EXEC モードに戻ります。 有効なコマンドを入力してから、コマンドラインの最後で Ctrl+Z を使用すると、コマンドの結果の設定がまず実行コンフィギュレーション ファイルに追加されます。

キーストローク	説明
上矢印キー	コマンド履歴の前のコマンドを表示します。
下矢印キー	コマンド履歴の次のコマンドを表示します。
右矢印キー 左矢印キー	コマンドストリング上でカーソルを前後に移動して、現在のコマンドを編集します。
?	使用可能なコマンドのリストを表示します。

Tab	ワードの最初の文字を入力して Tab キーを押すと、ワードが補完されます。文字に一致するすべてのオプションが表示されます。
	タブを使用して、次の項目を入力します。
	• コマンド名
	• ファイル システム内のスキーム名
	• ファイル システム内のサーバ名
	• ファイル システム内のファイル名
	例：
	`switch(config)# xm<Tab>`
	`switch(config)# xml<Tab>`
	`switch(config)# xml server`

	例：
	`switch(config)# c<Tab>`
	`callhome  class-map  clock  cts`
	`cdp  cli  control-plane`
	`switch(config)# cl<Tab>`
	`class-map  cli  clock`
	`switch(config)# cla<Tab>`
	`switch(config)# class-map`

	例：
	`switch# cd bootflash:<Tab>`
	`bootflash:  bootflash://sup-1/`
	`bootflash:///  bootflash://sup-2/`
	`bootflash://module-5/`
	`bootflash://sup-active/`
	`bootflash://module-6/  bootflash://sup-local/`

	例：
	`switch# cd bootflash://mo<Tab>`
	`bootflash://module-5/`
	`bootflash://module-6/cv`
	`switch# cd bootflash://module-`

コマンドの省略形

コマンドの最初の数文字を入力することで、コマンドおよびキーワードを省略できます。省略形には、コマンドまたはキーワードを一意に識別でき得る文字数を含める必要があります。コマンドの入力で問題が生じた場合は、システムプロンプトを確認し、疑問符（?）を入力して使用できるコマンドのリストを表示してください。コマンド モードが間違っているか、間違った構文を使用している可能性があります。

次の表に、コマンド省略形の例を示します。

表 *4*：コマンド省略形の例

コマンド	省略形
configure terminal	**conf t**
copy running-config startup-config	**copy run start**
interface ethernet 1/2	**int e 1/2**
show running-config	**sh run**

部分的なコマンド名の補完

完全なコマンド名を思い出せない場合や、入力の作業量を減らす場合は、コマンドの先頭の数文字を入力して、Tab キーを押します。コマンドライン パーサーは、入力されたストリングがコマンド モードで一意である場合に、コマンドを補完します。キーボードに Tab キーがない場合は、代わりに Ctrl+I キーを押します。

コマンドは、コマンドが一意になるのに十分な文字が入力されていれば、CLI によって認識されます。たとえば、EXEC モードで **conf** と入力すると、CLI はエントリを **configure** コマンドと関連付けることができます。これは、**conf** で始まるコマンドが **configure** コマンドしかないためです。

次の例では、Tab キーを押したときに、CLI によって EXEC モードで **conf** の一意の文字列が認識されます。

```
switch# conf<Tab>
switch# configure
```

コマンド補完機能を使用すると、CLI により完全なコマンド名が表示されます。コマンドは、Return キーまたは Enter キーを押すまで、CLI によって実行されません。この機能により、完全なコマンドが省略形によって意図したものでない場合に、コマンドを修正できます。入力した一連の文字に対して、対応するコマンドが複数ある場合は、一致するコマンドのリストが表示されます。

たとえば、**co<Tab>** と入力すると、EXEC モードで利用可能な、**co** で始まるすべてのコマンドがリストされます。

```
switch# co<Tab>
configure   copy
switch# co
```

コマンドエントリを補完できるよう、入力した文字は再びプロンプトに表示されることに注意してください。

コマンド階層での場所の特定

一部の機能では、コンフィギュレーション サブモード階層が 1 つのレベル以上ネストされます。この場合は、Present Working Context（PWC）に関する情報を表示できます。

手順の概要

1. **where detail**

手順の詳細

	コマンドまたはアクション	目的
ステップ **1**	**where detail** 例： switch# configure terminal switch(config)# interface mgmt0 switch(config-if)# where detail mode: conf interface mgmt0 username: admin	PWC を表示します。

コマンドの no 形式の使用方法

大部分のコンフィギュレーションコマンドには **no** 形式があり、これを使用して、機能をディセーブルにしたり、デフォルト値に戻したり、設定を削除したりできます。Cisco NX-OS のコマンドリファレンスの資料では、コマンドの **no** 形式が使用できる場合は常に no 形式の機能について説明しています。

次に、機能をディセーブルにする例を示します。

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# no feature tacacs+
```

次に、機能をデフォルト値に戻す例を示します。

