



## CLI の概要

---

この章では、Cisco NX-OS ソフトウェアの CLI について説明します。

この章で説明する内容は、次のとおりです。

- [CLI プロンプトの概要 \(p.3-2\)](#)
- [コマンドモード \(p.3-3\)](#)
- [特殊文字 \(p.3-7\)](#)
- [キーストローク ショートカット \(p.3-8\)](#)
- [コマンドの省略 \(p.3-10\)](#)
- [コマンド階層内の位置の確認 \(p.3-10\)](#)
- [コマンドの No 形式の使用法 \(p.3-11\)](#)
- [CLI 変数の設定 \(p.3-12\)](#)
- [コマンドスクリプト \(p.3-15\)](#)
- [状況依存ヘルプ \(p.3-17\)](#)
- [正規表現について \(p.3-19\)](#)
- [show コマンド出力の検索およびフィルタリング \(p.3-21\)](#)
- [BIOS loader> プロンプト \(p.3-23\)](#)
- [CLI の使用例 \(p.3-24\)](#)
- [詳細な資料 \(p.3-27\)](#)

## CLI プロンプトの概要

デバイスに正常にアクセスすると、コンソールポートまたはリモートワークステーションのターミナルウィンドウに CLI プロンプトが表示されます (例 3-1 を参照)。

