



コマンドラインインターフェースの概要

この章では、Cisco NX-OS ソフトウェア コマンドラインインターフェイス (CLI) について説明します。

この章は、次の内容で構成されています。

- [CLI プロンプトの概要, 2 ページ](#)
- [コマンド モード, 2 ページ](#)
- [特殊文字, 8 ページ](#)
- [キーストローク ショートカット, 9 ページ](#)
- [コマンドの短縮形, 11 ページ](#)
- [部分的なコマンド名の補完, 12 ページ](#)
- [コマンド階層での場所の特定, 13 ページ](#)
- [コマンドの no 形式の使用法, 13 ページ](#)
- [CLI 変数の設定, 14 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, 30 ページ](#)
- [コマンド履歴の使用法, 31 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, 33 ページ](#)
- [CLI 画面の色の設定, 34 ページ](#)
- [モジュールへのコマンドの送信, 34 ページ](#)

- [BIOS ローター プロンプト, 35 ページ](#)
- [CLI の使用例, 36 ページ](#)
- [CLI に関する追加情報, 38 ページ](#)
- [CLI 機能の履歴, 38 ページ](#)

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で設定したり、より具体的な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したりできます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： switch# configure terminal switch(config)#	グローバル コンフィギュレーション モードを開始します。 (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、『[Cisco MDS 9000 Family NX-OS Interfaces Configuration Guide](#)』と『[Cisco Nexus 7000 Series NX-OS Interfaces Command Reference](#)』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： switch# configure terminal switch(config)#	グローバル コンフィギュレーション モードを開始します。
ステップ 2	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、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、『Cisco Nexus 7000 Series NX-OS Interfaces Command Reference』を参照してください。

手順

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

	コマンドまたはアクション	目的
ステップ 2	interface <i>type number.subint</i> 例： <pre>switch(config)# interface ethernet 2/2.1 switch(config-subif)#</pre>	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。

