



正規表現、特殊文字、およびパターンの概要

この付録では、フィルタを使用してコマンド出力を検索する際に使用する正規表現、特殊文字またはワイルドカード文字、およびパターンについて説明します。フィルタ コマンドについては、「[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。

目次

- 「正規表現」(P.A-1)
- 「特殊文字」(P.A-2)
- 「文字パターンの範囲」(P.A-2)
- 「複数文字パターン」(P.A-3)
- 「乗算子を使用した複雑な正規表現」(P.A-3)
- 「パターンの選択」(P.A-4)
- 「アンカー文字」(P.A-4)
- 「アンダースコア ワイルドカード」(P.A-4)
- 「パターンの呼び出しに使用されるカッコ」(P.A-5)

正規表現

正規表現は、パターン（フレーズ、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が区別され、複雑な照合要件が許容されます。単純な正規表現には、`Serial`、`misses`、`138` などのエントリが含まれます。
- 複雑な正規表現には、`00210...`、`(is)`、`[Oo]output` などのエントリが含まれます。

正規表現は、単一文字パターンまたは複数文字パターンです。単一文字はコマンド出力内の同じ 1 つの文字と一致し、複数文字はコマンド出力内の同じ複数の文字と一致します。コマンド出力内のパターンは、ストリングと呼ばれます。

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。文字（A ~ Z および a ~ z）、数字（0 ~ 9）、およびその他のキーボード文字（! や ~ など）は、単一文字パターンとして使用できます。

特殊文字

キーボード文字の中には、正規表現で使用了場合に特別な意味を持つものがあります。表 A-1 に、特別な意味を持つキーボード文字を示します。

表 A-1 特別な意味を持つ文字

文字	特別な意味
.	ホワイトスペース（空白文字）を含む任意の 1 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。
?	パターンが 0 または 1 個一致します。
^	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、各文字の前に円記号 (\) を配置して特別な意味を除外します。次の例は、それぞれドル記号、アンダースコア、およびプラス記号と一致する単一文字パターンを示しています。

`\$ _ \+`

文字パターンの範囲

コマンド出力と一致する単一文字パターンの範囲を使用できます。単一文字パターンの範囲を指定するには、単一文字パターンを角カッコ ([]) で囲みます。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いずれか 1 つだけがストリングに含まれている必要があります。たとえば、`[aeiou]` は小文字のアルファベットの 5 つの母音のうちの任意の 1 文字と一致しますが、`[abcdABCD]` は小文字または大文字のアルファベットの 4 文字のうちの任意の 1 文字と一致します。

次の例のように、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、文字の範囲を簡素化できます。

`[a-dA-D]`

ダッシュを単一文字パターンとして検索範囲に追加するには、もう 1 つダッシュを含め、その前に円記号を配置します。

`[a-dA-D\]`

角カッコ (]) を単一文字パターンとして範囲に含めることもできます。

`[a-dA-D\]]`

範囲の先頭にキャレット (^) を含めると、範囲のマッチングの論理が反転します。次の例では、リストされている文字以外の文字と一致します。

`[^a-dqsv]`

次の例では、右角カッコ (]) または文字 d 以外のすべてと一致します。

`[^\]]d]`

複数文字パターン

複数文字の正規表現は、文字、数字、および特別な意味を持たないキーボード文字を連結することによって作成できます。複数文字パターンでは、順序が大切です。正規表現 `a4%` は、後に `4` が続き、その後に `%` が続く文字 `a` と一致します。ストリングに `a4%` がこの順序で含まれない場合、パターンマッチングは失敗します。

複数文字の正規表現 `a.` は、ピリオド文字が持つ特別な意味により、後に任意の 1 文字が続く文字と一致します。この例と同様、ストリング `ab`、`a!`、および `a2` すべて有効な正規表現です。

文字が文字どおりに解釈されるように指定するには、特別な意味を持つキーボード文字の前に円記号を配置します。ピリオド文字の特別な意味を除外するには、ピリオドの前に円記号を配置します。たとえば、コマンド構文で表現 `a\.` を使用している場合は、ストリング `a.` だけが一致します。

すべて文字、すべて数字、すべてキーボード文字で構成される複数文字の正規表現、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は、有効な正規表現です (例: `telebit 3107 v32bis`)。

乗算子を使用した複雑な正規表現

乗算子を使用すると、指定した正規表現が複数個一致するように Cisco IOS XR ソフトウェアに指令する、より複雑な正規表現を作成できます。表 A-2 に、正規表現との「複数個の一致」を指定する特殊文字を示します。