### 例 3-1 最初の CLI プロンプト

```
User Access Verification
login: admin
Password:<password>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2008, 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 ホスト名は変更できます (「デバイス ホスト名の変更」 [p.5-3] を参照)。

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

## コマンドモード

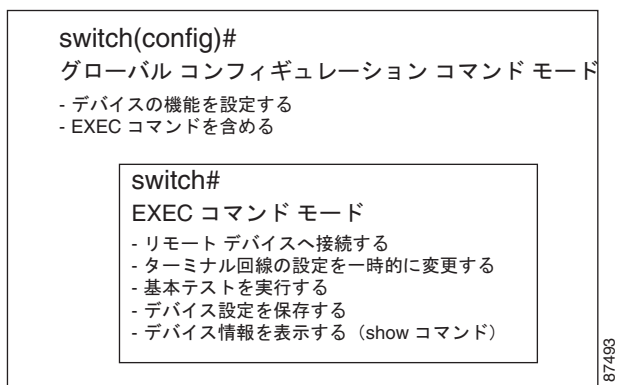
ここでは、次の内容について説明します。

- コマンドモードについて (p.3-3)
- EXEC コマンドモード (p.3-3)
- グローバル コンフィギュレーション コマンドモード (p.3-3)
- 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 (p.3-4)
- 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 (p.3-5)
- コンフィギュレーション コマンドモードの終了 (p.3-6)
- コマンドモードの概要 (p.3-6)

## コマンドモードについて

Cisco NX-OS CLI は、いくつかのコマンドモードに分けられます。ユーザが実行できるアクションは、コマンドモードによって定義されます。コマンドモードは「ネスト」されており、順番にアクセスしていく必要があります。別のコマンドモードへナビゲートするたびに、使用できるコマンド数が増えます。下位コマンドモードから、上位コマンドモードのすべてのコマンドにアクセスできます。たとえば、**show** コマンドは、どのコンフィギュレーション コマンドモードからでも使用できます。図 3-1 に、コマンドアクセスが EXEC モードからグローバル コンフィギュレーションモードへどのように形成されているかを示します。

図 3-1 コマンドモード




## EXEC コマンドモード

ログインすると、最初に Cisco NX-OS ソフトウェアによって 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、ユーザがデバイス設定に保存していない処理が実行されるその他のコマンドなどがあります。

## 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、最も広範囲のコマンドへアクセスできます。「グローバル」という用語は、デバイス全体に影響する特性または機能を意味します。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、デバイス全体を設定できます。あるいは、より限定されたコンフィギュレーション モードを開始すると、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できます。

グローバル コンフィギュレーション モードにアクセスするには、次の手順を実行します。

	コマンド	目的
ステップ 1	<code>configure terminal</code>	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。
	<pre>switch# configure terminal switch(config)#</pre>	 <p>(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 モードになったことが示されます。</p>

## 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 モードから開始する特定のコンフィギュレーション モードの例として、インターフェイス コンフィギュレーション モードがあります。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て、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 ベースでイネーブルにする必要のある機能は、数多くあります。インターフェイス コンフィギュレーション コマンドを実行すると、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) といった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 NX-OS Interfaces Configuration Guide*』Release 4.0を参照してください。

インターフェイス コマンドの詳細については、『*Cisco NX-OS Interfaces Command Reference*』Release 4.0を参照してください。

### 手順の要約

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

### 詳細な手順

	コマンド	目的
ステップ 1	<code>configure terminal</code>	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。
	<pre>switch# configure terminal switch(config)#</pre>	
ステップ 2	<code>interface type number</code>	設定対象のインターフェイスを指定します。
	<pre>switch(config)# interface ethernet 2/2 switch(config-if)#</pre>	<p>CLI によって、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 モードが開始されます。</p>  <p>(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 モードになったことが示されます。</p>

## 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、単一の物理インターフェイス上に複数の仮想インターフェイスを設定できます。プロトコルからは、サブインターフェイスが別々の物理インターフェイスとして示されます。

また、サブインターフェイスにより、単一のインターフェイス上でプロトコルの複数のカプセル化を使用できます。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 に関連付けられます。


サブインターフェイスの設定の詳細については、『Cisco NX-OS Interfaces Configuration Guide』Release 4.0 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 NX-OS Interfaces Command Reference』Release 4.0 を参照してください。

### 手順の要約


1. `configure terminal`
2. `interface type number.subint`

### 詳細な手順

	コマンド	目的
ステップ 1	<code>configure terminal</code>  switch# <code>configure terminal</code> 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	<code>interface type number.subint</code>  switch(config)# <code>interface ethernet 2/2.1</code> switch(config-subif)#	<p>設定対象の VLAN インターフェイスを指定します。</p> <p>CLI によって、指定した VLAN インターフェイスのサブインターフェイス コンフィギュレーション モードが開始されます。</p> <p> (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 モードになったことが示されます。</p>

## 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、次の作業のいずれかを行います。

コマンド	目的
<pre>exit  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。
<pre>end  switch(config)# end switch#</pre>	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。
<pre>Ctrl-z  switch(config)# ^z switch#</pre>	<p>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。</p> <p> <b>注意</b> 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。コンフィギュレーションモードを終了する場合は、<b>exit</b> または <b>end</b> コマンドを使用することを推奨します。</p>

## コマンドモードの概要

表 3-1 に、主なコマンドモードの概要を示します。

表 3-1 コマンドモードの概要

モード	アクセス方式	プロンプト	終了方式