次に、コマンドモードを保存し、復元する例を示します。

```
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

	コマンドまたはアクション	目的
ステップ 1	exit 例： <pre>switch(config-if)# exit switch(config)#</pre>	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。

	コマンドまたはアクション	目的
ステップ 2	end 例： <pre>switch(config-if)# end switch#</pre>	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： <pre>switch(config-if)# ^Z switch#</pre>	(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 を使用すると、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	アクセス方法	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。
VDC 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、 vdc コマンドを使用して、VDC 名を指定します。	switch(config-vdc)#	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。

モード	アクセス方法	プロンプト	終了方法
VRF 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、 vrf コマンドを使用して、ルーティングプロトコルを指定します。	switch(config-vrf) #	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。
デフォルト以外の VDC に対する EXEC	EXEC モードで、 switchto vdc コマンドを使用して、VDC を指定します。	switch-vdc2#	終了してデフォルトの VDC に戻るには、 exit コマンドまたは switchback コマンドを使用します。
デフォルト以外の VRF に対する EXEC	EXEC モードで、 routing-context vrf コマンドを使用して、VRF を指定します。	switch%red#	終了してデフォルトの VRF に戻るには、 routing-context vrf default コマンドを使用します。

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

文字	説明
{ }	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続け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。
Ctrl+G	コマンド文字列を削除せずに、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。

キーストローク	説明
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。すると、カーソルが1つ右の文字に移動します。
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。
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。

キーストローク	説明
Tab	<p>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。</p> <p>タブを使用して、次の項目を入力します。</p> <ul style="list-style-type: none"> • 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 <p>例 :</p> <pre>switch(config)# xm<Tab> switch(config)# xml<Tab> switch(config)# xml server</pre> <p>例 :</p> <pre>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</pre> <p>例 :</p> <pre>switch# cd bootflash:<Tab> bootflash: bootflash://sup-1/ bootflash:/// bootflash://sup-2/ bootflash://module-5/ bootflash://sup-active/ bootflash://module-6/ bootflash://sup-local/</pre> <p>例 :</p> <pre>switch# cd bootflash://mo<Tab> bootflash://module-5/ bootflash://module-6/cv switch# cd bootflash://module-</pre>

コマンドの短縮形

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、入力された文字列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。これにより、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

	コマンドまたはアクション	目的
ステップ 1	<p>where detail</p> <p>例 :</p> <pre>switch# configure terminal switch(config)# interface mgmt0 switch(config-if)# where detail mode: conf interface mgmt0 username: admin vdc: switch routing-context vrf: default</pre>	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) `TIMESTAMP` 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行う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

	コマンドまたはアクション	目的
ステップ 1	<p>cli var name <i>variable-name</i> <i>variable-text</i></p> <p>例： switch# cli var name testinterface ethernet 2/1</p>	<p>CLI セッション変数を設定します。 <i>variable-name</i> 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。 <i>variable-text</i> 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。</p>
ステップ 2	<p>show cli variables</p> <p>例： switch# show cli variables</p>	<p>(任意) CLI 変数の設定を表示します。</p>

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

	コマンドまたはアクション	目的
ステップ 1	<p>configure terminal</p> <p>例： switch# configure terminal switch(config)#</p>	<p>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。</p>

	コマンドまたはアクション	目的
ステップ 2	cli var name <i>variable-name</i> <i>variable-text</i> 例： <pre>switch(config)# cli var name testinterface ethernet 2/1</pre>	CLI 固定変数を設定します。変数名は、英数字ストリングで指定します。大文字と小文字が区別されます。変数名の先頭を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： <pre>switch(config)# exit switch#</pre>	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： <pre>switch# show cli variables</pre>	(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： <pre>switch(config)# copy running-config startup-config</pre>	(任意) 実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

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

	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。

	コマンドまたはアクション	目的
ステップ 5	copy running-config startup-config 例： <pre>switch# copy running-config startup-config</pre>	(任意) 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

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

	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： <pre>switch# terminal alias shintbr show interface brief</pre>	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。ユーザアカウントであとで使用するよう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 キーワードは短縮しないでください。

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

	コマンドまたはアクション	目的
ステップ 1	<p>run-script [bootflash: slot0: volatile:]filename</p> <p>例 :</p> <pre>switch# run-script testfile</pre>	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
<code>\b</code>	バック スペースを挿入します。
<code>\c</code>	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。
<code>\f</code>	フォーム フィード文字が挿入されます。
<code>\n</code>	改行文字が挿入されます。
<code>\r</code>	テキスト行の最初に戻ります。
<code>\t</code>	水平タブ文字が挿入されます。
<code>\v</code>	垂直タブ文字が挿入されます。
<code>\\</code>	バックslash文字が表示されます。
<code>\nnn</code>	対応する ASCII 8 進文字が表示されます。

手順

	コマンドまたはアクション	目的
ステップ 1	echo [backslash-interpret] [text] 例： <pre>switch# echo This is a test. This is a test.</pre>	backslash-interpret キーワードは、テキストストリングにフォーマットオプションが含まれることを示します。 <i>text</i> 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

	コマンドまたはアクション	目的
ステップ 1	sleep seconds 例： <pre>switch# sleep 30</pre>	遅延をある秒数発生させます。有効な範囲は 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
<pre>switch# clock ? set HH:MM:SS Current Time switch# clock</pre>	<p>EXEC モードの clock コマンドのコマンド構文を表示します。</p> <p>switch の出力は、clock コマンドの使用に set キーワードが必要であることを示しています。</p>
<pre>switch# clock set ? WORD HH:MM:SS Current Time switch# clock set</pre>	<p>時刻を設定するためのコマンド構文を表示します。</p> <p>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を、および時刻のフォーマット方法を示しています。</p>
<pre>switch# clock set 13:32:00<CR> % Incomplete command switch#</pre>	<p>現在時刻を追加します。</p> <p>CLI は、コマンドが不完全であることを示しています。</p>
<pre>switch# <Ctrl-P> switch# clock set 13:32:00</pre>	<p>入力した直前のコマンドを表示します。</p>
<pre>switch# clock set 13:32:00 ? <1-31> Day of the month switch# clock set 13:32:00</pre>	<p>clock set コマンドに対する追加の引数を表示します。</p>
<pre>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</pre>	<p>clock set コマンドに対する追加の引数を表示します。</p>
<pre>switch# clock set 13:32:00 18 April 08<CR> % Invalid input detected at '^' marker.</pre>	<p>クロック設定に日付を追加します。</p> <p>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。</p>
<pre>switch# clock set 13:32:00 18 April ? <2000-2030> Enter the year (no abbreviation) switch# clock set 13:32:00 18 April</pre>	<p>この年に対応する適切な引数を表示します。</p>
<pre>switch# clock set 13:32:00 18 April 2008<CR> switch#</pre>	<p>clock set コマンドの正しい構文を入力します。</p>

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.` (文字 `a` の後にピリオド) は、ピリオド文字の特別な意味を使用して、文字 `a` の後に任意の単一文字が続くストリングと一致します。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
<code>^</code>	ストリングの先頭と一致します。
<code>\$</code>	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`」で始まる任意のストリングと一致し、`sole$` は「`sole`」で終わる任意のストリングと一致します。



(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。これらのオプションは、CLI 状況依存ヘルプ機能を使用して表示できます。

