



## CLI

この章では、Catalyst 4500 シリーズ スイッチでの Cisco IOS CLI（コマンドライン インターフェイス）の概要と使用方法を説明します。この章の内容は次のとおりです。

- ヘルプの利用方法 (p.1-1)
- コマンド オプションの検索 (p.1-2)
- コマンド モードの概要 (p.1-5)
- コマンドの no 形式および default 形式の使用方法 (p.1-6)
- CLI ストリング検索の使用 (p.1-7)
- 設定変更の保存 (p.1-11)

Catalyst 4500 シリーズ スイッチ Cisco IOS 設定の概要については、『*Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide*』を参照してください。

## ヘルプの利用方法

システム プロンプトに疑問符 (?) を入力すると、各コマンド モードで使用できるコマンドのリストが表示されます。また、このコンテキスト ヘルプ機能を使用して、各コマンドのキーワードおよび引数を表示することもできます。

表 1-1 に、コマンド モード、コマンド、キーワード、または引数に対して特定のヘルプ情報を表示できるコマンドの一覧を表示します。

表 1-1 ヘルプの利用方法

| コマンド  | 説明  |
|---|---|
| <i>abbreviated-command-entry?</i>           | 指定した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 (コマンドと疑問符の間にスペースを入れないでください)。 |
| <i>abbreviated-command-entry&lt;Tab&gt;</i> | 指定したコマンド省略形から、完全なコマンド名が表示されます。                              |
| ?   | コマンド モードに対応するすべてのコマンドが表示されます。                               |
| <i>command ?</i>                            | コマンドに対応するすべてのキーワードが表示されます。コマンドと疑問符の間にスペースを入れてください。          |
| <i>command keyword ?</i>                    | キーワードに対応するすべての引数が表示されます。キーワードと疑問符の間にスペースを入れてください。           |

## コマンドオプションの検索

ここでは、コマンド構文を表示する例を示します。コマンド構文には、任意または必須のキーワードが含まれています。コマンドのキーワードを表示するには、コマンドプロンプトで、またはコマンドの一部とスペースを入力したあとで、疑問符(?)を入力します。Catalyst 4500 シリーズスイッチソフトウェアにより、使用できるキーワードのリストと、キーワードの簡単な説明が表示されます。たとえば、グローバルコンフィギュレーションモードで **arap** コマンドの全キーワードを調べたい場合には、**arap ?** と入力します。

表 1-2 では、次のコマンド入力を例にして、コマンド入力の場合の疑問符(?)の使用法を示します。

- **interface gigabitethernet 1/1**
- **channel-group 1 mode auto**

表 1-2 コマンドオプションの検索

| コマンド   | 説明   |
|--|--|
| Switch> <b>enable</b><br>Password: <password><br>Switch#   | <b>enable</b> コマンドおよびパスワードを入力して、特権 EXEC コマンドを開始します。<br><br>特権 EXEC モードが開始されると、プロンプトが Switch# に変わります。   |
| Switch# <b>configure terminal</b><br>Enter configuration commands, one per line. End with CNTL/Z.<br>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。<br><br>グローバル コンフィギュレーションモードが開始されると、プロンプトが Switch(config)# に変わります。  |
| Switch(config)# <b>interface gigabitethernet ?</b><br><1-9> GigabitEthernet interface number<br>Switch(config)# <b>interface gigabitethernet 1/1</b><br>Switch(config-if)# | <b>interface gigabitethernet</b> グローバルコンフィギュレーション コマンドを入力し、設定するギガビットイーサネット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します。<br><br>? を入力し、コマンドラインに次に入力する内容を調べます。この例では、 <i>module-number/port-number</i> のフォーマットで 1~9 のインターフェイス番号を入力する必要があります。<br><br>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ると、プロンプトが Switch(config-if)# に変わります。 |