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、 <b>exit</b> 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モードから、 <b>configure terminal</b> 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、 <b>end</b> または <b>exit</b> コマンドを使用するか、 <b>Ctrl-Z</b> 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から、インターフェイスコマンドを使用して、 <b>interface</b> コマンドによっ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、 <b>exit</b> コマンドを使用します。  終了してEXECモードに戻るには、 <b>end</b> コマンドを使用するか、 <b>Ctrl-Z</b> 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から、 <b>interface</b> 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、 <b>exit</b> コマンドを使用します。  終了してEXECモードに戻るには、 <b>end</b> コマンドを使用するか、 <b>Ctrl-Z</b> を押します。

表 3-1 (続き) コマンド モードの概要 (続き)

モード	アクセス方式	プロンプト	終了方式
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 モードから、 <b>vdc</b> 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、 <b>exit</b> コマンドを使用します。  終了して EXEC モードに戻るには、 <b>end</b> コマンドを使用するか、 <b>Ctrl-Z</b> を押します。
ルータ 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 モードから、 <b>router</b> 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-router)#	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、 <b>exit</b> コマンドを使用します。  終了して EXEC モードに戻るには、 <b>end</b> コマンドを使用するか、 <b>Ctrl-Z</b> を押します。
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 モードから、 <b>router</b> 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、 <b>exit</b> コマンドを使用します。  終了して EXEC モードに戻るには、 <b>end</b> コマンドを使用するか、 <b>Ctrl-Z</b> を押します。
非デフォルト VDC の EXEC	EXEC モードから、 <b>switchto vdc</b> 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 VDC に戻るには、 <b>exit</b> コマンドまたは <b>switchback</b> コマンドを使用します。
非デフォルト VRF の EXEC	EXEC モードから、 <b>routing-context vrf</b> 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 VRF に戻るには、 <b>routing-context vrf default</b> コマンドを使用します。

## 特殊文字

表 3-2 に、Cisco NX-OS のテキスト スtring 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用なコンテキストでのみ使用します。

表 3-2 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

表 3-3 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 3-3 キーストローク ショートカット

キー	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 文字分だけ後退させます。 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 文字分だけ進めます。
Ctrl-G	コマンド スtring 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	ターミナル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	カーソルの左にある文字を、カーソル位置の文字と置き換えます。
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	その後続くキーストロークの特別な意味を削除します。たとえば、正規表現の中で Ctrl-V を押してから疑問符 (?) を入力する場合があります。
Ctrl-W	カーソルの左にあるワードを削除します。
Ctrl-X, H	入力したコマンドの履歴のリストを表示します。  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タリを呼び出します (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。
↑	コマンド履歴の前のコマンドを表示します。
↓	コマンド履歴の次のコマンドを表示します。
← →	コマンド スtring を見つけるために、コマンド履歴内でカーソルを前または次のコマンドに移動します。
?	使用可能なコマンドのリストを表示します。



表 3-3 キーストローク ショートカット (続き)

キー	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>Tab キーを使用して補完できる項目は、次のとおりです。</p> <ul style="list-style-type: none"> <li>• コマンド名</li> <li>• ファイル システム内のスキーム名</li> <li>• ファイル システム内のサーバ名</li> <li>• ファイル システム内のファイル名</li> </ul> <p><b>例</b></p> <pre>switch(config)# xm&lt;Tab&gt; switch(config)# xml &lt;Tab&gt; switch(config)# xml server</pre> <p><b>例</b></p> <pre>switch(config)# c&lt;Tab&gt; callhome      class-map      clock      cts cdp           cli           control-plane</pre> <p>switch(config)# cl&lt;Tab&gt; class-map cli clock switch(config)# cla&lt;Tab&gt; switch(config)# class-map </p>
	<p><b>例</b></p> <pre>switch# cd bootflash:&lt;Tab&gt; bootflash:      bootflash://sup-1/ bootflash://sup-remote/ bootflash:///      bootflash://sup-2/ bootflash://sup-standby/ bootflash://module-5/  bootflash://sup-active/ bootflash://module-6/  bootflash://sup-local/</pre> <p><b>例</b></p> <pre>switch# cd bootflash://mo&lt;Tab&gt; bootflash://module-5/  bootflash://module-6/ cvswitch# cd bootflash://module-</pre>

## コマンドの省略

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る文字数を含める必要があります。コマンド入力の問題に遭遇した場合は、システム プロンプトを確認するとともに、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たり、間違った構文を使用している可能性があります。

表 3-4 に、コマンド省略形の例を示します。

表 3-4 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

## コマンド階層内の位置の確認

一部の機能には複数レベルの深さで入れ子になっているコンフィギュレーション サブモード階層構造があります。この場合、現在の作業コンテキスト (PWC) に関する情報を表示できます。

### 手順の要約

#### 1. where detail

### 詳細な手順

コマンド	目的
ステップ 1 where detail  <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 形式の使用法

ほとんどすべてのコンフィギュレーション コマンドには、機能をディセーブルしたり、デフォルト値に戻したり、設定を削除したりできる **no** 形式があります。Cisco NX-OS 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** 形式の機能を示しています。

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 bgp
switch(config)# no feature bgp
```