```
switch# configure terminal
switch(config)# banner motd #Welcome to the switch#
switch(config)# show banner motd
Welcome to the switch

switch(config)# no banner motd
switch(config)# show banner motd
User Access Verification
```

次に、機能の設定を削除する例を示します。

```
switch# configure terminal
switch(config)# radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
        10.10.2.2:
                available for authentication on port:1812
                available for accounting on port:1813

switch(config)# no radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
```

次に、EXEC モードでコマンドの **no** 形式を使用する例を示します。

```
switch# cli var name testinterface ethernet1/2
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
testinterface="ethernet1/2"

switch# cli no var name testinterface
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
```

CLI 変数の設定

ここでは、Cisco NX-OS CLI の CLI 変数について説明します。

CLI 変数について

Cisco NX-OS ソフトウェアは CLI コマンドでの変数の定義および使用をサポートします。

CLI 変数は、次の方法で参照できます。

- コマンドラインで直接入力する。
- **run-script** コマンドを使用して開始されたスクリプトに渡す。親シェルで定義した変数は子 **run-script** コマンド プロセスで使用できます。

CLI 変数には、次の特性があります。

- 入れ子状態の参照を使用して、別の変数から変数を参照することはできません。
- スイッチのリロード時に維持することも、現在のセッションのみに使用することもできます。

Cisco NX-OS は、事前定義された TIMESTAMP 変数をサポートします。この変数は、コマンドを実行するときの YYYY-MM-DD-HH.MM.SS フォーマットの現在時刻を参照します。

（注） TIIMESTAMP 変数名は大文字と小文字を区別します。文字はすべて大文字です。

CLI セッション限定の変数の設定

CLI セッションの間だけ有効な CLI セッション変数を定義できます。これらの変数は定期的に実行するスクリプトに役立ちます。丸括弧で名前を囲み、変数の前にドル記号（$）を付けることによって、変数を参照できます。たとえば、$(*variable-name*) です。

手順の概要

1. **cli var name** *variable-name variable-text*
2. （任意） **show cli variables**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**cli var name** *variable-name variable-text* 例： `switch# cli var name testinterface ethernet 2/1`	CLI セッション変数を設定します。*variable-name* 引数は、31 文字以内の英数字で指定します。大文字と小文字が区別されます。*variable-text* 引数は、200 文字以内の英数字で指定します。大文字と小文字が区別されます。スペースを含めることができます。
ステップ 2	**show cli variables** 例： `switch# show cli variables`	（任意） CLI 変数の設定を表示します。

固定 CLI 変数の設定

CLI セッションの終了後やデバイスのリロード後に保持される CLI 変数を設定できます。

手順の概要

1. **configure terminal**
2. **cli var name***variable-name variable-text*
3. **exit**
4. （任意） **show cli variables**
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	**cli var name***variable-name variable-text* 例： `switch(config)# cli var name testinterface ethernet 2/1`	CLI 固定変数を設定します。変数名は、英数字の文字列で指定します。大文字と小文字が区別されます。変数名の先頭は英字にする必要があります。31 文字以内で指定します。

ステップ 3	**exit**
	
	例：
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 4	**show cli variables** 例： `switch# show cli variables`	（任意） CLI 変数の設定を表示します。
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。

コマンド エイリアス

ここでは、コマンド エイリアスについて説明します。

コマンド エイリアスについて

コマンド エイリアスを定義して、使用頻度の高いコマンドを置き換えることができます。コマンド エイリアスは、コマンド構文の全体または一部を表すことができます。

コマンド エイリアスには、次の特性があります。

- コマンド エイリアスはすべてのユーザ セッションに対してグローバルです。
- コマンド エイリアスは、スタートアップ コンフィギュレーションに保存しておけば、再起動後も維持されます。
- コマンド エイリアス変換は常にすべてのコンフィギュレーション モードまたはサブモードのすべてのキーワードの中で最優先されます。
- コマンド エイリアスの設定は他のユーザ セッションに対してただちに有効になります。
- Cisco NX-OS ソフトウェアには、デフォルトのエイリアス **alias** が用意されています。このエイリアスは、**show cli alias** コマンドと同等であり、ユーザ定義のエイリアスをすべて表示します。
- デフォルトのコマンド エイリアス **alias** は、削除することも変更することもできません。
- エイリアスは最大深度 1 までネストできます。1 つのコマンド エイリアスは、有効なコマンドを参照する必要がある別のコマンド エイリアスを参照できますが、その他のコマンド エイリアスは参照できません。
- コマンドエイリアスは必ず、コマンドラインの最初のコマンドキーワードを置き換えます。
- 任意のコマンド モードでコマンドのコマンド エイリアスを定義できます。
- コマンドエイリアス内でCLI変数を参照すると、変数参照ではなくその変数の現在の値がエイリアス内で使用されます。
- コマンド エイリアスは **show** コマンドの検索およびフィルタリングに使用できます。

コマンド エイリアスの定義

よく使用するコマンドにはコマンド エイリアスを定義できます。

手順の概要

1. **configure terminal**
2. **cli alias name***alias-name alias-text*
3. **exit**
4. （任意） **alias**
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	**cli alias name***alias-name alias-text* 例： `switch(config)# cli alias name ethint interface ethernet`	コマンド エイリアスを設定します。エイリアス名は英数字で表します。大文字と小文字は区別されません。先頭は英字にする必要があります。30 文字以内で指定します。

ステップ 3	**exit**
	
	例：
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 4	**alias** 例： `switch# alias`	（任意） コマンド エイリアス設定を表示します。
ステップ 5	**copy running-config startup-config** 例： `switch# 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。

ユーザ セッション用のコマンド エイリアスの設定

Cisco NX-OS デバイス上の他のすべてのユーザが使用できない、現在のユーザ セッション用のコマンド エイリアスを作成できます。また、コマンド エイリアスを保存し、現在のユーザ アカウントであとで使用することもできます。

手順の概要

1. **terminal alias** [**persist**] *alias-name command -string*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**terminal alias** [**persist**] *alias-name command -string* 例： `switch# terminal alias shintbr show interface brief`	現在のユーザセッション用のコマンドエイリアスを設定します。このユーザアカウントにより将来使用するためにエイリアスを保存するには、**persist** キーワードを使用します。 （注） **persist** キーワードを省略しないでください。

コマンド スクリプト

ここでは、複数のタスクを実行するためにコマンドのスクリプトを作成する方法について説明します。

コマンド スクリプトの実行

