CLI の概要

IPS 5.1 の CLI では、Telnet、SSH、およびシリアル インターフェイス接続を使用してセンサーにアクセスできます。

この章は、次の内容で構成されています。

ユーザ ロール

IPS 5.1 の CLI を使用すると、複数のユーザが同時にログインできます。ローカル センサーからユーザを作成および削除することも可能です。一度に変更できるユーザ アカウントは 1 つだけです。各ユーザはロールに関連付けられ、そのロールによって各ユーザの変更できる範囲が制御されます。

CLI では、管理者、オペレータ、ビューア、およびサービスの 4 つのユーザ ロールがサポートされています。各ロールの権限レベルが異なるので、メニューおよび使用可能コマンドも各ロールで異なります。

- **管理者**：このユーザ ロールは、最高レベルの権限を持っています。管理者には無制限の表示アクセス権があり、次の機能を実行できます。
  - ユーザの追加とパスワードの割り当て
  - 物理インターフェイスおよび仮想センサーの制御の有効または無効化
  - 仮想センサーへの物理センシング インターフェイスの割り当て
  - エージェントの構成または表示時における、センサーに接続可能なホストのリストの変更
  - センサー アドレス コンフィギュレーションの変更
  - シグニチャの調整
  - 仮想センサーへのコンフィギュレーションの割り当て
  - ルータの管理
- **オペレータ**：このユーザ ロールには、2 番目に高い権限があります。オペレータには無制限の表示アクセス権があり、次の機能を実行できます。
  - 自分のパスワードの変更
  - シグニチャの調整
  - ルータの管理
  - 仮想センサーへのコンフィギュレーションの割り当て
- **ビューア**：このユーザ ロールには、最低位レベルの権限があります。ビューア ユーザはコンフィギュレーションおよびイベント データを表示でき、自分のパスワードを変更できます。

ヒント　モニタリング アプリケーションには、センサーに対するビューア アクセス権のみが必要です。CLI を使用してビューア権限を持つユーザ アカウントをセットアップし、その後イベント ビューアを構成してこのアカウントでセンサーに接続できます。

- **サービス**：このユーザ ロールには CLI への直接アクセス権はありません。サービス アカウント ユーザは、bash shell（Bourne-again shell）に直接ログインします。このアカウントは、サポートおよびトラブルシューティングの目的でのみ使用します。許可のない変更はサポートされていません。許可のない変更を行うと、適切な操作を保証するために、デバイスのイメージの再作成が必要となります。サービス ロールを持つユーザを 1 つだけ作成できます。

  サービス アカウントにログインすると、次の警告が表示されます。

```
****************************** WARNING ****************************************
UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED.
This account is intended to be used for support and troubleshooting purposes only.
Unauthorized modifications are not supported and will require this device to be
re-imaged to guarantee proper operation.
*******************************************************************************
```

（注）　サービス ロールは、必要に応じて CLI をバイパスできる特別なロールです。管理者権限を持つユーザだけがサービス アカウントを編集できます。

CLI の動作

IPS の CLI を使用するときは、次のヒントに従ってください。

プロンプト

- CLI コマンドに表示されるプロンプトは変更できません。
- システムが質問を表示して、その答えの入力を待つ場合は、ユーザ対話型プロンプトとなります。デフォルトの入力は大カッコ [ ] 内に表示されます。デフォルトの入力を受け入れるには、**Enter** キーを押します。

ヘルプ

- コマンドのヘルプを表示するには、コマンドの後に **?** を入力します。

  次の例で、? の機能を示します。

```
sensor# configure ?
terminal     Configure from the terminal
sensor# configure
```

  （注）　ヘルプの表示からプロンプトに戻ると、前に入力したコマンドが ? なしで表示されます。

- 不完全なトークンの後に **?** を入力して、コマンドを完成させる有効なトークンを参照することもできます。トークンと ? の間にスペースがあると、ambiguous command エラーが表示されます。

```
sensor# show c ?
% Ambiguous command : "show c"
```

  スペースなしでトークンを入力すると、それを完成させるために選択可能なトークンが表示されます（ヘルプ説明なし）。

```
sensor# show c?
clock configuration
sensor# show c
```