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)# role feature-group name security
switch(config-role-featuregrp)# feature aaa
switch(config-role-featuregrp)# feature radius
switch(config-role-featuregrp)# feature tacacs
switch(config-role-featuregrp)# exit
switch(config)# show role feature-group

feature group: L3
feature: router-bgp
feature: router-eigrp
feature: router-isis
feature: router-ospf
feature: router-rip

feature group: security
feature: aaa
feature: radius
feature: tacacs
switch(config)# no role feature-group name security
switch(config)# show role feature-group

feature group: L3
feature: router-bgp
feature: router-eigrp
feature: router-isis
feature: router-ospf
feature: router-rip
```

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# system standby manual-boot
system standby manual-boot option enabled
switch# system no standby manual-boot
system standby manual-boot option disabled
```

## CLI 変数の設定

ここでは、次の内容について説明します。

- [CLI 変数の概要 \(p.3-12\)](#)
- [CLI 変数の設定 \(p.3-12\)](#)

## CLI 変数の概要

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、次のように使用できます。

- コマンドラインで直接入力します。
- **run-script** コマンドを使用して開始されたスクリプトに渡します。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（「[コマンド スクリプトの実行](#)」[\[p.3-15\]](#)を参照）。

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。
- 現在のセッションの間だけ存在できます。

Cisco NX-OS は1つの事前定義の変数、TIMESTAMP をサポートします。この変数は YYYY-MM-DD-HH.MM.SS フォーマットのコマンドの実行時間を参照します。



(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 変数の設定

CLI セッション変数を定義し、CLI セッションの持続時間の間だけ持続することができます。CLI セッション変数は周期的に実行するスクリプトに役立ちます。たとえば、\$(*variable-name*) のように、名前をカッコで囲んでその前にドル記号 (\$) を付けることで変数を参照できます。

### 手順の要約

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

### 詳細な手順

	コマンド	目的
ステップ 1	<pre>cli var name variable-name variable-text</pre> <pre>switch# cli var name testinterface ethernet 2/1</pre>	CLI セッション変数を設定します。 <i>variable-name</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、最大文字数は 31 です。 <i>variable-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、スペースを含めることができ、最大文字数は 200 です。
ステップ 2	<pre>show cli variables</pre> <pre>switch# show cli variables</pre>	(任意) CLI 変数設定を表示します。

## コマンドエイリアス

コマンドエイリアスを定義して、頻繁に使用されるコマンドを置き換えられます。コマンドエイリアスは、コマンド構文の全部または一部を表示できます。



(注)

Cisco NX-OS ソフトウェアは、1つのデフォルトエイリアス、**alias** を提供します。これは、ユーザー定義のすべてのエイリアスを表示します。

ここでは、次の内容について説明します。

- [コマンドエイリアスについて \(p.3-13\)](#)
- [コマンドエイリアスの定義 \(p.3-13\)](#)

## コマンドエイリアスについて

コマンドエイリアスは、次の特性をサポートします。

- 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すれば再起動後も継続します。
- 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直ちに有効になります。
- **show cli alias** コマンドのエイリアスであるデフォルトのコマンドエイリアス **alias** は、削除または変更ができません。
- エイリアスのネスティングは最大深度 1 まで許容されます。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常にコマンドラインの最初のコマンドキーワードに取って代わります。
- すべて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で CLI 変数を参照すると、エイリアスには変数の参照ではなく変数の現在値が表示されます。

## コマンドエイリアスの定義

よく使用されるコマンドのコマンドエイリアスを定義できます。

### 手順の要約

1. **configure terminal**
2. **cli alias name alias-name alias-text**
3. **exit**
4. **alias**
5. **copy running-config startup-config**