表 A-2 乗算子として使用される特殊文字

文字	説明
*	0 個以上の単一文字パターンまたは複数文字パターンと一致します。
+	1 個以上の単一文字パターンまたは複数文字パターンと一致します。
?	単一文字パターンまたは複数文字パターンが 0 または 1 個一致します。

次の例では、文字 `a` が任意の個数 (0 個を含む) 一致します。

`a*`

次のパターンは、ストリング内の文字 `a` が 1 個以上一致するように要求します。

`a+`

次のパターンは、ストリング `bb` または `bab` と一致します。

`ba?b`

次のストリングは、任意の数のアスタリスク (*) と一致します。

`¥**`

乗算子を複数文字パターンと共に使用するには、パターンをカッコで囲みます。次の例では、パターンが、任意の数の複数文字ストリング `ab` と一致します。

`(ab)*`

より複雑な例として、次のパターンは、英数字のペアの 1 個以上のインスタンスと一致します。

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

乗算子 (*、+、および ?) を使用した照合の順序は、最長のコンストラクトが最初になります。ネストされたコンストラクトは外側から内側へ照合されます。連結されたコンストラクトは左側から照合されます。したがって、この正規表現では、数字の前に文字が指定されているため、**A9b3** は一致しますが、**9Ab3** は一致しません。

パターンの選択

パターン選択を使用すると、パターンの選択肢を指定してストリングを照合できます。パターンの選択肢は、縦棒 (|) で区切ります。選択肢のいずれか 1 つだけがストリングと一致します。たとえば、正規表現 **codex|telebit** は、ストリング **codex** と **telebit** の両方と一致するのではなく、ストリング **codex** または **telebit** のいずれか一方と一致します。

アンカー文字

アンカーを使用すると、ストリングの先頭または末尾に対して正規表現のパターンを照合できます。[表 A-3](#) に示す特殊文字を使用すると、正規表現をストリングの一部にアンカーできます。

表 A-3 アンカーに使用される特殊文字

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

たとえば、正規表現 **^con** は、**con** で始まる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。

^ は、ストリングの先頭を示すだけでなく、角カッコで囲まれた範囲内で使用すると、論理関数「not」を示す意味で使用できます。たとえば、正規表現 **[^abcd]** は、**a**、**b**、**c**、および **d** を除く任意の 1 文字と一致する範囲を示します。

アンダースコア ワイルドカード

アンダースコアを使用すると、ストリングの先頭 (^)、ストリングの末尾 (\$)、カッコ (()), スペース (), 波カッコ ({}), カンマ (,), およびアンダースコア (_) が一致します。アンダースコアを使用すると、ストリング内にパターンが存在することを指定できます。たとえば、**_1300_** は、ストリング内に **1300** を含み、その前後にスペース、波カッコ、カンマ、またはアンダースコアが続く任意の文字列と一致します。**_1300_** は、正規表現 **{1300_}** と一致しますが、正規表現 **21300** および **13000t** とは一致しません。

アンダースコアを使用して、長い正規表現のリストを置き換えることができます。たとえば、**^1300() ()1300\$ {1300, ,1300, {1300} ,1300, (1300,** と指定する代わりに、単に **_1300_** と指定できます。

パターンの呼び出しに使用されるカッコ

特定のパターンを複数個呼び出すには、複数文字正規表現でカッコを使用します。IOS XR では、正規表現の他の部分でできるように、パターンを記憶しておくことができます。

以前のパターンを呼び出す正規表現を作成するには、カッコを使用して特定のパターンをメモリに記憶することを示し、円記号 (¥) に続いて数字を指定することにより、記憶したパターンを再使用することを示します。この数字には、正規表現パターン内でのカッコの個数を指定します。たとえば、正規表現内に複数の記憶パターンが存在する場合、¥1 は最初の記憶パターンを示し、¥2 は 2 番目の記憶パターンを示します。

次の正規表現は、パターン呼び出しのカッコを使用しています。

a(.)bc(.)¥1¥2

この正規表現は、後に任意の文字（文字番号 1 とする）が続き、その後に任意の文字（文字番号 2 とする）が続き、そのまた後に文字番号 1 が再び続き、最後に文字番号 2 が再び続く文字 **a** と一致します。したがって、この正規表現は **aZbcTZT** と一致することが考えられます。この正規表現では、文字番号 1 は Z、文字番号 2 は T として記憶され、Z および T は後で再使用されます。

■ パターンの呼び出しに使用されるカッコ