表 1-2 コマンドオプションの検索 (続き)

| コマンド  | 説明  |
|---|---|
| <pre>Switch(config-if)#? Interface configuration commands:   access-expression  Build a bridge boolean access expression   apollo             Apollo interface subcommands   appletalk         Appletalk interface subcommands   arp               Set arp type (arpa, probe, snap) or timeout   backup            Modify backup parameters   bandwidth         Set bandwidth informational parameter   bgp-policy        Apply policy propogated by bgp community string   bridge-group      Transparent bridging interface parameters   carrier-delay     Specify delay for interface transitions   cdp               CDP interface subcommands   channel-group     Etherchannel/port bundling configuration   clns              CLNS interface subcommands   cmns              OSI CMNS   custom-queue-list Assign a custom queue list to an interface   decnet            Interface DECnet config commands   default           Set a command to its defaults   delay             Specify interface throughput delay   description       Interface specific description   dlsw              DLSw interface subcommands   dspu              Down Stream PU   exit              Exit from interface configuration mode   fair-queue        Enable Fair Queuing on an Interface   flowcontrol       Configure flow operation.   fras              DLC Switch Interface Command   help              Description of the interactive help system   hold-queue        Set hold queue depth   ip                Interface Internet Protocol config commands   ipx               Novell/IPX interface subcommands   isis              IS-IS commands   iso-igrp          ISO-IGRP interface subcommands . . .</pre> | <p>?を入力し、ギガビットイーサネットインターフェイスに対して使用できるすべてのインターフェイス コンフィギュレーション コマンドのリストを表示します。</p>   |
| <pre>Switch(config-if)# Switch(config-if)# channel-group ?   group channel-group of the interface  Switch(config-if)#channel-group</pre>  | <p>コントローラを設定するコマンドを入力します。この例では、<b>channel-group</b> コマンドを入力します。</p> <p>?を入力し、コマンドラインに次に入力する内容を調べます。この例では、<b>group</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt;が表示されていないので、コマンドを完成するためにさらに多くの情報を入力する必要があります。</p> |
| <pre>Switch(config-if)# channel-group ?   &lt;1-256&gt; Channel group number  Switch(config-if)#channel-group</pre>   | <p><b>group</b> キーワードを入力してから、?を入力して、コマンドラインに次に入力する内容を調べます。この例では、チャンネルグループ番号 1 ~ 256 を入力する必要があります。</p> <p>&lt;cr&gt;が表示されていないので、コマンドを完成するためにさらに多くの情報を入力する必要があります。</p>  |

表 1-2 コマンドオプションの検索 (続き)

| コマンド  | 説明   |
|---|--|
| <pre>Switch(config-if)# channel-group 1 ? mode Etherchannel Mode of the interface  Switch(config-if)#</pre>   | <p>チャンネル グループ番号を入力してから、<b>?</b>を入力して、コマンドラインに次に入力する内容を調べます。この例では、<b>mode</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないので、コマンドを完成するためにさらに多くの情報を入力する必要があります。</p>  |
| <pre>Switch(config-if)# channel-group 1 mode ? auto Enable PAgP only if a PAgP device is detected desirable Enable PAgP unconditionally on Enable Etherchannel only  Switch(config-if)#</pre> | <p><b>mode</b> キーワードを入力してから、<b>?</b>を入力して、コマンドラインに次に入力する内容を調べます。この例では、<b>auto</b>、<b>desirable</b>、または <b>on</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないので、コマンドを完成するためにさらに多くの情報を入力する必要があります。</p>          |
| <pre>Switch(config-if)# channel-group 1 mode auto ? &lt;cr&gt;  Switch(config-if)#</pre>  | <p>この例では、<b>auto</b> キーワードを使用します。<b>auto</b> キーワードを入力してから、<b>?</b>を入力して、コマンドラインに次に入力する内容を調べます。</p> <p>&lt;cr&gt; が表示されたので、Return を押せばコマンドを完成できます。さらにキーワードを一覧表示する場合は、キーワードを追加して Return を押し、コマンドを完成させることができます。</p> |
| <pre>Switch(config-if)# channel-group 1 mode auto Switch(config-if)#</pre>  | <p>この例では、Return を押して、コマンドを完成します。</p>   |

## コマンドモードの概要

Catalyst 4500 シリーズ スイッチの Cisco IOS のユーザ インターフェイスには、いくつかの異なるモードがあります。現在どのモードを実行しているかによって、使用できるコマンドが異なります。システム プロンプトに疑問符 (?) を入力すると、各コマンドモードで使用できるコマンドのリストを入手できます。