- 現在のモードで使用できるコマンドだけが、ヘルプで表示されます。

タブ補完

- 現在のモードで使用できるコマンドだけが、タブ補完およびヘルプで表示されます。
- コマンドの完全な構文が不明な場合は、コマンドの一部を入力して **Tab** を押すと、コマンドを完成できます。
- タブ補完に複数のコマンドが一致する場合は、何も表示されません。

再呼び出し

- モードで入力したコマンドを再呼び出しするには、上または下矢印キーを使用するか、**Ctrl+P** キーまたは **Ctrl+N** キーを押します。

  （注）　ヘルプおよびタブ補完の要求は、再呼び出しリストには表示されません。

- 再呼び出しリストの最後に、ブランクのプロンプトが表示されます。

大文字小文字の区別

- CLI は大文字小文字を区別しませんが、入力と同じ大文字小文字の型でテキストをエコーバックします。たとえば、次のようになります。

  ```
  sensor# CONF
  ```

  Tab を押すと、センサーに次のように表示されます。

  ```
  sensor# CONFigure
  ```

表示オプション

- –More– は、対話型プロンプトで、端末出力が割り当てられた表示スペースを超えたことを示します。残りの出力を表示するには、**スペースバー**を押して次ページの出力を表示するか、または **Enter** キーを押して一度に 1 行ずつ出力を表示します。
- 現在行の内容をクリアして、ブランクのコマンドラインに戻るには、**Ctrl+C** キーを押します。

コマンドライン編集

表 1-1 は、CLI で使用できるコマンドライン編集機能を示しています。

表 1-1　コマンドライン編集

キー	説明
Tab	部分的なコマンド名入力を補完します。コマンドを 1 つに特定できるところまで文字を入力して、Tab を押すと、コマンド名が補完されます。複数のコマンドを示す可能性がある文字を入力すると、警告音が鳴ってエラーが示されます。部分的なコマンドの直後（スペースなし）に疑問符（?）を入力してください。その文字列で始まるコマンドのリストが表示されます。
Backspace	カーソルの左側の文字を消去します。
Enter	コマンドラインで、Enter を押すとコマンドが処理されます。端末画面の `---More---` プロンプトで Enter キーを押すと、行が下にスクロールします。
スペースバー	端末画面で、追加の出力を表示できます。画面に `---More---` 行が表示されているときにスペースバーを押すと、次画面が表示されます。
左矢印	カーソルを 1 文字左に移動します。1 行を超えるコマンドを入力した場合、左矢印キーを繰り返し押すと、システム プロンプトの方にスクロールバックし、コマンド入力の開始部分を検証できます。
右矢印	カーソルを 1 文字右に移動します。
上矢印または Ctrl+P キー	履歴バッファ内のコマンドを、最新のコマンドから再呼び出しします。より古いコマンドへと順に連続して再呼び出しするには、キー シーケンスを繰り返します。
下矢印または Ctrl+N キー	上矢印または Ctrl+P キーでコマンドを再呼び出しした後、履歴バッファ内のより新しいコマンドに戻ります。より新しいコマンドへと順に連続して再呼び出しするには、キー シーケンスを繰り返します。
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 文字後に移動します。
Ctrl+D	カーソルの位置の文字を削除します。
Ctrl+E	カーソルをコマンドラインの末尾に移動します。
Ctrl+F	カーソルを 1 文字前に移動します。
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	画面を消去して、システム プロンプトとコマンドラインを再表示します。
Ctrl+T	カーソルの左側の文字をカーソル位置の文字で置き換えます。
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	コードを挿入して、直後の入力を編集キー *ではなく*、コマンド入力として処理することをシステムに指示します。
Ctrl+W	カーソルの左側の語を削除します。
Ctrl+Y	削除バッファ内の最新のエントリを再呼び出しします。削除バッファには、削除または切り取りをした最新の 10 項目が格納されています。
Ctrl+Z	コンフィギュレーション モードを終了して、EXEC プロンプトに戻ります。
Esc+B	カーソルを 1 語後に移動します。
Esc+C	カーソル位置の語を大文字にします。
Esc+D	カーソル位置から語の末尾までを削除します。
Esc+F	カーソルを 1 語前に移動します。