## ■ コマンドエイリアス

## 詳細な手順

	コマンド	目的
ステップ 1	<code>configure terminal</code>  switch# <code>configure terminal</code> 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	<code>cli alias name alias-name alias-text</code>  switch(config)# <code>cli alias name ethint</code> interface ethernet	コマンドエイリアスを設定します。エイリアス名は、大文字と小文字を区別しない英数字文字列であり、先頭は英字にする必要があります。最大文字数は 30 です。
ステップ 3	<code>exit</code>  switch(config)# <code>exit</code> 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	<code>alias</code>  switch# <code>alias</code>	(任意) コマンドエイリアス設定を表示します。
ステップ 5	<code>copy running-config startup-config</code>  switch# <code>copy running-config startup-config</code>	(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。

## コマンドスクリプト

コマンドスクリプトを作成して、複数のタスクを実行できます。

ここでは、次の内容について説明します。

- [コマンドスクリプトの実行 \(p.3-15\)](#)
- [ターミナルへの情報のエコー \(p.3-15\)](#)
- [ターミナルへの情報のエコー \(p.3-15\)](#)

### コマンドスクリプトの実行

ファイル形式のコマンドリストを作成して、CLI から実行できます。コマンドスクリプト内で CLI 変数を使用できます（「[CLI 変数の設定](#)」[p.3-12] を参照）。



(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。リモートデバイスでスクリプトファイルを作成して、それを Cisco NX-OS デバイスにコピーできます。ここでは、スクリプトファイルが bootflash: にあると想定しています。

#### 手順の要約

1. `run-script filename`

#### 詳細な手順

	コマンド	目的
ステップ 1	<code>run-script filename</code>  switch# run-script testfile	ファイル内のコマンドを実行します。

### ターミナルへの情報のエコー

コマンドスクリプトから特に有用な情報をターミナルにエコーできます。エコー対象のテキスト内の CLI 変数を参照できます。

#### 手順の要約

1. `echo [text]`

#### 詳細な手順

	コマンド	目的
ステップ 1	<code>echo [text]</code>  switch# echo This is a test. This is a test.	ターミナル上で情報を表示します。 <code>text</code> 引数は、英数字で指定します。大文字と小文字が区別され、ブランクを含めることができます。最大文字数は 200 です。デフォルトは、ブランク行です。

## ■ コマンドスクリプト

## コマンド処理の遅延

一定の時間、コマンド処理を遅延させることができます。これは、コマンドスクリプト内では特に有用です。

### 手順の要約

1. `sleep seconds`

### 詳細な手順

	コマンド	目的
ステップ 1	<code>sleep seconds</code>  <code>switch# sleep 30</code>	指定の秒数の間、遅延を生じさせます。範囲は0～2147483647です。



## 状況依存ヘルプ

Cisco NX-OS ソフトウェアでは、CLI で状況依存ヘルプが提供されます。コマンドのいずれかの位置で疑問符 (?) を使用すると、有効な入力オプションのリストを表示できます。

CLI では、キャレット (^) 記号を使用して入力エラーを隔離します。^ 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。

表 3-5 に、クロックを設定する際のエラー隔離と状況依存ヘルプの使用方法を示します。

表 3-5 状況依存ヘルプの例

	コマンド	目的
ステップ 1	<pre>clock ? switch# clock ?       set HH:MM:SS Current Time switch# clock</pre>	<p>EXEC モードで <b>clock</b> コマンドのコマンド構文を表示します。</p> <p>スイッチの出力に、<b>clock</b> コマンドを使用するには <b>set</b> キーワードが必要であることが示されています。</p>
ステップ 2	<pre>clock set ? switch# clock set ?       WORD HH:MM:SS Current Time switch# clock set</pre>	<p>時間を設定するコマンド構文を表示します。</p> <p>ヘルプ出力に、クロックを設定するには現在の時刻が必要であること、および時間のフォーマットが示されています。</p>
ステップ 3	<pre>clock set HH:MM:SS switch# clock set 13:32:00&lt;CR&gt; % Incomplete command switch#</pre>	<p>現在の時刻を追加します。</p> <p>スイッチには、コマンドが不完全であることが示されています。</p>
ステップ 4	<pre>Ctrl-P or the Up Arrow switch# &lt;Ctrl-P&gt; 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>
ステップ 5	<pre>clock set HH:MM:SS ? switch# clock set 13:32:00 ?       &lt;1-31&gt;      Day of the month switch# clock set 13:32:00</pre>	<p><b>clock set</b> コマンドの追加の引数を表示します。</p>