ファイルでコマンドのリストを作成し、CLI からこれらのコマンドを実行できます。コマンド スクリプトでは CLI 変数を使用できます。

（注） CLI プロンプトではスクリプト ファイルを作成できません。スクリプト ファイルは、リモート デバイスで作成し、Cisco NX-OS デバイス上の bootflash:、slot0:、または volatile: ディレクトリにコピーします。

手順の概要

1. **run-script** [**bootflash:** | **slot0:** | **volatile:**]*filename*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**run-script** [**bootflash:** | **slot0:** | **volatile:**]*filename* 例： `switch# run-script testfile`	デフォルト ディレクトリのファイル内のコマンドを実行します。

端末への情報のエコー

端末に情報をエコーできます。これは、コマンド スクリプトで特に役立ちます。CLI 変数を参照し、エコーされるテキストでフォーマット オプションを使用できます。

次の表に、テキストに挿入できるフォーマット オプションを示します。

表 5：***echo*** **コマンドのフォーマット オプション**

フォーマット オプション	説明
\b	バック スペースを挿入します。
\c	テキスト ストリングの最後にある改行文字が削除されます。
\f	フォーム フィード文字が挿入されます。
\n	改行文字が挿入されます。
\r	テキスト行の最初に戻ります。
\t	水平タブ文字が挿入されます。
\v	垂直タブ文字が挿入されます。
\\	バックスラッシュ文字が表示されます。
nnn	対応する ASCII 8 進文字が表示されます。

手順の概要

1. **echo** [**backslash-interpret**] [*text*]

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**echo** [**backslash-interpret**] [*text*]
	
	例：
	`switch# echo This is a test.`
	`This is a test.`
		backslash-interpret キーワードは、テキスト文字列にフォーマット オプションが含まれていることを示します。*text* 引数は、英数字で指定します。大文字と小文字が区別されます。空白を含めることができます。200 文字以内で指定します。デフォルトは空白行です。

コマンド処理の遅延

コマンドアクションを一定の時間、遅延できます。これは、コマンドスクリプト内で特に役に立ちます。

手順の概要

1. **sleep***seconds*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**sleep***seconds* 例： `switch# sleep 30`	遅延をある秒数発生させます。指定できる範囲は 0 ～ 2147483647 です。

状況依存ヘルプ

Cisco NX-OS ソフトウェアの CLI には、状況依存ヘルプが用意されています。コマンド内の任意の位置に疑問符（?）を使用することで、有効な入力オプションを表示できます。

CLI では、入力エラーを特定するためにキャレット（^）記号が使用されます。^ 記号は、コマンド ストリング内の誤ったコマンド、キーワード、または引数が入力された位置に表示されます。

次の表に、状況依存ヘルプの出力例を示します。

表 *6*：状況依存ヘルプの例

出力例	説明
switch# **clock ?** set HH:MM:SS Current Time switch# clock	EXEC モードの **clock** コマンドのコマンド構文を表示します。 このスイッチの出力では、**clock** コマンドを使用するためには **set** キーワードが必要であることが示されています。
switch# clock **set ?** WORD HH:MM:SS Current Time switch# clock set	時刻を設定するためのコマンド構文を表示します。 ヘルプ出力は、クロックの設定に現在時刻が必要であること、および時刻のフォーマット方法を示しています。
switch# clock set **13:32:00<CR>** % Incomplete command switch#	現在時刻を追加します。 CLI は、コマンドが不完全であることを示しています。
switch# **<Ctrl-P>** switch# clock set 13:32:00	入力した直前のコマンドを表示します。
switch# clock set 13:32:00 **?** <1-31> Day of the month switch# clock set 13:32:00	**clock set** コマンドに対する追加の引数を表示します。

switch# clock set 13:32:00 18 **?**
  April Month of the year
  August Month of the year
  December Month of the year
  February Month of the year
  January Month of the year
  July Month of the year
  June Month of the year
  March Month of the year
  May Month of the year
  November Month of the year
  October Month of the year
  September Month of the year
switch# clock set 13:32:00 18
	clock set コマンドに対する追加の引数を表示します。

switch# clock set 13:32:00 18 **April 08<CR>** % Invalid input detected at '^' marker.	クロック設定に日付を追加します。 CLI は、08 の位置にキャレット記号（^）を使用してエラーを示しています。
switch# clock set 13:32:00 18 April **?** <2000-2030> Enter the year (no abbreviation) switch# clock set 13:32:00 18 April	この年に対応する適切な引数を表示します。
switch# clock set 13:32:00 18 April **2008<CR>** switch#	**clock set** コマンドの正しい構文を入力します。

正規表現について

Cisco NX-OS ソフトウェアは、CLI 出力（**show** コマンドなど）の検索またはフィルタリングで正規表現をサポートしています。正規表現では大文字と小文字が区別され、また複雑な一致要件を設定することができます。

特殊文字

他のキーボード文字（!や~など）を、単一文字パターンとして使用することもできますが、特定のキーボード文字は、正規表現内で使用した場合特別な意味を持ちます。

次の表に、特別な意味を持つキーボード文字を示します。

表 7: 特別な意味を持つ特殊文字

文字	特別な意味
.	スペースを含む任意の単一文字と一致します。
*	0個以上のパターンのシーケンスと一致します。
+	1個以上のパターンのシーケンスと一致します。
?	0 または 1 回のパターンと一致します。
^	ストリングの先頭と一致します。
$	ストリングの末尾と一致します。
_（アンダースコア）	カンマ（,）、左波カッコ（{）、右波カッコ（}）、左カッコ（(）、右カッコ（)）、ストリングの先頭、ストリングの末尾、またはスペースと一致します。 （注） アンダースコアは、BPG 関連のコマンドの場合にのみ正規表現として扱われます。

これらの特殊文字を単一文字パターンとして使用するときは、各文字の前にバックスラッシュ（\）を置いて特別な意味を除外してください。次の例には、ドル記号（$）、アンダースコア（_）、およびプラス記号（+）にそれぞれ一致する単一文字パターンが含まれています。

\$ _ \+

複数文字のパターン

文字、数字、または特別な意味を持たないキーボード文字を連結して、複数文字のパターンを指定することもできます。たとえば、a4% は複数文字の正規表現です。

複数文字パターンでは、順序が大切です。a4% という正規表現は、文字が a、4、パーセント記号（%）の順に並んでいる場合に一致します。ストリングの中に a4% という文字がその順序で含まれていないと、パターン マッチングは失敗します。複数文字正規表現 a.（後にピリオドが続く文字）では、特別な意味のピリオド文字が使用され、任意の 1 文字が後に続く文字に一致します。この例では、ab、a!、または a2 というストリングはすべてこの正規表現と一致します。

特殊文字の特別な意味は、特殊文字の前にバックスラッシュを挿入することで無効にできます。たとえば、表現 a\. がコマンド構文で使用されている場合、ストリング a. だけが一致します。

位置指定

特殊文字を使用してストリング内での正規表現の位置を指定することで、正規表現パターンをストリングの先頭または末尾と一致させることができます。

次の表に、位置指定に使用可能な特殊文字を示します。

表 *8*：位置指定に用いられる特殊文字

文字	説明
^	ストリングの先頭と一致します。
$	ストリングの末尾と一致します。

たとえば、正規表現 **^con** は **con** で始まる文字列に一致し、**sole$** は **sole** で終わる文字列に一致します。

（注） ^ 記号は、角カッコで囲まれた範囲に論理関数「not」を指定する場合にも使用されます。たとえば、正規表現 [^abcd] が示す範囲は、a、b、c、d のいずれでもない任意の単一文字と一致します。

show コマンド出力の検索とフィルタリング

多くの場合、**show** コマンドの出力は、長くて煩雑になります。Cisco NX-OS ソフトウェアでは、情報を簡単に見つけ出すために、出力の検索およびフィルタリングを行うことができます。検索

およびフィルタリングのオプションは、**show** コマンドの末尾にパイプ記号（|）を付け、その後に指定します。CLI の状況依存ヘルプ機能を使用してオプションを表示できます。

