



CHAPTER 5

コマンドライン インターフェイス (CLI) の ヒント、手法、およびショートカット

この章では、Cisco IOS XR ソフトウェアの Command-Line Interface (CLI; コマンドライン インターフェイス) の手法について説明します。

目次

- 「CLI のヒントおよびショートカット」 (P.5-1)
- 「show コマンドを使用したシステム情報の表示」 (P.5-6)
- 「ワイルドカード、テンプレート、およびエイリアス」 (P.5-13)
- 「コマンド履歴」 (P.5-18)
- 「キーの組み合わせ」 (P.5-19)



(注)

コマンドは、大文字、小文字、または大文字と小文字を混在させて入力できます。大文字と小文字の区別があるのはパスワードだけです。ただし、シスコシステムのマニュアル表記法に従い、ここではコマンドを小文字で表記しています。

CLI のヒントおよびショートカット

ここでは、CLI を使用する際に役立つヒントおよびショートカットについて説明します。

- 「省略されたコマンドの入力」 (P.5-2)
- 「オンスクリーン コマンド ヘルプを表示するための疑問符 (?) の使用」 (P.5-2)
- 「不完全なコマンド名を完全にするための Tab キーの使用」 (P.5-4)
- 「コマンド構文エラーの識別」 (P.5-5)
- 「no 形式のコマンドの使用」 (P.5-5)
- 「画面幅よりも長いコマンドラインの編集」 (P.5-6)

省略されたコマンドの入力

コマンドおよびキーワードは、一意の省略形として認識可能な文字数まで省略できます。たとえば、**configure** コマンドは **config** と省略できます。それは、このコマンドの省略形が一意であるためです。ルータは、省略されたコマンドを受け入れて実行します。

オンスクリーン コマンド ヘルプを表示するための疑問符 (?) の使用

使用可能なコマンドおよびコマンドの正しい構文を調べるには、疑問符 (?) を使用します。表 5-1 に、オンスクリーン ヘルプのオプションをまとめます。



ヒント

疑問符 (?) の前にスペースを含めるかどうかは、重要な意味を持ちます。疑問符の前にスペースを含めると、コマンドまたは CLI モードに対して使用できるすべてのオプションが表示されます。スペースを含めない場合は、特定の文字ストリングで始まるコマンドのリストが表示されます。

表 5-1 オンスクリーン ヘルプ コマンド

コマンド	説明
不完全なコマンド名 ?	不完全なコマンド名の末尾に疑問符 (?) を入力すると、入力されている文字で始まるコマンドのリストが表示されます。 RP/0/RSP0/CPU0:router# co? configure copy (注) コマンドと疑問符の間には、スペースを含めないでください。
? コマンド?	特定のコマンド モードで使用できるすべてのコマンドが表示されます。 疑問符 (?) の前にスペースを含めると、そのコマンドに属するキーワードおよび引数のリストが表示されます。 RP/0/RSP0/CPU0:router# configure ? exclusive Configure exclusively from this terminal terminal Configure from the terminal <cr> (注) 多くのコマンドの場合、<cr> 記号は、すでに入力している構文を使用してコマンドを実行できることを示します。前の例では、 Enter キーを押すと、グローバル コンフィギュレーション モードが開始されます。
コマンドキーワード ?	キーワードの末尾に (?) を入力すると、そのコマンドに対して次に使用できる構文オプションのリストが表示されます。 RP/0/RSP0/CPU0:router# show aaa ? taskgroup Show all the local taskgroups configured in the system userdb Show all local users with the usergroups each belong to usergroup Show all the local usergroups configured in the system (注) キーワードと疑問符の間には、スペースを含めてください。

次に、エントリをアクセス リスト 99 に追加する例を示します。追加するエントリにより、サブネット 172.0.0.0 上のすべてのホストへのアクセスが拒否され、0 ~ 255 の範囲内で始まる IPv4 アドレスのビットが無視されます。次の手順は、オンスクリーン コマンド ヘルプの例を示しています。

ステップ 1 **access-list** コマンドを入力し、続けてスペースと疑問符を入力すると、そのコマンドに使用できるオプションのリストが表示されます。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list ?

log-update   Control access lists log updates
maximum      Out of resources configuration
WORD         Access list name - maximum 32 characters
```



(注) 数字の範囲 (かぎカッコ内) は包含範囲です。

ステップ 2 アクセス リスト名 **list1** に続けてスペースと疑問符を入力すると、キーワードおよび簡単な説明に適用される引数が表示されます。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list list1 ?

<1-2147483646> Sequence number for this entry
deny           Specifies packets to reject
permit        Specifies packets to forward
remark        Comment for access list
<cr>
```

```
RP/0/RSP0/CPU0:router(config)#ipv4 access-list list1
```

ステップ 3 **deny** オプションと疑問符を入力すると、さらにコマンド オプションが表示されます。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list list1 deny ?