表 3-5 状況依存ヘルプの例 (続き)

	コマンド	目的
ステップ 6	<pre>clock set HH:MM:SS day ?  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><b>clock set</b> コマンドの追加の引数を表示します。</p>
ステップ 7	<pre>clock set HH:MM:SS day month year  switch# clock set 13:32:00 18 April 08                                      ^ % Invalid input detected at '^' marker.</pre>	<p>日付をクロック設定に追加します。</p> <p>CLIでは、08の位置にエラーがあることがキャレット記号によって示されています。</p>
ステップ 8	<pre>clock set HH:MM:SS day month year ?  switch# clock set 13:32:00 18 April ? &lt;2000-2030&gt; Enter the year (no abbreviation) switch# clock set 13:32:00 18 April</pre>	<p>年の正しい引数を表示します。</p>
ステップ 9	<pre>clock set HH:MM:SS day month year  switch# clock set 13:32:00 18 April 2008</pre>	<p><b>clock set</b> 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、**show** コマンドなどのコマンドで、検索およびフィルタリングを行うために用いる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、複雑な一致要件を指定することが可能です。

ここでは、次の内容について説明します。

- 特殊文字 (p.3-19)
- 単一文字パターン (p.3-19)
- 複数文字パターン (p.3-20)
- 位置指定 (p.3-20)

## 特殊文字

単一文字パターンとしてその他のキーボード文字 ([!] および [-] など) も使用できますが、ある特定のキーボード文字は正規表現で用いられた場合に特別な意味を持ちます。表 3-6 に、特別な意味を持つキーボード文字の一覧を示します。

表 3-6 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に一致します。
+	1 個以上のパターンのシーケンスに一致します。
?	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。

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。次の例は、それぞれドル記号 (\$)、アンダースコア (\_)、プラス記号 (+) に一致する単一文字パターンを示しています。

```
\$ \_ \+
```

## 単一文字パターン

最も単純な正規表現は、コマンド出力内の同じ 1 つの文字と一致する単一文字です。単一文字パターンとして、任意の文字 (A ~ Z, a ~ z) または数字 (0 ~ 9) を使用することができます。単一文字パターンを範囲指定して、コマンド出力とのマッチングを行うことができます。

単一文字パターンを範囲指定するには、単一文字パターンを角カッコ ([ ]) で囲みます。たとえば、次の文字のうち 1 つを含むストリングに一致する正規表現を作成できます。a、e、i、o、または u。パターンマッチングが成功するためには、これらの文字のうち 1 つだけがストリングに存在しなくてはなりません。この場合、[aeiou] は小文字アルファベットの 5 つの母音のうち任意の 1 文字と一致しますが、[abcdABCD] は小文字または大文字アルファベットの最初の 4 つの文字のうち任意の 1 文字と一致します。ダッシュ (-) で区切って範囲の終点だけを入力することにより範囲を簡略化することができます。また、範囲の開始位置にキャレット (^) を含めると、範囲の文字を含まないストリングに一致します。

表 3-7 に、文字の範囲を使用した正規表現の例を示します。

表 3-7 文字の範囲を使用した正規表現の例

例	説明
[a-dA-D]	文字 abcdABCD に一致します。
[a-dA-D\ -]	文字 abcdABCD およびハイフン (-) に一致します。
[a-dA-D\ -\ ]]	文字 abcdABDC、ハイフン (-)、および右角カッコ (]) と一致します。
[^a-dqsv]	a-dqsv 以外の文字と一致します。
[^\ ]d]	右角カッコ (]) または文字 d 以外のすべてと一致します。