表 1-1　コマンドライン編集（続き）

キー	説明
Esc+L	カーソル位置の語を小文字に変更します。
Esc+U	カーソル位置から語の末尾までを大文字にします。

IPS コマンド モード

IPS の CLI には、次のコマンド モードがあります。

- 特権 EXEC：CLI インターフェイスにログインすると、このモードになります。
- グローバル コンフィギュレーション：特権 EXEC モードから **configure terminal** と入力すると、このモードになります。

  コマンド プロンプトは sensor(config)# です。
- サービス モード 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 **service** *service-name* と入力すると、このモードになります。

  コマンド プロンプトは sensor(config-ser)# です。ここで、ser はサービス名の先頭の 3 文字です。
- マルチインスタンス サービス モード：グローバル コンフィギュレーション モードから **service** *service-name log-instance-name* と入力すると、このモードになります。

  コマンド プロンプトは sensor(config-log)# です。ここで、log はログ インスタンス名の先頭の 3 文字です。システムのマルチインスタンス サービスは、シグニチャ定義とイベント アクション ルールのみです。

正規表現の構文

正規表現は、文字列の照合に使用されるテキスト パターンです。正規表現には平文テキストと特殊文字が混在し、どのような照合をするかを指定します。たとえば、数字を検索する場合の正規表現は「[0-9]」です。大カッコは、比較される文字が大カッコで囲まれたいずれか 1 つの文字と一致することを示します。0 と 9 の間のハイフン（-）は、0 から 9 までの範囲であることを示します。したがって、この正規表現は 0 から 9 のいずれかの文字（つまり、数字）と一致します。

特定の特殊文字を検索するには、特殊文字の前に \ 記号を使用する必要があります。たとえば、単一文字の正規表現「*」は、単一のアスタリスク（*）と一致します。

この項で定義されている正規表現は、POSIX Extended Regular Expression 定義のサブセットと類似しています。特に、「[..]」、「[==]」、および「[::]」表現は、サポートされていません。ただし、単一文字を表すエスケープ表現はサポートされています。文字は 16 進値で表現できます。たとえば、\x61 は「a」に相当するため、\x61 は文字「a」を表すエスケープ表現になります。

表 1-2 に、特殊文字の一覧を示します。

表 1-2　正規表現の構文

文字	説明
^	文字列の先頭。「^A」表現は、文字列の先頭でだけ「A」と一致します。
^	左大カッコ（[）の直後。対象の文字列との照合から大カッコ内にある文字を除外します。「[^0-9]」表現は、対象文字が数字ではないことを示します。
$	文字列の末尾との照合。「abc$」表現は、文字列の一部「abc」が文字列の末尾にある場合のみ一致します。
|	両側の表現を対象の文字列と照合します。「a|b」表現は、「a」および「b」と一致します。
.	任意の文字と一致します。
*	表現内のアスタリスクの左側にある文字が 0 個以上一致することを示します。
+	アスタリスク（*）の場合と似ていますが、表現内の + 記号の左側の文字が 1 つ以上一致する必要があります。
?	その左側の文字が 0 または 1 回一致します。
()	パターン評価の順序に影響し、また一致した文字列の一部を別の表現に置換するとき使用されるタグ付き表現としても機能します。
[]	文字セットを囲む大カッコ（[および]）は、囲まれた文字のいずれかが対象の文字と一致することを示します。
\	この記号が使用されない場合に特別に解釈される文字の指定を可能にします。 \xHH は、その値が（HH）、つまり 16 進数値 [0-9A-Fa-f] で表現される値と同じであることを示します。値はゼロ以外にする必要があります。 BEL は \x07 と同じで、BS は \x08、FF は \x0C、LF は \x0A、CR は \x0D、TAB は \x09、そして VT は \x0B と同じです。 他の文字「c」の場合、「\c」は「c」と同じで、特別に解釈されることはありません。