<0-255> An IPv4 Protocol Number
A.B.C.D Source IP address or prefix
ahp     Authentication Header Protocol
any     Any source host
eigrp   Cisco's EIGRP Routing Protocol
esp     Encapsulation Security Payload
gre     Cisco's GRE Tunneling
host    A single source host
icmp    Internet Control Message Protocol
igmp    Internet Gateway Message Protocol
igrp    Cisco's IGRP Routing Protocol
ipinip  IP in IP tunneling
ipv4    Any IPv4 Protocol
nos     KA9Q NOS Compatible IP over IP Tunneling
ospf    OSPF Routing Protocol
pcp     Payload Compression Protocol
pim     Protocol Independent Multicast
sctp    Stream Control Transmission Protocol
tcp     Transport Control Protocol
udp     User Datagram Protocol

RP/0/RSP0/CPU0:router(config)#ipv4 access-list list1 deny
```

ステップ 4 IP アドレスを入力し、続けてスペースと疑問符 (?) を入力すると、追加のオプションのリストが表示されます。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list list1 deny 172.31.134.0 ?

A.B.C.D      Wildcard bits
log          Log matches against this entry
log-input    Log matches against this entry, including input interface
<cr>
```

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list list1 deny 172.31.134.0
```

<cr> 記号は単独で、それ以上キーワードまたは引数がないことを示します。

ステップ 5 **Enter** キーを押してコマンドを実行します。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list list1 deny 172.31.134.0
```



(注)

設定は、**commit** コマンドを入力してターゲット コンフィギュレーションを実行コンフィギュレーションに追加するまで、アクティブになりません。

不完全なコマンド名を完全にするための Tab キーの使用

完全なコマンド名を思い出せない場合や、入力する必要がある文字数を減らしたい場合は、コマンドの先頭の数字文字を入力して、**Tab** キーを押します。入力した文字ストリングで始まるコマンドが 1 つしかない場合は、自動的に完全なコマンド名が入力されます。入力した文字で始まるコマンドが複数存在する場合は、文字ストリングが一意でないことがビープ音によって示され、入力済みの文字と一致するコマンドのリストが表示されます。

次の例では、**conf** は、EXEC モードで一意のストリングとして認識されるため、**Tab** キーを押すと完全なコマンド名が入力されます。

```
RP/0/RSP0/CPU0:router# conf <Tab>

RP/0/RSP0/CPU0:router# configure
```

CLI には完全なコマンド名が表示されますが、このコマンドを実行するには、**Enter** キーを押す必要があります。このため、ユーザは、提示されたコマンド名を変更したり拒否したりできます。

次の例では、入力した文字と一致するコマンドが 2 つ認識されます。

```
RP/0/RSP1/CPU0:router# co<Tab>
configure copy
RP/0/RSP1/CPU0:router# con<Tab>

RP/0/RSP1/CPU0:router# configure
```



ヒント

キーボード上に **Tab** キーがない場合は、代わりに **Ctrl** キーを押した状態で **I** キーを押します。

コマンド構文エラーの識別

正しくないコマンドを入力すると、エラーメッセージが返され、エラー ポイントにキャレット (^) が表示されます。次の例では、コマンド内の正しくない文字が入力されている位置にキャレットが表示されます。

```
RP/0/RSP0/CPU0:router# configure termiMal
                                     ^
% Invalid input detected at '^' marker.
```



(注)

パーセント記号 (%) は、その行にエラー メッセージが表示されていることを示します。

正しいコマンド構文を表示するには、コマンドの後に「?」を入力します。

```
RP/0/RSP0/CPU0:router# configure ?

exclusive  Configure exclusively from this terminal
terminal   Configure from the terminal
<cr>
```

no 形式のコマンドの使用

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 **no** 形式があります。コマンドによっては、**no** 形式を使用して、機能をイネーブルまたはディセーブルにする場合があります。たとえば、インターフェイスを設定する場合には、**no shutdown** コマンドでインターフェイスを始動し、**shutdown** コマンドでインターフェイスをシャットダウンします。また、有効なユーザ名を入力している場合には、**username** コマンドで新しいユーザを作成し、**no username** コマンドでユーザを削除します。

Cisco IOS XR ソフトウェア のコマンドリファレンスには、コンフィギュレーション コマンドの詳細な構文と、**no** 形式のコマンドの機能に関する説明が記載されています。詳細については、「[表記法](#)」(P.xii) を参照してください。

画面幅よりも長いコマンドラインの編集

画面上で 1 行を超えるコマンドには、ラップアラウンド機能を使用できます。カーソルが右マージンに達すると、そのコマンドラインは 10 文字だけ左へシフトされます。行の先頭から 10 文字までは見えなくなりますが、左へスクロールしてコマンドの先頭部分の構文をチェックできます。コマンドラインを左へスクロールするには、Ctrl キーを押した状態で B キーを押すか、左矢印キーを繰り返し押しします。コマンドラインの先頭に直接移動するには、Ctrl キーを押した状態で A キーを押します。

次の例では、**ipv4 access-list** コマンド エントリは長すぎるため、1 行に表示できません。カーソルが行末に達すると、その行は左へシフトされ、再表示されます。コマンドプロンプトの後のドル記号 (\$) は、その行が左へスクロールされ、コマンドの先頭が非表示になっていることを示します。

```
RP/0/RSP0/CPU0:router(config)# $s-list 101 permit tcp 172.31.134.5 255.255.255.0
172.31.135.0
```

次は、Ctrl キーを押した状態で A キーを押してコマンドラインの先頭を表示したときの例です。コマンドラインの末尾のドル記号 (\$) は、そのコマンドが右へスクロールされ、コマンドの末尾が非表示になっていることを示します。