```
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 <i>string</i> 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
<p>cut [-d <i>character</i>] {-b -c -f -s}</p> <p>例： show file testoutput cut -b 1-10</p>	<p>一部の出力行のみを表示します。一定のバイト数 (-b)、文字数 (-vcut [-d <i>character</i>] {-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、行の表示にデリミタが含まれないようにします。</p>
<p>end <i>string</i></p> <p>例： show running-config end interface</p>	<p>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。</p>
<p>exclude <i>string</i></p> <p>例： show interface brief exclude down</p>	<p>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。</p>
<p>head [<i>lines lines</i>]</p> <p>例： show logging logfile head lines 50</p>	<p>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。</p>
<p>human</p> <p>例： show version human</p>	<p>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。</p>
<p>include <i>string</i></p> <p>例： show interface brief include up</p>	<p>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。</p>
<p>last [<i>lines</i>]</p> <p>例： show logging logfile last 50</p>	<p>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。</p>
<p>no-more</p> <p>例： show interface brief no-more</p>	<p>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。</p>
<p>sscp <i>SSH-connection-name filename</i></p> <p>例： show version sscp MyConnection show_version_output</p>	<p>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。</p>
<p>wc [<i>bytes lines words</i>]</p> <p>例： show file testoutput wc bytes</p>	<p>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。</p>

キーワードの構文	説明
xml 例 : <code>show version xml</code>	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。



注意

show tech-support コマンドなど、出力が非常に長い **show** コマンドには、diff ユーティリティを使用しないでください。

diff ユーティリティの構文は次のとおりです。

diff [**--left-column**] [**-B**] [**-I**] [**-W columns**] [**-b**] [**-c lines**] [**-I**] [**-q**] [**-s**] [**-y**] [**again**] [**echo**]

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。

表 10: diff ユーティリティのキーワード

キーワード	説明
--left-column	並列形式の 2 つの共通行の左側のカラムだけを印刷します。
-B	ブランク行の挿入または削除のみを行う変更を無視します。
-I	正規表現に一致する行の挿入または削除のみを行う変更を無視します。
-W columns	並列形式の出力カラムの幅を指定します。有効な範囲は 0 ~ 4294967295 です。
-b	スペースの量に対する変更を無視します。デフォルトでは、スペースの相違が表示されます。
-c lines	表示されるコンテキストの行数を設定します。デフォルトの行数は 3 です。有効な範囲は 0 ~ 4294967295 です。
-I	大文字と小文字の相違を無視します。デフォルトでは、大文字と小文字の相違が報告されます。

キーワード	説明
-q	ファイルが異なるかどうかを示しますが、詳細な相違は表示されません。デフォルトでは、相違が表示されます。
-s	2つの出力が同じかどうかを示します。デフォルトでは、出力が同じであることを示す表示はありません。
-y	並列形式を使用して、出力の相違を示します。デフォルトでは、古い出力行が最初に表示され、次に現在の出力行が続きます。
again	新規の出力ファイルを作成しません。古いファイルを使用して、表示オプションを変更するか、フィルタを追加します。
echo	現在のコマンド出力をエコーします。このキーワードは、以前のコマンド出力がない場合のみ有効です。