Catalyst 4500 シリーズ スイッチでセッションを開始するときには、ユーザ モード (別名、ユーザ EXEC モード) から始めます。EXEC モードで使用できるのは、限定的なコマンドサブセットです。すべてのコマンドにアクセスするには、特権 EXEC モードを開始する必要があります。特権 EXEC モードを開始するには、通常、パスワードを入力する必要があります。特権 EXEC モードからは、任意の EXEC コマンドを実行できるほか、グローバル コンフィギュレーション モードを開始できます。ほとんどの EXEC コマンドは、ある項目の現在のステータスを表示する **show** コマンドや、カウンタまたはインターフェイスをクリアする **clear** コマンドなどの 1 回限りのコマンドです。Catalyst 4500 シリーズ スイッチを再起動したときに、EXEC コマンドは保存されません。

コンフィギュレーション モードでは、実行コンフィギュレーションを変更できます。変更をコンフィギュレーションに保存すると、Catalyst 4500 シリーズ スイッチを再起動した場合に、変更が維持されます。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 コンフィギュレーション モード、サブインターフェイス コンフィギュレーション モード、およびその他のプロトコル固有モードを開始できます。

ROM モニタ モードは、Catalyst 4500 シリーズ スイッチが正常に起動しないときに使用する個別のモードです。Catalyst 4500 シリーズ スイッチまたはアクセス サーバの起動時に、有効なシステム イメージが検出されない場合、またはスタートアップ時にコンフィギュレーション ファイルが破壊されている場合、ROM モニタ モードが開始されることがあります。

表 1-3 に、Cisco IOS ソフトウェアの主要なコマンドモードの概要を示します。

表 1-3 主要コマンドモードの概要

| コマンドモード              | アクセス方式  | プロンプト              | 終了方法  |
|----------------------|---|--------------------|---|
| ユーザ EXEC             | ログインします。  | Switch>            | <b>logout</b> コマンドを入力します。   |
| 特権 EXEC              | ユーザ EXEC モードで、 <b>enable</b> EXEC コマンドを入力します。                     | Switch#            | ユーザ EXEC モードに戻るには、 <b>disable</b> コマンドを入力します。<br><br>グローバル コンフィギュレーション モードを開始するには、 <b>configure terminal</b> 特権 EXEC コマンドを入力します。  |
| グローバル コンフィギュレーション    | 特権 EXEC モードで、 <b>configure terminal</b> 特権 EXEC コマンドを入力します。       | Switch(config)#    | 特権 EXEC モードに戻るには、 <b>exit</b> コマンドまたは <b>end</b> コマンドを入力するか、 <b>Ctrl-Z</b> キーを押します。<br><br>インターフェイス コンフィギュレーション モードを開始するには、 <b>interface</b> コンフィギュレーション コマンドを入力します。  |
| 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、 <b>interface</b> コマンドを入力し、インターフェイスを指定します。 | Switch(config-if)# | 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> コマンドを入力します。<br><br>特権 EXEC モードに戻るには、 <b>exit</b> コマンドを入力するか、 <b>Ctrl-Z</b> キーを押します。<br><br>サブインターフェイス コンフィギュレーション モードを開始するには、 <b>interface</b> コマンドを入力し、サブインターフェイスを指定します。 |

表 1-3 主要コマンド モードの概要 (続き)

| コマンドモード                   | アクセス方式   | プロンプト                 | 終了方法   |
|---------------------------|--|-----------------------|--|
| サブインターフェイス<br>コンフィギュレーション | インターフェイス<br>コンフィギュレーション<br>モードで <b>interface</b> コマンドを入力し、サブインターフェイスを指定します。        | Switch(config-subif)# | 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> コマンドを入力します。<br><br>特権 EXEC モードに戻るには、 <b>end</b> コマンドを入力するか、 <b>Ctrl-Z</b> キーを押します。   |
| ROM モニタ                   | 特権 EXEC モードで、 <b>reload EXEC</b> コマンドを入力します。システムの起動後、60 秒以内に <b>Break</b> キーを押します。 | Rommon>               | ROM モニタ モードを終了するには、 <b>boot</b> コマンドを入力してイメージをリロードする必要があります。ファイル名または他の起動命令を指定せずに <b>boot</b> コマンドを使用すると、システムはデフォルトのフラッシュ イメージ (オンボードフラッシュ メモリ上の最初のイメージ) から起動します。また、特定のフラッシュ イメージから起動するように指定することもできます ( <b>boot system flash filename</b> コマンドを使用)。 |