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list 101 permit tcp 172.31.134.5 255.255.255.0
17$
```

次は、右矢印キーを使用して右へスクロールしたときの例です。行の両端にドル記号が表示されていることに注目してください。これは、このコマンドの先頭および末尾にあるコマンド情報が非表示になっていることを示します。

```
RP/0/RSP0/CPU0:router(config)# $ccess-list 101 permit tcp 172.31.134.5 255.255.255.0
172.31.$
```

デフォルトでは、Cisco IOS XR ソフトウェアは幅が 80 カラムの端末画面を使用します。異なる画面幅に調整するには、EXEC モードで **terminal width** コマンドを使用します。

ラインラップ機能とコマンド履歴機能を併用すると、以前入力した複雑なコマンド エントリを呼び出して修正できます。

show コマンドを使用したシステム情報の表示

show コマンドは、システムおよびその設定に関する情報を表示するために使用します。ここでは、一般的な **show** コマンドを示し、それらのコマンドの出力を管理するための手法について説明します。

- 「一般的な **show** コマンド」 (P.5-7)
- 「**--More--** プロンプトが表示された場合の表示出力の閲覧」 (P.5-7)
- 「画面出力表示の停止」 (P.5-8)
- 「ファイルへの出力のリダイレクト」 (P.5-8)
- 「大規模な設定出力の制限」 (P.5-9)
- 「**show** コマンド出力のフィルタリング」 (P.5-10)

一般的な show コマンド

表 5-2 に、最も一般的な show コマンドを示します。

表 5-2 Cisco IOS XR ソフトウェアに用意されている一般的な show コマンド

コマンド	説明	コマンド モード
show version	システム情報を表示します。	EXEC モードまたは管理 EXEC モード
show configuration	コンフィギュレーション セッション中に加えられた未保存の設定変更を表示します。	グローバルまたは管理コンフィギュレーション モード
show running-config (EXEC またはグローバル コンフィギュレーション モード)	接続先の RP の現在の実行コンフィギュレーションを表示します。	EXEC またはグローバル コンフィギュレーション モード
show running-config (管理 EXEC または管理コンフィギュレーション モード)	ルータ全体に適用される現在の実行コンフィギュレーションを表示します。	管理 EXEC または管理コンフィギュレーション モード
show tech-support	トラブルシューティングのための大量のシステム情報を収集します。この出力は、テクニカルサポート担当者に問題を報告する際に提供できます。	EXEC モードまたは管理 EXEC モード
show platform (EXEC モード)	接続先の RP に割り当てられているカードおよびモジュールに関する情報を表示します。	EXEC モード
show platform (管理 EXEC モード)	ルータのすべてのカードおよびモジュールに関する情報を表示します。	管理 EXEC モード
show environment	ファン、LED、電源電圧および電流、温度など、システムのハードウェア情報を表示します。追加のコマンドオプションを表示するには、 show environment ? と入力します。	EXEC モードまたは管理 EXEC モード

これらのコマンドの使用の詳細については、「表記法」(P.xii) を参照してください。

--More-- プロンプトが表示された場合の表示出力の閲覧

?. show、more など、コマンド出力が複数画面に及ぶコマンドの場合、出力は 1 画面ずつ表示され、画面の下部に --More-- プロンプトが表示されます。

追加のコマンド出力を表示するには、次のいずれかを行います。

- Enter キーを押して次の行を表示する。
- スペース バーを押して出力の次の画面を表示する。

次の例は、1 画面分のデータと --More-- プロンプトを示しています。

```
RP/0/RSP0/CPU0:router# show ?

aaa                Show AAA configuration and operational data
adjacency          Adjacency information
aliases            Display alias commands
alphadisplay       Shows the message being displayed on the alpha display
aps                SONENT APS information
arm                IP ARM information
arp                ARP table
as-path-access-list List AS path access lists
asic-errors        ASIC error information
atc                Attractor Cache related
auto-rp            Auto-RP Commands
buffer-manager     Show all buffer manager memory related information
bundle             Show hardware related information for Bundles.
calendar           Display the system calendar
cdp                CDP information
cef                Cisco Express Forwarding
cetftp             HFR control plane ethernet TFTP server
checkpoint         Show checkpoint services
cinetd             cinetd daemon
clns                Display CLNS related information
clock              Display the system clock
commit             Show commit information
--More--
```



ヒント

--More-- プロンプトが表示されない場合は、EXEC モードで **terminal length** コマンドを使用して、画面長の値を入力してみてください。**length** の値を 0 に設定している場合、コマンド出力は一時停止されません。次の例は、**terminal length** の設定方法を示しています。

```
RP/0/RSP1/CPU0:router# terminal length 20
```

CLI の出力の検索やフィルタリングについては、「[show コマンド出力のフィルタリング](#)」(P.5-10) を参照してください。

画面出力表示の停止

画面出力を中断して表示を終了するには、次の例のように、Ctrl キーを押した状態で C キーを押します。

```
RP/0/RSP0/CPU0:router# show running-config
<Ctrl-C>
```

ファイルへの出力のリダイレクト

デフォルトでは、CLI のコマンド出力は画面上に表示されます。**show** コマンド構文の後にファイル名および場所を入力することにより、ユーザが指定したファイルに CLI のコマンド出力をリダイレクトできます。出力をファイルにリダイレクトするには、次のコマンド構文を使用します。