```
switch# show running-config | ?
  cut      Print selected parts of lines.
  diff     Show difference between current and previous invocation (creates temp files:
           remove them with 'diff-clean' command and don't use it on commands with big
           outputs, like 'show tech'!)
  egrep    Egrep - print lines matching a pattern
  grep     Grep - print lines matching a pattern
  head     Display first lines
  human    Output in human format
  last     Display last lines
  less     Filter for paging
  no-more  Turn-off pagination for command output
  perl     Use perl script to filter output
  section  Show lines that include the pattern as well as the subsequent lines that are
           more indented than matching line
  sed      Stream Editor
  sort     Stream Sorter
  sscp     Stream SCP (secure copy)
  tr       Translate, squeeze, and/or delete characters
  uniq     Discard all but one of successive identical lines
  vsh      The shell that understands cli command
  wc       Count words, lines, characters
  xml      Output in xml format (according to .xsd definitions)
  begin    Begin with the line that matches
  count    Count number of lines
  end      End with the line that matches
  exclude  Exclude lines that match
  include  Include lines that match
```

フィルタリングおよび検索のキーワード

Cisco NX-OS CLI には、**show** コマンドと併用してコマンド出力の検索やフィルタリングを実行できる、一連のキーワードが用意されています。

次の表に、CLI 出力のフィルタリングや検索を行うためのキーワードを示します。

表 *9*：フィルタリングおよび検索のキーワード

キーワードの構文	説明
begin*string* 例： **show version | begin Hardware**	検索ストリングと一致するテキストが含まれている行から表示を開始します。検索ストリングは大文字と小文字が区別されます。
count 例： **show running-config | count**	コマンド出力の行数を表示します。

キーワードの構文	説明
cut [**-d***character*] {**-b** | **-c** | **-f** | **-s**} 例： `show file testoutput | cut -b 1-10`	出力行の一部分だけを表示します。一定のバイト数（**-b**）、文字数（**-vcut** [**-d***character*] {**-b** | **-c** | **-f** | **-s**}）、またはフィールド数（**-f**）を表示できます。また、**-d** キーワードを使用して、デフォルトのタグ文字以外のフィールドデリミタを定義することもできます。**-s** キーワードはデリミタが含まれない行の表示を抑制します。
end*string* 例： `show running-config | end interface`	検索ストリングの最後の一致になるまですべての行を表示します。
exclude*string* 例： `show interface brief | exclude down`	検索ストリングが含まれていない行をすべて表示します。検索ストリングは大文字と小文字が区別されます。
head [**lines***lines*] 例： `show logging logfile | head lines 50`	出力の先頭を指定の行数だけ表示します。デフォルトの行数は 10 です。
human 例： `show version | human`	**terminal output xml** コマンドを使用して出力形式が XML に設定されている場合に、出力を通常形式で表示します。 （注） **terminal output xml** コマンドには、1.0 バージョンと 6.x / 7.x バージョンがあり、XML の検証時にスキーマ（.xsd）ファイルを設定するために使用されます。1.0 を設定すると、スイッチは /isan/etc/schema/ <comp>.xsd ファイルを検証し、6.x /7.x を設定すると、スイッチは、スイッチが実行しているイメージのバージョンに基づき、/isan/ect/schema/6.x/ <comp>.xsd ファイルおよび /isan/ect/schema/7.x/ <comp>.xsd ファイルを検証します。
include*string* 例： `show interface brief | include up`	検索ストリングが含まれている行をすべて表示します。検索ストリングは大文字と小文字が区別されます。
last [*lines*] 例： `show logging logfile | last 50`	出力の末尾を指定の行数だけ表示します。デフォルトの行数は 10 です。

キーワードの構文	説明
no-more 例： `show interface brief | no-more`	途中で停止せずにすべての出力を表示します。画面の一番下に「`--More--`」プロンプトは表示されなくなります。
sscp*SSH-connection-namefilename* 例： `show version | sscp MyConnection show_version_output`	Streaming Secure Copy（sscp）を使用して出力を名前付き SSH 接続にリダイレクトします。名前付きの SSH 接続は、**ssh name** コマンドを使用して作成できます。
wc [**bytes** | **lines** | **words**] 例： `show file testoutput | wc bytes`	文字数、行数、または単語数を表示します。デフォルトでは、行数、単語数、および文字数を表示します。
xml 例： `show version | xml`	出力を XML 形式で表示します。

diff ユーティリティ

show コマンドからの出力と、そのコマンドを以前に実行したときの出力を比較できます。

diff-clean [**all-session**] [**all-users**]

次の表で、diff ユーティリティのキーワードについて説明します。

キーワード	説明
all-sessions	現在のユーザのすべてのセッション（過去および現在のセッション）から比較の一時ファイルが削除されます。
all-users	すべてのユーザのすべてのセッション（過去および現在のセッション）から比較の一時ファイルが削除されます。