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 11 : *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。デフォルト値は 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。デフォルト値は 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。
<i>expression</i>	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して、**show** コマンド出力の内容を 1 画面ずつ表示できます。「:」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) ユーティリティを次のように使用して、**show** コマンド出力のフィルタリングや操作を実行できます。

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の説明を示します。

表 12: 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。
-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 13: --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。 <i>lines</i> 引数を使用すると、その値が新しいデフォルト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 <i>lines</i> 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 <i>lines</i> 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。
[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。

コマンド	説明
<code>[count]/expression</code>	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。正規表現の複数回の繰り返しで行を検索する場合は、オプションの <code>count</code> 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。
<code>[count]n</code>	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。複数の一致をスキップする場合は、オプションの <code>count</code> 引数を使用します。
<code>{! :![shell-cmd]}</code>	<code>shell-cmd</code> 引数に指定し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。デフォルトでは、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。**terminal history no-exec-in-config** コマンドを使用すると、コンフィギュレーションモードのときに、EXEC モード コマンドの再呼び出しを回避できます。

手順

	コマンドまたはアクション	目的
ステップ 1	[no] terminal history no-exec-in-config 例： <pre>switch# terminal history no-exec-in-config</pre>	コンフィギュレーションモードで呼び出しキーストローク ショートカットを使用するときに EXEC コマンドを除外するよう CLI 履歴を設定します。デフォルトでは、EXEC コマンドが呼び出されます。デフォルト値に戻すには、このコマンドの no 形式を使用します。

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

	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： <pre>switch# terminal edit-mode vi</pre>	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間で保持されます。 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。

show cli history [*lines*] [**config-only** | **exec-only** | **this-mode-only**] [**unformatted**]

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、コマンド履歴内のコンフィギュレーション コマンドだけを表示する例を示します。

```
switch(config)# show cli history config-only
```

次に、コマンド履歴内の EXEC コマンドだけを表示する例を示します。

```
switch(config)# show cli history exec-only
```

次に、現在のコマンドモードに関する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-if)# show cli history this-mode-only
```

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

	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間で保持されます。 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送信された色です。

手順

	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間で保持されます。 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、コマンドストリング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上で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>
```

Cisco NX-OS ソフトウェアを loader> プロンプトからロードする方法については、『Cisco Nexus 7000 Series NX-OS Troubleshooting Guide』『Cisco MDS 9000 Family NX-OS 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	参照先
Cisco NX-OS のライセンス	『Cisco NX-OS Licensing Guide』
コマンドリファレンス	『Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference』

CLI 機能の履歴

次の表に、この機能のリリースの履歴を示します。

表 14: CLI 機能の履歴

機能名	リリース	機能情報
コマンドモードの保存および復元	4.1(2)	push コマンドと pop コマンドを追加しました。
永続的な CLI 変数	4.1(2)	デバイスをリロードしても維持される CLI 変数のサポートを追加しました。
show コマンドの出力のフィルタリング	4.1(2)	show コマンドの出力のフィルタリングに、 end キーワードを追加しました。

機能名	リリース	機能情報
I/O モジュール コマンド	4.2(1)	スーパーバイザ モジュールセッションから I/O モジュールにコマンドを送信できるようになりました。
コマンドエイリアス	4.2(1)	show コマンドのフィルタリングにコマンドのエイリアスが使用できるようになりました。 ユーザセッションに対してコマンドのエイリアスが使用できるようになりました。
コマンド履歴	4.2(1)	show cli history コマンドの出力を制限するオプションのキーワードを追加しました。
確認プロンプト	4.2(1)	コマンドの確認プロンプトをイネーブルまたはディセーブルにできるようになりました。
端末の色	4.2(1)	端末画面で CLI の要素に使用する色を変更できるようになりました。