```
show command | file filename
```


この機能を使用すると、将来の分析および参照用に、任意の **show** コマンドの出力をファイルに保存できます。コマンド出力をリダイレクトする場合は、次の注意事項に従ってください。

- ファイルのフルパスを指定しない場合は、アカウントのデフォルト ディレクトリが使用されます。ターゲット コンフィギュレーション ファイルは、常にこの場所に保存する必要があります。
- 保存した出力をコンフィギュレーション ファイルとして使用する場合は、容易に識別できるように、ファイル名の最後に拡張子 `cfg` を付けます。この拡張子は必須ではありませんが、ターゲット コンフィギュレーション ファイルを特定する際に役立つ場合があります。

例：myconfig.cfg

次の例では、ターゲット コンフィギュレーション ファイルはデフォルト ユーザ ディレクトリに保存されます。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# show configure | file disk0:myconfig.cfg
RP/0/RSP0/CPU0:router(config)# abort
RP/0/RSP0/CPU0:router#
```

大規模な設定出力の制限

大規模な実行コンフィギュレーションを表示すると、何千行もの出力データが生成されます。確認する必要がある項目だけが表示されるように **show** コマンドの出力を制限するには、次のセクションの手順に従います。

- 「特定の機能またはインターフェイスへの **show** コマンド出力の制限」(P.5-9)
- 「インターフェイスのすべてのインスタンスを表示するためのワイルドカードの使用」(P.5-9)

特定の機能またはインターフェイスへの **show** コマンド出力の制限

show コマンドにキーワードや引数を入力すると、**show** コマンドの出力は、特定のキーワードまたはインターフェイスに制限されます。

次の例では、スタティック IP ルート設定に関する情報だけが表示されます。

```
RP/0/RSP1/CPU0:router# show running-config router static
```

```
router static
address-family ipv4 unicast
  0.0.0.0/0 10.21.0.1
  0.0.0.0/0 Gi0/1/0/1 10.21.0.1
!
```

次の例では、特定のインターフェイスの設定が表示されます。

```
RP/0/RSP0/CPU0:router# show running-config interface Gi 0/1/0/1
```

```
interface Gi0/1/0/1
  ipv4 address 10.21.54.31 255.255.0.0
!
```

インターフェイスのすべてのインスタンスを表示するためのワイルドカードの使用

すべてのインスタンスの設定を表示するには、アスタリスク (*) ワイルドカード文字を入力します。



(注) 詳細については、「[show コマンドでインターフェイスを特定するためのワイルドカードの使用](#)」(P.5-13) を参照してください。

次の例では、すべてのギガビット イーサネット インターフェイスの設定が表示されます。

```
RP/0/RSP1/CPU0:router# show running-config interface gi *

interface Gi0/1/0/0
  ipv4 address 10.2.3.4 255.255.255.0
  gi
    crc 32
  !
  shutdown
  keepalive disable
  !
interface Gi0/1/0/1
  ipv4 address 10.2.3.5 255.255.255.0
  gi
    crc 32
  !
  shutdown
  keepalive disable
  !
interface Gi0/1/0/2
  ipv4 address 10.2.3.6 255.255.255.0
  gi
    crc 32
  !
  shutdown
  keepalive disable
  !
interface Gi0/1/0/3
  ipv4 address 10.2.3.7 255.255.255.0
  gi
    crc 32
  !
  shutdown
  keepalive disable
  !

--More--
```

show コマンド出力のフィルタリング

show コマンドの出力では、大量のデータが生成される可能性があります。情報の一部だけを表示するには、パイプ文字 (|) を入力し、続けてキーワード (**begin**、**include**、**exclude**、または **file**) と正規表現を入力します。表 5-3 に、**show** コマンドのフィルタリング オプションを示します。

表 5-3 show コマンドのフィルタリング オプション

コマンド	説明
<code>show command begin regular-expression</code>	フィルタリングを使用せずに、正規表現を含む最初の行から show コマンドの出力を開始します。
<code>show command exclude regular-expression</code>	正規表現を含まない出力行を表示します。

表 5-3 show コマンドのフィルタリング オプション

コマンド	説明
<code>show command include regular-expression</code>	正規表現を含む出力行を表示します。
<code>show command file device0:path/file</code>	正規表現を含む出力行を、指定されたデバイスの指定されたファイルに書き込みます。

次の例では、**show interface** コマンドは、「protocol」という表現を含む行だけを表示しています。

```
RP/0/RSP0/CPU0:router# show interface | include protocol
```

```
Null0 is up, line protocol is up
0 drops for unrecognized upper-level protocol
Gi0/2/0/0 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
Gi0/2/0/1 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
Gi0/2/0/2 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
Gi0/2/0/3 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
MgmtEthernet0/RSP0/CPU0/0 is administratively down, line protocol is administratively
down
MgmtEthernet0/RSP0/CPU0/0 is administratively down, line protocol is administratively
down
0 drops for unrecognized upper-level protocol
```



(注)

フィルタリングが使用できるのは、サブモード、完全なコマンド、および「?」を指定したときに出力に <cr> が表示される場合だけです。

--More-- プロンプトでのフィルタの追加

show コマンドの出力の **--More--** プロンプトで、フィルタを指定できます。そのためには、スラッシュ (/) を入力し、続けて正規表現を入力します。フィルタは、コマンドの出力が完了するか、中断される (Ctrl キーを押した状態で Z キーを押すか、Ctrl キーを押した状態で C キーを押す) までアクティブになります。この手法には、次の規則が適用されます。

