



Junos® ソフトウェア基本シリーズ

DAY ONE: JUNOS CLIの探求

It's day one and you have a job to do. Start using the Junos CLI today. In this Day One booklet you're just a few hours away from modifying, saving, and loading configuration files on your Junos device.

著者: キャシー・ガデッキ、マイケル・スクラッグス

JUNOS® ソフトウェア基本シリーズ

Day One: JUNOS CLIの探究

著者: キャシー・ガデッキ、マイケル・スクラッグス

第1章: CLIについて	1
第2章: 基本操作	7
第3章: 運用モードを理解する	23
第4章: 設定モードを使いこなす	31
参考資料	57

© 2009 by Juniper Networks, Inc. All rights reserved.

Juniper Networks, Juniper Networks のロゴ、JUNOS、NetScreen、ScreenOS は、Juniper Networks, Inc. (以下、ジュニパーネットワークス) の米国およびその他の国における登録商標です。JUNOSe はジュニパーネットワークスの商標です。その他の商標、サービスマーク、登録商標、登録サービスマークは、それぞれの権利者に帰属する知的財産です。

ジュニパーネットワークスは、本書に誤りが含まれることがあっても責任を負いません。ジュニパーネットワークスは予告なく本書を変更、修正、転載、別の形態に改訂する権利を留保します。ジュニパーネットワークスが製造、販売する製品、あるいはその部品は、ジュニパーネットワークスが保有する、あるいはライセンスを受けた以下の米国特許のうち 1 件または複数により保護されている場合があります。米国特許第 5,473,599 号、第 5,905,725 号、第 5,909,440 号、第 6,192,051 号、第 6,333,650 号、第 6,359,479 号、第 6,406,312 号、第 6,429,706 号、第 6,459,579 号、第 6,493,347 号、第 6,538,518 号、第 6,538,899 号、第 6,552,918 号、第 6,567,902 号、第 6,578,186 号、第 6,590,785 号。

発行 : ジュニパーネットワークス
印刷 : Vervante.com (米国)
編集 : ナンシー・ケルベル
出版管理 : Ames & Eames, Inc.

改訂 : 第 2 版、2009 年 4 月

著者の紹介

キャシー・ガデッキはジュニパーネットワークスのシニア製品マーケティングマネージャで、JUNOS ソフトウェアの責任者です。ガデッキには JUNOS ソフトウェアの実用ガイドブック「JUNOS for Dummies」(共著)などの著書があります。

マイケル・スクラッグスはジュニパーネットワークスの技術出版マネージャです。

著者の謝辞

著者は、本書の制作に手を貸して下さった多くの方々にお礼を申し上げます。このプロジェクトの出版管理マネージャであるパトリック・エイムズ氏 (Ames & Eames) は、出版分野での豊富な知識と経験を「Day One シリーズ」の性格付けと、この最初のブックレットの実現に活かしてくださいました。ナンシー・ケルベル氏には私たちの原稿の編集でお世話になりました。スーザン・ハリス氏には本書の校正段階で調査と草稿の作成を手伝っていただきました。ジョナサン・ルーニー氏とマイケル・ビュシオン氏はサンプルと演習問題に加えて、幅広い専門知識を提供してくださいました。そして、ジュニパーネットワークスの技術出版部門と教育サービス部門に原稿を提供されているたくさんの方々のライターの方々には、参考となる資料、記述、サンプルなどを提供していただきました。マイケル・バティア氏とバーナデット・ウイリス氏の助力にも感謝します。

Day One へようこそ

Day One ブックレットは、あなたが新しいテーマにすぐに取りかかれるように、作業の初日 (Day One) に必要となる情報だけをまとめたものです。

Day One ブックレットでは、まわりくどい説明は避け、ステップごとに手順を説明し、そのまま使えそうな詳しいサンプルを数多く掲載するように心がけています。「試してみよう」のセクションでは、あなたのノート PC カターミナルで、コマンド、設定、その他の手順を独習できるようになっています。ブックレットを読み進みながら欄外の注記にも目を向けて下さい。そこには「**おすすめ**」、「**警告!**」、「**ショートカット**」、「**ヒント**」、「**注**」などがあります。

なぜ Day One ブックレットなのか？

前提条件はシンプルです。つまり、あなたがジュニパーの機器をなるべく短時間で、しかも効果的に使いこなしたいと思っていることです。あなたには入手可能な技術資料すべてに目を通して時間を余裕はありません。どこから手を着ければよいのかも分からないかもしれません。知りたいのは初日に何をすべきかということだけです。JUNOS ソフトウェア基本シリーズは、そんなあなたの目的にぴったりです。いちどに読む必要があるのはブックレット 1 冊だけです。