## 複数文字パターン

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組み合わせることで、複数の文字を含む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。正規表現 `a4%` は、`a` という文字のあとに `4` が続き、そのあとにパーセント記号 (%) が続く文字と一致します。この複数文字の正規表現 `a` (文字 `a` のあとにピリオドが続く) では、ピリオド文字の特別な意味を使用しており、`a` という文字のあとに任意の文字が 1 つ来る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。

特殊文字の前にバックスラッシュを挿入することにより、特殊文字から特別な意味を除外できます。たとえば、コマンド構文に `a\.` という表現が使われている場合は、`a.` というストリングのみが一致します。

## 位置指定

ストリングの最初または最後に対して正規表現パターンのマッチングを行うことができます。それには、表 3-8 に示す特殊文字を用いたストリングの一部に対して、これらの正規表現を「位置指定」します。

表 3-8 位置指定に用いられる特殊文字

文字	説明
^	ストリングの最初と一致します。
\$	ストリングの最後と一致します。

たとえば、正規表現 `^con` は `con` で始まるストリングに一致し、`$sole` は `sole` で終わるストリングに一致します。



(注)

また、角カッコで囲まれた範囲に使用することで、`^` 記号を論理関数「not」の意味で使用できます。たとえば、正規表現 `[^abcd]` は、`a`、`b`、`c`、または `d` 以外の任意の単一文字に一致する範囲を示します。

## show コマンド出力の検索およびフィルタリング

**show** コマンド出力が長すぎて扱いにくいことがあります。Cisco NX-OS ソフトウェアでは、情報を見つけやすくするために、出力を検索およびフィルタリングする手段が提供されます。**show** コマンドの最後に、パイプ文字 (|) を入力してから検索オプションおよびフィルタリング オプションを続けます。CLI 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 | ?
  egrep      Egrep
  grep       Grep
  head       Stream Editor
  last       Display last lines
  less       Stream Editor
  no-more    Turn-off pagination for command output
  sed        Stream Editor
  wc         Count words, lines, characters
  begin      Begin with the line that matches
  count      Count number of lines
  exclude    Exclude lines that match
  include    Include lines that match
```

## キーワードのフィルタリングおよび検索

Cisco NX-OS CLI では、**show** コマンドと併用してコマンド出力を検索およびフィルタリングできる一連のキーワードが提供されます (表 3-9 を参照)。

表 3-9 キーワードのフィルタリングおよび検索

キーワードの構文	説明	例
<b>begin</b> <i>string</i> <sup>1</sup>	検索文字列に一致するテキストが含まれる行から表示を開始します。	<b>show version   begin Hardware</b>
<b>count</b>	コマンド出力の行数を表示します。	<b>show running-config   count</b>
<b>exclude</b> <i>string</i> <sup>1</sup>	検索文字列を含まないすべての行を表示します。	<b>show interface brief   exclude down</b>
<b>head</b> [-n <i>lines</i> ]	出力の先頭から指定した行数を表示します。デフォルトの行数は 10 です。	<b>show logging logfile   head -n 50</b>
<b>include</b> <i>string</i> <sup>1</sup>	検索文字列を含むすべての行を表示します。	<b>show interface brief   include up</b>
<b>no-more</b>	画面の最後の -- More -- プロンプトで出力が停止しないようにすべての出力を表示します。	<b>show interface brief   no-more</b>

1. 検索文字列では、大文字と小文字が区別されます。

## 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、次のように **show** コマンド出力をフィルタリングできます。