- 元のコマンドまたは直前の **--More--** プロンプトでフィルタが指定されている場合は、2 番目のフィルタを適用できない。
- キーワード **begin** を使用してもフィルタが構成されない。
- 正規表現の前にマイナス (-) 記号を配置すると、正規表現が含まれない出力行が表示される。
- 正規表現の前にプラス (+) 記号を配置すると、正規表現が含まれる出力行が表示される。

次の例では、**--More--** プロンプトでフィルタを追加することにより、残りの出力に正規表現「ip」を含む行だけを表示しています。

```
RP/0/RSP0/CPU0:router# show configuration running | begin line
```

```
Building configuration...
line console
  exec-timeout 120 120
!
logging trap
--More--
/ip
filtering...
```

■ show コマンドを使用したシステム情報の表示

```
ip route 0.0.0.0 255.255.0.0 Gi0/2/0/0
interface Gi0/2/0/0
  ip address 172.19.73.215 255.255.0.0
end
```



ヒント

ほとんどのシステムでは、いつでも **Ctrl** キーを押した状態で **C** キーを押して出力を中断し、EXEC モードに戻ることができます。

詳細については、[付録 A「正規表現、特殊文字、およびパターンの概要」](#) を参照してください。

マルチパイプのサポート

マルチパイプ機能では、コマンドライン インターフェイス上で複数のパイプがサポートされます。この機能を使用すると、拡張ユーティリティ セットで出力を処理できます。さまざまなユーティリティの組み合わせを使用して、**show** コマンドの出力を収集、フィルタリング、および書式設定できます。コマンドライン インターフェイス上でサポートされるパイプ数は無原則に 8 パイプに制限されていますが、パイプを使用して指定した個々のコマンドの長さによっては、1 行に入力できる文字数の制限 (1,024) がこの制限に優先します。

また、パイプ文字 (|) をパターンとして指定する場合は、二重引用符で囲む必要があります。次に例を示します。

```
RP/0/RP1/CPU0:single8-hfr# show running-config|include "gi|ospf"|file disk0:/usr/a.log
```

show parser dump 拡張機能

show parser dump コマンドを実行すると、特定のサブモードに対する CLI 構文オプションが表示されます。

これは、ルータでサポートされているパーサー コマンドをダンプするためのユーティリティであり、サブモードで使用できる行ごとのコマンドを表示するツールでもあります。このコマンドはすべてのモードで使用でき、そのモードで使用できるコマンドセットを表示します。これは、モードに使用できる CLI コマンドを収集するための非常に便利なツールです。

show parser dump コマンドは、フィルタをサポートしています。コマンドの最初の部分を指定すると、一致するコマンドが表示されます。

```
RP/0/RSP1/CPU0:router(config-un)# show parser dump
```

```
show
show configuration merge
show configuration running sanitized desanitize rpl
show configuration running sanitized
show configuration running
show configuration
show configuration failed noerrors
show configuration failed
show configuration failed load
show running-config
show running-config sanitized desanitize rpl
show running-config sanitized
show running-config submode
show parser dump
show history detail
show history
pwd
exit
```

ワイルドカード、テンプレート、およびエイリアス

この項では、次のトピックについて取り上げます。

- 「`show` コマンドでインターフェイスを特定するためのワイルドカードの使用」 (P.5-13)
- 「コンフィギュレーションテンプレートの作成」 (P.5-14)
- 「エイリアス」 (P.5-17)
- 「コマンドエイリアスとして使用できるキーストローク」 (P.5-18)

show コマンドでインターフェイスを特定するためのワイルドカードの使用

ワイルドカード (*) を使用して、`show` コマンドでインターフェイスのグループを特定します。表 5-4 に、インターフェイスのグループを特定するためのワイルドカードの使用例を示します。

表 5-4 ワイルドカードの使用例

ワイルドカードの構文	説明
*	すべてのインターフェイスを指定します。
gi*	システムのすべてのギガビットイーサネットインターフェイスを指定します。
gi0/1/*	ラック 0、スロット 1 にあるすべてのギガビットイーサネットインターフェイスを指定します。
gi0/3/4.*	Gi0/3/4 のすべてのサブインターフェイスを指定します。



(注)

ワイルドカード (*) は、インターフェイス名の最後の文字である必要があります。

例

次の例では、ラック 0、スロット 1 にあるすべてのギガビットイーサネットインターフェイスの設定が表示されます。

```
RP/0/RSP1/CPU0:router# show running-config interface Gi0/1/*
```

```
interface Gi0/1/0/0
  ipv4 address 10.2.3.4 255.255.255.0
  gi
    crc 32
  !
  keepalive disable
interface Gi0/1/0/1
  ipv4 address 10.2.3.5 255.255.255.0
  gi
    crc 32
  !
  keepalive disable
interface Gi0/1/0/2
  ipv4 address 10.2.3.6 255.255.255.0
```

■ ワイルドカード、テンプレート、およびエイリアス

```

gi
  crc 32
  !
  keepalive disable
interface Gi0/1/0/3
  ipv4 address 10.2.3.7 255.255.255.0
  gi
  crc 32
  !
  keepalive disable

--More--