Cisco NX-OS ソフトウェアは、現在および以前のすべてのユーザ セッションに対する **show** コマンドの最新の出力について、一時ファイルを作成します。これらの一時ファイルを削除するには、**diff-clean** コマンドを使用します。

diff-clean [**all-sessions** | **all-users**]

デフォルトでは、**diff-clean** コマンドによって現在のユーザのアクティブ セッションに対する一時ファイルが削除されます。**all-sessions** キーワードを指定すると、現在のユーザの過去および現在の全セッションに対する一時ファイルが削除されます。**all-users** キーワードを指定すると、すべてのユーザの過去および現在の全セッションに対する一時ファイルが削除されます。

grep および egrep ユーティリティ

Global Regular Expression Print（grep）および Extended grep（egrep）コマンドライン ユーティリティを使用して、**show** コマンド出力をフィルタリングすることができます。

grep と egrep の構文は次のとおりです。

{**grep** | **egrep**} [**count**] [**ignore-case**] [**invert-match**] [**line-exp**] [**line-number**] [**next** *lines*] [**prev** *lines*] [**word-exp**] *expression*}]

次の表に、**grep** と **egrep** のパラメータを示します。

表 ***10*** : ***grep*** および ***egrep*** のパラメータ

パラメータ	説明
count	一致した行の合計数のみを表示します。
ignore-case	一致した行の大文字と小文字の相違を無視するように指定します。
invert-match	表現が一致しない行を表示します。
line-exp	行に完全に一致する行だけを表示します。
line-number	一致した各行の前の行番号を表示するように指定します。
next*lines*	一致した行の後に表示する行数を指定します。デフォルトは 0 です。有効な範囲は 1 ～ 999 です。
prev*lines*	一致した行の前に表示する行数を指定します。デフォルトは 0 です。有効な範囲は 1 ～ 999 です。
word-exp	単語が完全に一致する行だけを表示します。
expression	出力を検索するための正規表現を指定します。

less ユーティリティ

less ユーティリティを使用すると、**show** コマンドの出力内容を一度に 1 ページずつ表示することができます。「:」プロンプトにおいて **less** コマンドを入力できます。使用可能な **less** コマンドをすべて表示するには、「:」プロンプトで h を入力します。

sed ユーティリティ

ストリーム エディタ（sed）ユーティリティを使用して、次のように **show** コマンドの出力をフィルタリングしたり、操作したりすることができます。

sed *command*

command 引数には、sed ユーティリティのコマンドを指定します。

sort ユーティリティ

sort ユーティリティを使用して、**show** コマンド出力をフィルタリングできます。

sort ユーティリティの構文は次のとおりです。

sort [**-M**] [**-b**] [**-d**] [**-f**] [**-g**] [**-i**] [**-k***field-number*[**.***char-position*][*ordering*]] [**-n**] [**-r**] [**-t***delimiter*] [**-u**]
次の表に、sort ユーティリティのパラメータを示します。

表 *11* : ***sort*** ユーティリティのパラメータ

パラメータ	説明
-M	月でソートします。
-b	先頭のブランク（空白文字）を無視します。デフォルトのソートでは、先頭のブランクが考慮されます。
-d	ブランクと英数字のみを比較してソートします。デフォルトのソートでは、すべての文字が考慮されます。
-f	小文字を大文字として処理します。
-g	一般的な数値を比較してソートします。
-i	印刷可能な文字だけを使用してソートします。デフォルトのソートでは、印刷不可能な文字も考慮されます。
-k*field-number*[**.***char-position*][*ordering*]	キー値に従ってソートします。デフォルトのキー値はありません。
-n	数値ストリングの値に従ってソートします。
-r	ソート結果の順序を逆にします。デフォルトのソート出力は昇順です。

パラメータ	説明
-t*delimiter*	指定のデリミタを使用してソートします。デフォルトのデリミタは空白文字です。
-u	ソート結果から重複行を取り除きます。ソート出力では重複行が表示されます。

--More-- プロンプトからの検索およびフィルタリング

show コマンド出力の `--More--` プロンプトで出力の検索やフィルタリングを実行できます。

次の表に、`--More-` プロンプト コマンドの説明を示します。

表 *12*： ***--More--*** プロンプト コマンド

コマンド	説明
[*lines*]<space>	指定した行数か現在の画面サイズ分の出力行を表示します。
[*lines*]**z**	指定した行数か現在の画面サイズ分の出力行を表示します。*lines* 引数を使用すると、その値が新しいデフォルトの画面サイズになります。
[*lines*]<return>	指定した行数か現在のデフォルトの行数で出力行を表示します。初期のデフォルトは 1 行です。オプションの *lines* 引数を使用すると、その値が、このコマンドで表示する新しいデフォルトの行数になります。
[*lines*]**d** または [*lines*]Ctrl+Shift+D	指定した行数か現在のデフォルトの行数で出力行をスクロールします。初期のデフォルトは11行です。オプションの *lines* 引数を使用すると、その値が、このコマンドで表示する新しいデフォルトの行数になります。
q または **Q** または Ctrl+C	`--More--` プロンプトを終了します。
[*lines*]**s**	指定した行数か現在のデフォルトの行数だけ出力をスキップし、1 画面分の出力行を表示します。デフォルトは 1 行です。

コマンド	説明
[*lines*]**f**	指定した画面数か現在のデフォルトの画面数だけ出力をスキップし、1 画面分の出力行を表示します。デフォルトは 1 画面です。
=	現在の行番号を表示します。
[*count*]/*expression*	正規表現に一致する行までスキップし、1 画面分の出力行を表示します。式が複数回出現する行を検索するには、オプションの *count* 引数を使用します。このコマンドにより、他のコマンドで使用可能な現在の正規表現が設定されます。
[*count*]**n**	現在の正規表現に次に一致する行までスキップし、1 画面分の出力行を表示します。一致を通り越してスキップするには、オプションの *count* 引数を使用します。
{**!** | **:!**[*shell-cmd*]}	*shell-cmd* 引数で指定されたコマンドをサブシェルで実行します。
.	前のコマンドを繰り返します。

コマンド履歴の使用方法

Cisco NX-OS ソフトウェアの CLI では、現在のユーザ セッションのコマンド履歴にアクセスできます。コマンドを呼び出し、そのまま再実行できます。また、実行前に修正することも可能です。コマンド履歴はクリアすることもできます。

コマンドの呼び出し

コマンド履歴内のコマンドを呼び出して、必要に応じて修正し、再入力できます。

次に、コマンドを呼び出して再入力する例を示します。