```
{grep | egrep} [-A lines] [-B lines] [-c] [-i] [-n] [-v] [-w] [-x] expression
```

表 3-10 に、grep および egrep のパラメータの説明を示します。

表 3-10 grep および egrep のパラメータ

パラメータ	説明
-A lines	一致した行から何行先まで表示するのかを指定します。デフォルトは 0 です。範囲は 1 ~ 999 です。
-B lines	一致した行の何行前まで表示するのかを指定します。デフォルトは 0 です。範囲は 1 ~ 999 です。
-c	一致した行の合計数のみを表示します。
-i	一致した行で大文字と小文字の区別を無視するように指定します。
-n	一致した各行の前に行番号を表示するように指定します。
-v	表現に一致しない行を表示します。
-w	完全なワードに一致する行のみを表示します。
-x	完全な行に一致する行のみを表示します。
expression	出力の検索用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** コマンド出力の内容を同時に 1 つの画面に表示できます。less コマンドは、:プロンプトで入力できます。使用できるすべての less コマンドを表示するには、:プロンプトで **h** と入力します。

## sed ユーティリティ

Stream Editor (sed) ユーティリティを使用して、**show** コマンド出力を次のようにフィルタリングして操作できます。

```
sed command
```

command 引数には、sed ユーティリティ コマンドが含まれます。

## --More-- プロンプトからの検索およびフィルタリング

--More-- プロンプトから出力を検索およびフィルタリングできます。--More-- プロンプトから **show** コマンド出力を検索およびフィルタリングするには、表 3-11 で説明するコマンドを使用します。

表 3-11 --More-- プロンプト コマンド

コマンド	説明
[lines]<space>	指定した行数または現在の画面サイズ分の出力行を表示します。
[lines]z	指定した行数または現在の画面サイズ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画面サイズになります。
[lines]<return>	指定した行数または現在のデフォルト行数の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に対応する新しいデフォルトの表示行数になります。
[lines]d または [lines]Ctrl+shift+D	指定した行数または現在のデフォルト行数だけ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に対応する新しいデフォルトの表示行数になります。
q または Q または Ctrl-C	--More-- プロンプトを終了します。
[lines]s	指定した行数または現在のデフォルト行数の出力にスキップして、行画面を表示します。デフォルトは 1 行です。
[lines]f	指定した画面数または現在のデフォルト画面数の出力にスキップして、行画面を表示します。デフォルトは 1 画面です。
=	現在の行数を表示します。
[count]/expression	正規表現に一致する行にスキップして、出力行の画面を表示します。正規表現の複数のオカレンスを持つ行を検索するには、オプションの count 引数を使用します。このコマンドでは、他のコマンドで使用可能な現在の正規表現を設定します。
[count]n	現在の正規表現に一致する次の行にスキップして、出力行の画面を表示します。以前の一致をスキップするには、オプションの count 引数を使用します。
{! : } shell-cmd	サブシェルでの shell-cmd 引数で指定したコマンドを実行します。
.	前のコマンドを繰り返します。

## BIOS loader> プロンプト

スーパーバイザ モジュールに電源が入ると、特殊な BIOS イメージが自動的にロードされて、システムを起動するための有効なキックスタート イメージを見つけます。有効なキックスタート イメージが見つからない場合には、次の BIOS loader 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の詳細については、『Cisco NX-OS Troubleshooting Guide』 Release 4.0 を参照してください。

## CLI の使用例

ここでは、次の内容について説明します。

- [CLI エイリアスの定義 \(p.3-24\)](#)
- [CLI セッション変数の使用方法 \(p.3-24\)](#)
- [システム定義の Timestamp 変数の使用方法 \(p.3-25\)](#)
- [コマンドスクリプトの実行 \(p.3-26\)](#)

## CLI 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

構文 `$(variable-name)` 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.4da
c)
  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 Timestamp 変数の使用方法

次に、**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) を使用する例を示します。

```
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

次に、スクリプトファイル内の指定した CLI コマンドを表示する例を示します。

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

次に、**run-script** コマンド実行の出力を表示する例を示します。

```
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
```

## 詳細な資料

図 1 の実装に関連する詳細情報は、次のとおりです。

- [関連資料 \(p.3-27\)](#)

## 関連資料

関連トピック	マニュアル タイトル
ライセンス	『Cisco NX-OS Licensing Guide』 Release 4.0
コマンドリファレンス	『Cisco NX-OS Fundamentals Command Reference』 Release 4.0