コマンドモードの詳細については、『*Configuration Fundamentals Configuration Guide*』の「Using the Command Line Interface」の章を参照してください。

## コマンドの no 形式および default 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 **no** 形式があります。通常、コマンドの機能をディセーブルにする場合に **no** 形式のコマンドを指定します。ディセーブルにした機能を再びイネーブルにしたり、デフォルトでディセーブルに設定されている機能をイネーブルにしたりするには、**no** キーワードを付けないコマンドを使用します。たとえば、IP ルーティングはデフォルトでイネーブルに設定されています。IP ルーティングをディセーブルにするには、**no ip routing** コマンドを指定します。IP ルーティングを再びイネーブルにするには、**ip routing** コマンドを指定します。このマニュアルでは、コンフィギュレーション コマンドの完全な構文およびコマンドの **no** 形式について説明します。

一部のコンフィギュレーション コマンドには **default** 形式もあります。**default** 形式のコマンドは、コマンドの機能をデフォルト設定に戻します。ほとんどのコマンドはデフォルトでディセーブルに設定されているので、**default** 形式は **no** 形式と同じ結果になります。ただし、一部のコマンドはデフォルトでイネーブルに設定され、変数にデフォルト値が割り当てられています。このようなコマンドを **default** 形式で実行すると、コマンドがイネーブルになり、コマンドの変数がデフォルト値に戻されます。

## CLI スtring検索の使用

コマンド出力のパターンはStringと呼ばれます。CLI String検索機能を使用すると、**show** コマンドまたは **more** コマンド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。また、**--More--** プロンプトで検索およびフィルタリングを行うことができます。この機能は、大量の出力をソートする必要がある場合や、出力から不要な情報を除外する場合に役立ちます。

検索機能を使用すると、フィルタリングされていない出力で、指定された正規表現を含むものを最初の行から始めることができます。コマンド1つあたり、最大1つのフィルタを指定できます。または **--More--** プロンプトから新しい検索を開始できます。

正規表現は、ソフトウェアが **show** コマンドまたは **more** コマンド出力のマッチングを行うために用いるパターン（句、番号、またはより複雑なパターン）です。正規表現では、大文字と小文字は区別され、複雑な一致要件を可能にします。単純な正規表現の例は、**Serial**、**misses**、**138** などです。複合正規表現の例は、**00210...**、**( is )**、**[Oo]utput** などです。

3種類のフィルタリングを行うことができます。

- 指定した正規表現を含む行で出力を始めるには、**begin** キーワードを使用します。
- 指定した正規表現を含む出力行を含めるには、**include** キーワードを使用します。
- 指定した正規表現を含む出力行を除外するには、**exclude** キーワードを使用します。

次に、このフィルタリングされた出力を **--More--** プロンプトで検索できます。



(注)

CLI String検索機能を使用しても、以前の出力へ逆方向に検索またはフィルタリングすることはできません。また、HTTP を使用して CLI にアクセスしている場合は、フィルタリングを指定できません。

## 正規表現

正規表現は、コマンド出力内の同じ1つの文字に一致する1つの文字でもかまいませんし、コマンド出力内の同じ複数の文字に一致する複数の文字でもかまいません。このセクションでは、単一文字パターンおよび複数文字パターンを作成する方法、および繰り返し指定、選択、位置指定、およびカッコを用いたより複雑な正規表現を作成する方法について説明します。

## 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は任意の文字（**A ~ Z**、**a ~ z**）または数字（**0 ~ 9**）を使用できます。他のキーボード文字（**!** や **~** など）も単一文字パターンとして使用できますが、あるキーボード文字は、正規表現として用いられた場合に特別な意味を持ちます。表 1-4 に特別な意味を持つキーボード文字を一覧表示します。

表 1-4 特別な意味を持つ文字

| 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。   |
| ^  | Stringの最初と一致します。        |

表 1-4 特別な意味を持つ文字（続き）

| 文字          | 特別な意味  |
|-------------|--|
| \$          | Stringの最後と一致します。   |
| _ (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 ()), Stringの最初、Stringの最後、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入力する場合は、各文字の前にバックスラッシュ (\) を置いて特別な意味を持たないようにしてください。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です。