```

次の例では、すべてのギガビット イーサネット インターフェイスの状態が表示されます。

```
RP/0/RSP1/CPU0:router# show interfaces gi* brief
```

Intf Name	Intf State	LineP State	Encap Type	MTU (byte)	BW (Kbps)
Gi0/1/0/0	up	up	HDLC	4474	2488320
Gi0/1/0/1	up	up	HDLC	4474	2488320
Gi0/1/0/2	up	up	HDLC	4474	2488320
Gi0/1/0/3	up	up	HDLC	4474	2488320
Gi0/1/0/4	up	up	HDLC	4474	2488320
Gi0/1/0/5	up	up	HDLC	4474	2488320
Gi0/1/0/6	up	up	HDLC	4474	2488320
Gi0/1/0/7	up	up	HDLC	4474	2488320
Gi0/1/0/8	up	up	HDLC	4474	2488320
Gi0/1/0/9	up	up	HDLC	4474	2488320
Gi0/1/0/10	up	up	HDLC	4474	2488320
Gi0/1/0/11	up	up	HDLC	4474	2488320
Gi0/1/0/12	up	up	HDLC	4474	2488320
Gi0/1/0/13	up	up	HDLC	4474	2488320
Gi0/1/0/14	up	up	HDLC	4474	2488320
Gi0/1/0/15	up	up	HDLC	4474	2488320

コンフィギュレーション テンプレートの作成

コンフィギュレーション テンプレートにより、コンフィギュレーション コマンドのグループを表す名前を作成できます。テンプレートを定義すると、そのテンプレートをすべてのユーザがインターフェイスに適用できるようになります。ネットワークの規模が大きくなり、ノードやポートの数が多くなると、テンプレートを使用して複数のポートをすばやく設定する機能を使用することで、インターフェイスの設定するための時間を大幅に削減できます。

テンプレートを使用する作業での 2 つの主要なステップは、テンプレートの作成とテンプレートの適用です。ここでは、コンフィギュレーション テンプレートの作成方法について説明します。

手順概要

1. **configure**
2. **template** *template-name* [(*\$parameter \$parameter...*)] [*config-commands*]
3. **template** コマンドを入力します。
4. **end-template**
5. **commit**
6. **show running-config template** *template-name*

詳細手順

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code> 例： Router# configure	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>template template-name [(\$parameter \$parameter...)] [config-commands]</code> 例： RP/0/RSP0/CPU0:router(config)# template tmplt_1	テンプレート コンフィギュレーション モードを開始して、テンプレートを作成します。 <ul style="list-style-type: none"> • <i>template-name</i> : 実行コンフィギュレーションに適用するテンプレートの一意の名前。 • <i>parameter</i> : (省略可能) テンプレートの定義で指定した変数の実際の値。カッコ内に、パラメータを 5 つまで指定できます。各パラメータは、\$ 記号で始まる必要があります。パラメータを使用するかどうかは任意です。 • <i>config-commands</i> : (省略可能) テンプレートの定義に追加するグローバル コンフィギュレーション コマンド。コマンド内のすべての名前 (サーバ名、グループ名など) はパラメータ化できます。つまり、これらのパラメータは、(\$ で始まる) <code>template</code> コマンド内で使用され、適用時に実際の引数に置き換えられます。 • テンプレートを削除するには、このコマンドの no 形式を使用します。
ステップ 3	template コマンドを入力します。 例： RP/0/RSP0/CPU0:router(config-TPL)# hostname test	template コマンドを定義します。
ステップ 4	<code>end-template</code> 例： RP/0/RSP0/CPU0:router(config-TPL)# end-template	テンプレートの定義セッションを終了し、テンプレート コンフィギュレーション モードを終了します。 <ul style="list-style-type: none"> • テンプレートのセッションを終了すると、グローバル コンフィギュレーション モードに戻ります。
ステップ 5	<code>commit</code> 例： RP/0/RSP0/CPU0:router(config-TPL)# commit	ターゲット コンフィギュレーション コマンドを実行コンフィギュレーションに適用します。
ステップ 6	<code>show running-config template template-name</code> 例： RP/0/RSP0/CPU0:router# show running-config template tmplt_1	テンプレートの詳細を表示します。

例

次の例では、簡単なテンプレートを定義しています。テンプレートの内容を表示するには、例のように **show running-config template *template-name*** と入力してください。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# template jbstest
RP/0/RSP0/CPU0:router(config-TPL)# hostname test
RP/0/RSP0/CPU0:router(config-TPL)# end-template
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:router(config)# show running-config template jbstest

template jbstest
  hostname test
end-template
```

次の例では、テンプレートを定義していますが、このテンプレートにはパラメータが必要です。テンプレートの内容を表示するには、例のように **show running-config template *template-name*** と入力してください。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# template test2 (hostname)
RP/0/RSP0/CPU0:router(config-TPL)# hostname $hostname
RP/0/RSP0/CPU0:router(config-TPL)# end-template
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:router(config)# show running-config template test2

template test2 (hostname )
  hostname $hostname
end-template
```

コンフィギュレーション テンプレートの適用

テンプレートを適用するには、グローバル コンフィギュレーション モードで **apply-template *template-name* [(*parameter*)]** コマンドを入力します。その場合、次の注意事項に従ってください。