詳しく知りたい

Day One ブックレットは実際に何かを試みることを主眼にしているので、サンプルをたくさん掲載してご自分で調べられるようになっています。お望みなら、ご自分でサンプルを試してみましょ。掲載されたサンプルは等幅フォントによる画面出力で、あなたが入力するコマンドは**太字**で示されています。

始める前に

JUNOS CLI (コマンドライン・インタフェース) にアクセスするには、まずデバイス自体にアクセスできなければなりません。これは、ネットワークを使わずに直接コンソールポートへアクセスするのでも、ネットワーク経由で管理用ポートへアクセスするのでもかまいません。ネットワークごとに差異があるため、導入済みの機器にログインする手順を逐一説明することはこのブックレットが目的とする範囲を超えます。そこでログインする前に、あなたのネットワークがどのように設定されているかを知っておくか、あるいはデバイスに物理的にアクセスする手段を確保しておいてください。

デバイスへのアクセス方法についての情報が必要な場合は、該当する製品の「**Quick Start**」ガイドをご覧ください。このガイドは次の URL にあります。

<http://www.juniper.net/techpubs/>

このブックレットで、できるようになること

「*Day One: JUNOS CLI の探究*」は、あなたが各種のコマンドとコマンドライン・インタフェースの仕組みを理解するのを支援します。この後に続く JUNOS ソフトウェア基本シリーズの各ブックレットは、その理解をさらに深め、あなたがデバイスの設定を完了させてネットワーク内で稼働させることができるように支援します。

このブックレットによる学習を終了すると、あなたは以下のことができるようになります。

- ✓ JUNOS ソフトウェアで動作しているデバイス上で、CLI の運用モードと設定モードをナビゲートする。
- ✓ 各モードの基礎となる階層構造を理解する。
- ✓ オンボードのヘルプ機能を活用し、キーボード・ショートカットを使用して、作業を効率化する。
- ✓ 運用モードで、デバイスのステータス、アラーム、その他の役に立つ情報を表示する。
- ✓ 運用に及ぼすリスクを最小限に抑えて、設定ファイルを修正、セーブ、ロードする。
- ✓ 基本的な設定コマンドを使用する (`show`、`set`、`delete` など)。
- ✓ JUNOS ソフトウェアのコミットモデルの安全機能を活用する。
- ✓ システムの変更を事前に準備する。
- ✓ ショートカットやベテランユーザーからのヒントを利用して、一般的なトラブルを回避する。

お読みになる前に知っておいていただきたいこと

「*Day One:JUNOS CLI*の探究」の内容は、JUNOSソフトウェアを学習し、後続の*Day One*ブックレット(JUNOSソフトウェア基本シリーズとして刊行)を理解する上で、基礎知識となるものです。

「*Day One:JUNOS CLI*の探究」は、JUNOSまたはジュニパー製品を初めてご利用になるユーザーを対象としていますが、ベテランのJUNOS管理者にも参考あるいは知識のリフレッシュとして役立つように書かれています。また、基本的なネットワーク技術の概念を理解していれば、このブックレットによる学習を進める上で助けとなるでしょう。

ジュニパーへのフィードバック方法

*Day One*ブックレットとして実現できるご意見や素晴らしいアイデアをお持ちでしょうか。建設的なご意見やご批判をお待ちしています。お時間のあるときに、どうぞメール(英文)で、dayone@juniper.netまでお知らせください。

飛び乗ろう!

では、待ちきれない人のために一気に先に進んで、新品の JUNOS ベースのデバイス(工場出荷時の設定になっているもの)で、発見の道に入ってみましょう。それには以下の 4 つのステップを順に進みます。

ステップ 1: 接続

あなたの管理用 PC と新品のジュニパーデバイスのコンソールポートをクロスケーブル (ヌルモデム) で接続し、PC の電源を投入して端末エミュレータを起動してから、ジュニパーデバイスの電源を投入します。

ステップ 2: ログイン

ログインプロンプトで **root** と入力し、パスワードプロンプトで **Enter** キーを押します。以下のように表示されます。太字はユーザー入力の文字列です。

```
host (ttyd0)
login: root
password:
```

ステップ 3: CLI に入る

認証が完了したら、UNIX シェルに入ります。プロンプトから CLI に入るには、**cli** と入力します。次のように表示されます。太字はユーザー入力の文字列です。

```
root@Amnesiac% cli
```