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

一連の単一文字パターンを指定して、コマンド出力とのマッチングを行うことができます。たとえば、a、e、i、o、またはuのうちの1つを含むStringに一致する正規表現を作成できます。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の1つだけがStringに存在しなくてはなりません。一連の単一文字パターンを指定するには、単一文字パターンを角カッコ ([ ]) で囲みます。次に例を示します。

```
[aeiou]
```

小文字アルファベットの5つの母音のうちの任意の1文字と一致します。

```
[abcdABCD]
```

小文字または大文字アルファベットの最初の4つの文字のうちの任意の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できます。上記の範囲は次のように簡略化できます。

```
[a-dA-D]
```

範囲に単一文字パターンとしてダッシュを追加する場合は、もう1つダッシュを追加して、その前にバックスラッシュを置きます。

```
[a-dA-D\-]
```

範囲に単一文字パターンとして右角カッコ (]) を含めることもできます。次のように記述してください。

```
[a-dA-D\]\]
```

上記の例は、大文字または小文字のアルファベットの最初の4文字のうちの任意の1文字、ダッシュ、または右角カッコに一致します。

範囲の最初にキャレット (^) を含めることにより、範囲の一致を逆にできます。次の例では、列挙された文字以外の任意の文字に一致します。

```
[^a-dqsv]
```

次の例では、右角カッコ (]) またはdという文字以外の任意の文字に一致します。

```
[^\]d]
```

## 複数文字パターン

正規表現を作成する場合、複数の文字を含むパターンを指定することもできます。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と複数文字パターン正規表現を作成できます。たとえば、`a4%` は複数文字の正規表現です。特別な意味を持つキーボード文字からその特別な意味をなくしたい場合には、キーボード文字の前にバックスラッシュを置きます。

複数文字パターンでは、順序が重要です。正規表現 `a4%` は、`a` という文字の後ろに `4` が続き、その後ろに `%` 記号が続く文字と一致します。そのストリングに `a4%` という文字がその順序で含まれていない場合、パターンマッチングは失敗します。この複数文字の正規表現では、

`a.`

は、ピリオド文字の特別な意味を使用し、`a` という文字の後ろに任意の文字が 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での有効な一致となります。

ピリオド文字の前にバックスラッシュを置くことにより、ピリオド文字の特別な意味をなくすことができます。次の表現では

`a\.`

`a.` というストリングのみが、この正規表現に一致します。

すべての文字、すべての数字、すべてのキーボード文字、または文字、数字、およびその他のキーボード文字の組み合わせを含む複数文字の正規表現を作成できます。次の例はすべて有効な正規表現です。

`telebit 3107 v32bis`

## 繰り返し指定

ある特殊文字を単一および複数文字パターンとともに使用することにより、指定された正規表現の繰り返しに一致する、より複雑な正規表現を作成できます。表 1-5 に正規表現の「繰り返し」を指定する特殊文字を一覧表示します。

表 1-5 繰り返し指定として使用される特殊文字

| 文字 | 説明                                   |
|----|--------------------------------------|
| *  | 0 個以上の単一または複数文字パターンに一致します。           |
| +  | 1 個以上の単一または複数文字パターンに一致します。           |
| ?  | 単一または複数文字パターンの 0 または 1 回の繰り返しに一致します。 |

次の例は、任意の数の文字 `a` (0 個を含む) に一致します。

`a*`

次のパターンでは、1 個以上の文字 `a` に一致している必要があります。

`a+`

次のパターンでは、ストリング `bb` または `bab` に一致します。

`ba?b`

次のストリングでは、任意の数のアスタリスク (\*) に一致します。

`\**`

複数文字パターンで繰り返し指定を使用する場合は、パターンをカッコで囲みます。次の例では、パターンは任意の数の複数文字String `ab` に一致します。

`(ab)*`

より複雑な例として、次のパターンは、1つまたは複数の英数字ペアに一致します（ただし、0、すなわち空Stringには一致しません）。

`([A-Za-z][0-9])+`

繰り返し指定（\*、+、または?）を使用している一致の記述順序では、最も長い構造が最初にマッチします。ネ스팅された構造でのマッチングは外側から内側へ行われます。連結構造では、構造の左側からマッチングされます。したがって、この正規表現は番号の前に文字が指定されるため、`A9b3` には一致しますが、`9Ab3` には一致しません。

## 選択

選択を使用すると、Stringとのマッチングに選択パターンを指定できます。選択パターンは、縦棒（|）で区切ります。選択肢のうちの1つだけが、Stringと一致できます。たとえば、正規表現