```
switch(config)# show cli history
0  11:04:07   configure terminal
1  11:04:28   show interface ethernet 2/24
2  11:04:39     interface ethernet 2/24
3  11:05:13       no shutdown
4  11:05:19     exit
5  11:05:25   show cli history
switch(config)# !1
switch(config)# show interface ethernet 2/24
```

Ctrl+P と Ctrl+N のキーストローク ショートカットを使用してコマンドを呼び出すこともできます。

CLI 履歴の再呼び出しの制御

CLI 履歴から再呼び出しするコマンドを制御するには、Ctrl+P キーストローク ショートカットと Ctrl+N キーストローク ショートカットを使用します。Cisco NX-OS ソフトウェアは、現在のコマンドモード以上のコマンドモードのすべてのコマンドを再呼び出しします。たとえば、グローバル コンフィギュレーション モードで作業をしている場合は、コマンド呼び出しキーストローク ショートカットを使用すると、EXEC モード コマンドとグローバル コンフィギュレーション モード コマンドの両方が呼び出されます。

CLI 編集モードの設定

Ctrl+P および Ctrl+N キーストローク ショートカットを使用して CLI 履歴からコマンドを呼び出し、コマンドを再発行する前に編集することができます。デフォルトの編集モードは、emacs です。編集モードを vi に変更できます。

手順の概要

1. [**no**] **terminal edit-mode vi** [**persist**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	[**no**] **terminal edit-mode vi** [**persist**] 例： `switch# terminal edit-mode vi`	ユーザ セッションの CLI 編集モードを vi に変更します。**persist** キーワードを使用すると、現在のユーザ名の設定がセッションをまたいで保持されます。 emacs の使用に戻すには、**no** を使用します。

コマンド履歴の表示

show cli history コマンドを使用して、コマンド履歴を表示できます。

show cli history コマンドの構文は次のとおりです。

デフォルトで表示される行数は 12 であり、出力にはコマンド番号とタイムスタンプが含まれます。

次に、コマンド履歴のデフォルトの行数を表示する例を示します。