- 一度に 1 つのテンプレートだけを適用できます。
- 同じテンプレートを 2 回以上適用すると、最新のアプリケーションによって以前のアプリケーションが上書きされます。
- テンプレートには、正確な数のパラメータを指定してください。
- テンプレートは、「ベストエフォート型」処理として適用されます。つまり、有効な変更だけが保存されます。テンプレート内のコマンドのいずれかが失敗すると、そのコマンドは廃棄されます。
- テンプレートを適用したら、**show configuration** コマンドを使用して対象の設定変更を表示します。ターゲット コンフィギュレーションは、実行コンフィギュレーションの一部となるように、(**commit** コマンドを使用して) 保存する必要があります。

例

次の例では、簡単なテンプレートを定義しています。テンプレートの内容を表示するには、例のように **show running-config template *template-name*** と入力してください。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# apply-template jbstest
RP/0/RSP0/CPU0:router(config)# show running-config template jbstest
```



```
Building configuration...
hostname test
end
```

次の例では、パラメータを 1 つ指定したテンプレートを適用し、`show configuration` コマンドを使用して結果を表示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# apply-template test2 (router)
RP/0/RSP0/CPU0:router(config)# show configuration
```

```
Building configuration...
hostname router
end
```

エイリアス

Cisco IOS XR ソフトウェアでは、ルータ内の任意の物理エンティティまたは論理エンティティに対してコマンドラインエイリアスを定義できます。定義したエイリアスは、CLI で実際のエンティティを参照するために使用できます。

コマンドエイリアスを作成するには、グローバル コンフィギュレーション モードまたは管理コンフィギュレーション モードで `alias` コマンドを入力します。

```
alias alias-name [(parameter1 parameter2...)] command-syntax [$parameter1] [command-syntax [$parameter2]]
```

表 5-5 に、`alias` コマンド構文の定義を示します。

表 5-5 `alias` コマンド構文

構文	エイリアスの作成対象
<i>alias-name</i>	コマンドエイリアスの名前。エイリアス名は、1 つの単語、またはダッシュ (-) で結合された複数の単語で構成されます。
<i>command-syntax</i>	元のコマンド構文。 <i>command-syntax</i> の引数には、元のコマンド構文引数の有効な省略形を入力できます。
(<i>parameterx</i>)	<i>command-syntax</i> の引数に指定したコマンドに属する引数またはキーワード。エイリアス名に続けてカッコ内にパラメータを入力している場合は、エイリアスにパラメータが必要です。エイリアス内のコマンドにパラメータを関連付けるには、パラメータ名の前に \$ 記号を入力します。

1 つのコマンドエイリアスで複数のコマンドをサポートできます。また、コマンドごとに複数の変数をサポートできます。1 つのエイリアスで複数のコマンドを指定すると、各コマンドは、その `alias` 内にリストされている順序で実行されます。

次の例では、管理イーサネット インターフェイスに対して `my-cookie` という名前のエイリアスを作成し、その新しいエイリア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 モードを開始します。

```
RP/0/RSP0/CPU0:router(config)# alias my-cookie mgmtEth 0/0/CPU0/0

RP/0/RSP0/CPU0:router(config)# interface my-cookie
RP/0/RSP0/CPU0:router(config)# interface mgmtEth 0/0/CPU0/0
RP/0/RSP0/CPU0:router(config-if)#
```

エイリアスを使用してコマンドを入力すると、入力したコマンドがエイリアスの値と共に表示されるため、そのエイリアスの値を確認できます。

特定のエイリアスを削除するには、**alias** コマンドの **no** 形式をエイリアス名と共に入力します。

コマンド エイリアスとして使用できるキーストローク

特定のキーストローク（キーの組み合わせまたはシーケンス）をコマンド エイリアスとして認識するようにシステムを構成できます。つまり、キーストロークは、コマンドを実行するためのショートカットとして設定できます。システムがキーストロークをコマンドとして解釈できるようにするには、コマンド シーケンスを入力する前に、Ctrl キーを押した状態で V キーを押すか、Esc キーを押してから Q キーを押します。

コマンド履歴

Cisco IOS XR ソフトウェアでは、最近入力または削除したコマンドの履歴を表示できます。コンソール メッセージが表示されているときに、コマンドラインを再表示することもできます。ここでは、コマンド履歴機能について説明します。

- 「以前入力したコマンドの表示」(P.5-18)
- 「以前入力したコマンドの呼び出し」(P.5-18)
- 「削除したエントリの呼び出し」(P.5-19)
- 「コマンドラインの再表示」(P.5-19)



(注)

以前コミットしたコンフィギュレーションにロールバックする方法については、「[コンフィギュレーション履歴の管理とロールバック](#)」(P.4-6) を参照してください。

以前入力したコマンドの表示

Cisco IOS XR ソフトウェアの履歴バッファには、コマンドラインで実行された最新のコマンドが 10 個まで記録されています。この機能は、アクセス リストなど、長いまたは複雑なコマンドやエントリを呼び出す場合、特に便利です。

履歴バッファのコマンドを表示するには、次のように **show history** コマンドを入力します。

```
RP/0/RSP0/CPU0:router# show history

show configuration history commit
show configuration commit list
sho config commit changes 1000000001
show history
```