`codex | telebit`

は、`codex` または `telebit` というStringと一致できますが、`codex` と `telebit` の両方と一致することはできません。

## 位置指定

String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。つまり、Stringの最初または最後に固有のパターンが含まれるよう指定できます。Stringの特定の部分に対してこれらの正規表現の「位置指定」を行うには、表 1-6 に示す特殊文字を使用します。

表 1-6 位置指定に用いられる特殊文字

| 文字              | 説明               |
|-----------------|------------------|
| <code>^</code>  | Stringの最初と一致します。 |
| <code>\$</code> | Stringの最後と一致します。 |

次の正規表現がStringと一致するのは、Stringが `abcd` で始まるときだけです。

`^abcd`

対照的に、次の表現は、`a`、`b`、`c`、または `d` という文字以外のすべての単一文字に一致します。

`[^abcd]`

次の例では、正規表現は `.12` で終わるStringと一致します。

`$.12`

これらの位置指定文字を特殊文字アンダースコア（`_`）と対比させてください。アンダースコアは、Stringの始め（`^`）、Stringの終わり（`$`）、カッコ（`()`）、スペース（）、波カッコ（`{ }`）、カンマ（`,`）、またはアンダースコア（`_`）と一致します。アンダースコア文字を使用した場合、パターンはStringのどこに存在していてもかまいません。

たとえば、

```
_1300_
```

は、ストリング内のどこかに 1300 が含まれるストリングに一致します。ストリングの 1300 の前後にスペース、波カッコ、カンマ、またはアンダースコアを置くことができます。たとえば、

```
{1300_
```

は、正規表現に一致しますが、21300 および 13000 は一致しません。

アンダースコア文字を用いると、

```
^1300$ ^1300(space) (space)1300 {1300, ,1300, {1300} ,1300, (1300
```

のような長い正規表現リストを以下に置き換えることができます。

```
_1300_
```

## リコール機能用カッコ

「繰り返し指定」(p.1-9) に示すように、カッコを複数文字の正規表現とともに使用して、パターンを繰り返すことができます。また、単一文字パターンまたは複数文字パターンを囲むカッコを使用して、正規表現の他の場所で使用するパターンをリコールできます。

以前のパターンをリコールする正規表現を作成するには、リコールする特定のパターンを示すカッコ、バックスラッシュ (\)、整数という順序で記述します。これにより、記憶されたパターンを再び使用できます。整数は、正規表現パターン内のカッコの繰り返し数を示します。正規表現内に複数のリコールパターンがある場合、\1 は最初のリコールパターン、\2 は 2 番めのリコールパターン、というようになります。

次の正規表現は、リコール機能用カッコを使用します。

```
a(.)bc(.)\1\2
```

この正規表現は、a の後ろに任意の文字が来て (文字 1 と呼ぶ)、その次に bc が来て、その次に任意の文字 (文字 2 と呼ぶ) が来て、その次に再び文字 1 が来て、その次に再び文字 2 が来るストリングと一致します。したがって、この正規表現は aZbcTZT と一致します。ソフトウェアは、文字 1 が Z で文字 2 が T で、正規表現の後ろに再び Z および T が使用されることを覚えています。

## 設定変更の保存

設定変更をスタートアップ コンフィギュレーションに保存して、システムのリロードまたは停電時に設定変更が失われないようにするには、次のコマンドを入力します。

```
Switch# copy system:running-config nvram:startup-config  
Building configuration...
```

設定を保存するには 1 ~ 2 分かかります。設定が保存されたあと、次の出力が表示されます。

```
[OK]  
Switch#
```

ほとんどのプラットフォームでは、この手順により設定が NVRAM (不揮発性 RAM) に保存されます。クラス A フラッシュ ファイル システム プラットフォームでは、この手順により設定が CONFIG\_FILE 環境変数によって指定された場所に保存されます。CONFIG\_FILE 環境変数のデフォルトは NVRAM です。

## show platform コマンド

このコマンドは、テクニカル サポート担当者と共同作業しながら問題のトラブルシューティングを行う場合のみ、使用してください。テクニカル サポート担当者から指示があるまで、これらのコマンドは使用しないでください。



---

(注) **show platform** コマンドについては、このマニュアルで説明しません。

---