```
switch# show cli history
```

次に、コマンド履歴の 20 行を表示する例を示します。

```
switch# show cli history 20
```

次に、コマンド番号とタイムスタンプなしでコマンド履歴内のコマンドだけを表示する例を示します。

```
switch(config)# show cli history unformatted
```

CLI 確認プロンプトのイネーブルまたはディセーブル

多くの機能について、作業を続行する前に確認を求めるプロンプトが、Cisco NX-OS ソフトウェアによって CLI に表示されます。これらのプロンプトをイネーブルにしたり、ディセーブルにしたりできます。デフォルトではイネーブルになっています。

手順の概要

1. **[no] terminal dont-ask [persist]**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**[no] terminal dont-ask [persist]** 例： `switch# terminal dont-ask`	CLI 確認プロンプトをディセーブルにします。**persist** キーワードを使用すると、現在のユーザ名の設定がセッションをまたいで保持されます。デフォルトではイネーブルになっています。 CLI 確認プロンプトをイネーブルにするには、コマンドの **no** 形式を使用します。

CLI 画面の色の設定

表示する CLI の色は次のように変更できます。

- 直前のコマンドが成功した場合は、プロンプトが緑色で表示されます。
- 直前のコマンドが失敗した場合は、プロンプトが赤色で表示されます。
- ユーザ入力は青色で表示されます。
- コマンド出力はデフォルトの色で表示されます。

デフォルトの色は、ターミナル エミュレータ ソフトウェアにより設定された色です。

手順の概要

1. **terminal color** [**evening**] [**persist**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**terminal color** [**evening**] [**persist**] 例： `switch# terminal color`	ターミナル セッションに対して CLI 画面の色を設定します。**evening** キーワードはサポートされません。**persist** キーワードを使用すると、現在のユーザ名の設定がセッションをまたいで保持されます。デフォルト設定は保持されません。

モジュールへのコマンドの送信

slot コマンドを使用して、スーパーバイザ モジュール セッションからモジュールにコマンドを直接送信できます。

slot の構文は次のとおりです。

slot*slot-number* [**quoted**] *command-string*

デフォルトでは、*command-string* 引数内のキーワードと引数はスペースで区切られます。モジュールに複数のコマンドを送信するには、スペース文字、セミコロン（;）、スペース文字でコマンドを区切ります。

quoted キーワードは、コマンド文字列の先頭と末尾に二重引用符（"）が使用されることを示します。スーパーバイザ モジュール セッションでだけサポートされている diff などのフィルタリング ユーティリティにモジュール コマンド出力をリダイレクトする場合は、このキーワードを使用します。

次に、モジュール情報を表示し、フィルタリングする例を示します。