以前入力したコマンドの呼び出し

Cisco IOS XR ソフトウェアの履歴バッファには、コマンドラインで実行された最新のコマンドが 10 個まで記録されています。この機能は、アクセス リストなど、長いまたは複雑なコマンドやエントリを呼び出す場合、特に便利です。

履歴バッファのコマンドを呼び出すには、表 5-6 に示すコマンドまたはキーの組み合わせのいずれかを使用します。

表 5-6 コマンド履歴

コマンドまたはキーの組み合わせ	目的
Ctrl+P または上矢印キー	直前に入力したコマンドから始めて、履歴バッファに保管されているコマンドを呼び出します。キー シーケンスを繰り返すと、古いコマンドが順に呼び出されます。
Ctrl+N または下矢印キー	Ctrl+P または上矢印キーでコマンドを呼び出してから、履歴バッファ内のより新しいコマンドに戻ります。キー シーケンスを繰り返すと、より新しいコマンドが順に呼び出されます。

削除したエントリの呼び出し

Cisco IOS XR CLI では、削除したコマンドやキーワードも履歴バッファに保存されます。履歴バッファには、Ctrl キーを押した状態で K キーを押す、Ctrl キーを押した状態で U キーを押す、または Ctrl キーを押した状態で X キーを押すことによって削除した最新の 10 項目が保存されます。Back Space キーを押すか、Ctrl キーを押した状態で D キーを押すことによって削除した個々の文字は保存されません。

表 5-7 に、削除したエントリをコマンドラインに呼び出すために使用するキーストロークの組み合わせを示します。

表 5-7 削除したエントリをコマンドラインに呼び出すためのキーストロークの組み合わせ

コマンドまたはキーの組み合わせ	呼び出される内容
Ctrl+Y	履歴バッファ内の最新のエントリ (2 つのキーを同時に押す)
Esc、Y	履歴バッファ内の直前のエントリ (2 つのキーを連続して押す)



(注)

キー シーケンス Esc、Y が機能するためには、まずキーの組み合わせ Ctrl+Y を押す必要があります。Esc、Y を押した回数が 10 回を超えると、バッファ内の最新のエントリに戻って循環します。

コマンドラインの再表示

コマンドの入力中にシステムからメッセージが送信された場合は、キーの組み合わせ Ctrl+L または Ctrl+R を使用して、現在のコマンドライン エントリを再表示できます。

キーの組み合わせ

ここでは、キーの組み合わせについて説明します。

- 「カーソルを移動するためのキーの組み合わせ」 (P.5-20)
- 「キャピタリゼーションを制御するためのキーストローク」 (P.5-20)

- 「CLI のエントリを削除するためのキーストローク」 (P.5-21)
- 「誤って入力した文字の置き換え」 (P.5-21)

カーソルを移動するためのキーの組み合わせ

表 5-8 に、修正または変更を加える際、コマンドラインでカーソルを移動するために使用できるキーの組み合わせまたはキー シーケンスを示します。カーソル制御キーを使用する場合は、次の注意事項に従ってください。

- Ctrl は Ctrl キーを示し、関連する文字キーと同時に押す必要があります。
- Esc は Esc キーを示し、このキーを最初に押してから、続けて関連する文字キーを押す必要があります。
- キーに大文字と小文字の区別はありません。

表 5-8 カーソルを移動するために使用するキーの組み合わせ

キーストローク	機能	カーソルの移動方向
左矢印または Ctrl+B	1 文字後退	1 文字左へ移動します。1 行を超えるコマンドを入力する場合は、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 プロンプトの方へスクロールし、コマンド エントリの先頭に戻るか、キーの組み合わせ Ctrl+A を押します。
右矢印または Ctrl+F	1 文字前進	1 文字右へ移動します。
Esc、B	1 語後退	1 語後退します。
Esc、F	1 語前進	1 語前進します。
Ctrl+A	行頭	コマンドラインの先頭へ移動します。
Ctrl+E	行末	コマンドラインの末尾へ移動します。

キャピタリゼーションを制御するためのキーストローク

単純なキー シーケンスを使用して、大文字と小文字を変更できます。表 5-9 に、キャピタリゼーションを制御するために使用するキーストロークの組み合わせを示します。



(注) Cisco IOS XR コマンドには、一般に大文字と小文字の区別はありませんが、通常はすべて小文字で表記します。

表 5-9 キャピタリゼーションを制御するために使用するキーストローク

キーストローク	目的
Esc、C	カーソル位置の文字を大文字にします。
Esc、L	カーソル位置の単語を小文字にします。
Esc、U	カーソル位置から単語の末尾の文字を大文字にします。

CLI のエントリを削除するためのキーストローク

表 5-10 に、コマンドライン エントリを削除するために使用するキーストロークを示します。

表 5-10 エントリを削除するためのキーストローク

キーストローク	削除対象
Del または Back Space	カーソル位置の左の文字。
Ctrl+D	カーソル位置の文字。
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。
Ctrl+U または Ctrl+X	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。
Ctrl+W	カーソル位置の左の単語。
Esc、D	カーソル位置から単語の末尾まで。

誤って入力した文字の置き換え

誤って入力した文字を置き換えるには、キーの組み合わせ Ctrl+T を使用します。

■ キーの組み合わせ