次に、特殊文字の例を示します。

- `a*` は、任意の数の文字 a と一致します（なしも含む）。
- `a+` では、少なくとも 1 つの文字 a が一致する文字列に存在する必要があります。

- **ba?b** は、文字列 bb または bab と一致します。
- ****** は、任意の数のアスタリスク（*）と一致します。

複数文字のパターンの乗数を使用するには、パターンをカッコで囲みます。

- **(ab)*** は、任意の数の複数文字列 ab と一致します。
- **([A-Za-z][0-9])+** は 1 つ以上の英数字の組み合せと一致します。ただし、なしは対象としません（つまり、空の文字列は一致しない）。

乗数（*、+、または ?）を使用した照合の順序は、最も長い指定文字列が最初になります。ネスト化された指定文字列は、外側から内側に照合されます。連結された指定文字列は、その左側から照合されます。したがって、正規表現は A9b3 とは一致しますが、9Ab3 とは一致しません。文字が数字の前に指定されているためです。

単一または複数文字のパターンをカッコで囲み、正規表現の別の場所で使用するパターンをソフトウェアに覚えておくように指示することもできます。

以前のパターンを再呼び出しする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のメモリを指定し、\ 記号の後に数字を続けてどの記憶されたパターンを再使用するかを指定します。数字は、正規表現パターン内のカッコの出現位置を指定します。正規表現に複数の記憶されたパターンがある場合、\1 は最初に記憶されたパターン、\2 は 2 番目に記憶されたパターン（以降も同様）を示します。

次の正規表現は、再呼び出しにカッコを使用しています。

- **a(.)bc(.)\1\2** は、*a* とそれに続く任意の文字、その後に *bc* と任意の文字が続き、さらに最初の*任意*の文字が再度続き、2 番目の*任意*の文字が再度続きます。

  たとえば、正規表現は aZbcTZT と一致します。最初の文字は Z で、2 番目の文字は T であることがソフトウェアで記憶され、その後 Z と T が再度、正規表現に使用されます。

汎用 CLI コマンド

次の CLI コマンドは、IPS 5.1 で汎用的に使用されます。

- **configure terminal**：グローバル コンフィギュレーション モードに入ります。

  グローバル コンフィギュレーション コマンドは、1 つのプロトコルやインターフェイスだけでなく、システム全体に影響を及ぼす機能に適用されます。

```
sensor# configure terminal
sensor(config)#
```

- **service**：次のコンフィギュレーション サブモードに入ります。analysis-engine、authentication、event-action-rules、host、interface、logger、network-access、notification、signature-definition、ssh-known-hosts、trusted-certificates、および web-server。

  （注） event-action-rules サブモードと signature-definition サブモードは、マルチインスタンス サービスです。それぞれ 1 つの事前定義されたインスタンスのみがサポートされます。event-action-rules では、サポートされるインスタンスは rules0 だけです。signature-definition では、サポートされるインスタンスは sig0 だけです。

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

- **end**：コンフィギュレーション モードまたは任意のコンフィギュレーション サブモードを終了します。最上位の EXEC メニューに戻ります。

```
sensor# configure terminal
sensor(config)# end
sensor#
```

- **exit**：任意のコンフィギュレーション モードを終了、またはアクティブなターミナル セッションを閉じて、EXEC モードを終了します。直前のメニュー セッションに戻ります。

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)# exit
sensor(config)# exit
sensor#
```

CLI キーワード

一般的に、機能を無効にするには、コマンドの **no** 形式を使用します。キーワード **no** を指定しないでコマンドを使用すると、無効になっている機能を有効にできます。たとえば、**ssh host-key** *ipaddress* コマンドを入力すると既知のホスト テーブルにエントリが追加され、**no ssh host-key** *ipaddress* コマンドを入力すると既知のホスト テーブルからエントリが削除されます。そのコマンドの **no** 形式の動作の詳細については、個々のコマンドを参照してください。

サービス コンフィギュレーション コマンドには、default 形式も使用できます。**default** 形式のコマンドを使用すると、コマンド設定をデフォルトに戻すことができます。このキーワードは、アプリケーションのコンフィギュレーションに対して使用する **service** サブメニュー コマンドに適用されます。コマンドで **default** を指定すると、パラメータがデフォルト値にリセットされます。コマンドで **default** キーワードを指定できるのは、コンフィギュレーション ファイルにデフォルト値を指定できるコマンドのみです。