```
switch# slot 2 show version | grep lc
```

次に、スーパーバイザモジュールセッションに関するモジュール情報をフィルタリングする例を示します。

```
switch# slot 2 quoted "show version" | diff
switch# slot 4 quoted "show version" | diff -c
*** /volatile/vsh_diff_1_root_8430_slot__quoted_show_version.old       Wed Apr 29 20:10:41
 2009
--- -   Wed Apr 29 20:10:41 2009
***************
*** 1,5 ****
! RAM 1036860 kB
! lc2
  Software
```

```
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
--- 1,5 ----
! RAM 516692 kB
! lc4
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
***************
*** 12,16 ****
  Hardware
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 34 second(s)

--- 12,16 ----
  Hardware
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 42 second(s)
```

BIOS ローダー プロンプト

スーパーバイザ モジュールの起動時に、特殊な BIOS イメージが、システム起動用の有効なキックスタート イメージを自動的にロードしたり、検索しようとしたりします。有効なキックスタート イメージが見つからない場合は、次の BIOS ローダー プロンプトが表示されます。

```
loader>
```

`<loader>` プロンプトから Cisco NX-OS ソフトウェアをロードする方法については、ご使用のデバイスの『Cisco Nexus troubleshooting guide』を参照してください。

CLI の使用例

ここでは、CLI の使用例を示します。

コマンド エイリアスの定義

次に、コマンド エイリアスを定義する例を示します。

```
cli alias name ethint interface ethernet
cli alias name shintbr show interface brief
cli alias name shintupbr shintbr | include up | include ethernet
```

次に、コマンド エイリアスを使用する例を示します。

```
switch# configure terminal
switch(config)# ethint 2/3
switch(config-if)#
```

CLI セッション変数の使用方法

$(*variable-name***)** 構文を使用して変数を参照できます。
次に、ユーザ定義の CLI セッション変数を参照する例を示します。

```
switch# show interface $(testinterface)
Ethernet2/1 is down (Administratively down)
  Hardware is 10/100/1000 Ethernet, address is 0000.0000.0000 (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 0 bytes/sec, 0 packets/sec
  5 minute output rate 0 bytes/sec, 0 packets/sec
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
```

システム定義のタイムスタンプ変数の使用方法

次の例では、**show** コマンド出力をファイルにリダイレクトするときに $(TIMESTAMP) を使用します。

```
switch# show running-config > rcfg.$(TIMESTAMP)
Preparing to copy....done
switch# dir
       12667     May 01 12:27:59 2008  rcfg.2008-05-01-12.27.59

Usage for bootflash://sup-local
8192 bytes used
20963328 bytes free
20971520 bytes total
```

コマンド スクリプトの実行

次の例では、スクリプト ファイル内で指定されている CLI コマンドを表示します。

```
switch# show file testfile
configure terminal
interface ethernet 2/1
no shutdown
end
show interface ethernet 2/1
```

次の例では、**run-script** コマンドの実行時の出力を表示します。

```
switch# run-script testfile
`configure terminal`
`interface ethernet 2/1`
`no shutdown`
`end`
`show interface ethernet 2/1 `
Ethernet2/1 is down (Link not connected)
  Hardware is 10/100/1000 Ethernet, address is 0019.076c.4dac (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is trunk
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters 1d26.2uh
  5 minute input rate 0 bytes/sec, 0 packets/sec
  5 minute output rate 0 bytes/sec, 0 packets/sec
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
```

CLI に関する追加情報

ここでは、CLI に関する追加情報について説明します。

CLI の関連資料

関連項目	マニュアル タイトル
Cisco NX-OS ライセンス設定	『*Cisco NX-OS Licensing Guide*』
コマンド リファレンス	『*Cisco Nexus 5000 Series NX-OS Command Reference*』