ステップ 4: 探究

コマンドライン・インタフェースの運用モードに入りました。コマンド階層のどこでも、疑問符「?」を入力すると、そのとき入力として有効な文字列のリストが表示されます。さあ、やってみましょう。? を入力します。

表示されるのは、運用モードの最上位の階層で有効なコマンドのリストです。? はコマンドの記述を完結させるために使える文字列を見つける用途でも使えます。例えば次のように使います。

```
user@Amnesiac > show i?
```

こうすると、**show** コマンドの **i** という文字で始まるオプションがすべて表示されます。いろいろ考えて、他に何を発見できるかを想像しましょう。例えば、**configure?** と入力するとどうなるでしょう。

第 1 章

CLIについて

コマンドモードの概要	2
運用コマンドの階層を理解する.....	3
設定ステートメントの階層を理解する.....	4

CLI (コマンドライン・インタフェース) は、ソフトウェア・インタフェースで、これを使用してあなたのデバイスにアクセスします。CLIを利用して、プラットフォームを設定し、その運用状況をモニタし、必要に応じて設定を調整します。

他のネットワークデバイスを操作したことがあるユーザーは、JUNOS CLIに親しみを感じるはずですが、見たことのないコマンドがいくつか追加されていることにも、すぐに気づくことでしょう。悩む必要はありません。この変更で、新しいツールと保護機能が豊富に提供されて、ネットワークを効率よく管理し、高い稼働率を維持するのに役立つのです。

CLIには、たくさんのショートカットと、ヘルプを利用するためのコマンドが含まれます。このショートカットとコマンドを習得すれば、キーボードをたたく時間はずっと短くなるでしょう。ほんの少し努力すれば、なぜこれほど多くの人々が、JUNOSは時間の(たいていは大幅な)節約になり、反復作業を軽減し、ミス防止に役立つと言うのかがお分かりになると思います。

J-Web

CLIよりもWeb GUIのほうが好きという場合は、ざっとJ-Webを見てみましょう。これはJUNOSデバイスで利用できる強力なWebベースの管理用インタフェースです。J-WebでもCLIと同様の操作を行うことができます。J-Webで提供される実用的なツールを使用して、デバイスをモニタし、設定し、トラブルシューティングを行い、管理することができます。

もっと詳しく? あなたのデバイス向けのJ-Webの技術資料を次のURLで参照して下さい。

www.juniper.net/techpubs/

コマンドモードの概要

JUNOS CLI 探究の第一歩は、2種類のコマンドモードを理解することです。

- **運用モード**: デバイスの運用状況を管理し、モニタする。例えば、デバイスインタフェースのステータスをモニタしたり、シャーシの警報をチェックしたり、デバイスのオペレーティングシステムをアップグレードあるいはダウングレードする。
- **設定モード**: デバイスと、そのインタフェースを設定する。これには、ユーザーアクセス、インタフェース、プロトコル、セキュリティサービス、システム・ハードウェア・プロパティが含まれる。

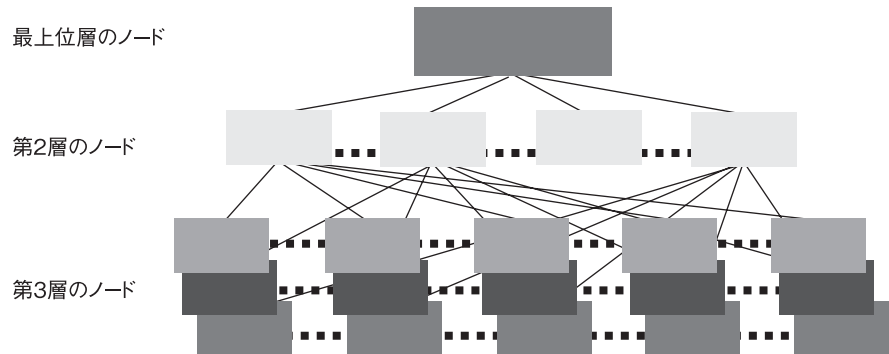


図1.1 JUNOS CLIのコマンドモード階層構造

JUNOS CLIは、各モードでのアクティビティ(機能の要素)を、図1.1のように複数の階層で構造化しています。各モードの階層構造は、共通して使用される機能が段階的に枝分かれしながら連結されたカスケード型の構造を集めたものです。

階層的に構造化されているということは、JUNOS CLIが持つ数多くのユニークな特徴の1つであり、ユーザーに好感を持たれています。JUNOS CLIは、複数のアクティビティを論理的なグループにまとめることで、規則性と一貫性のあるシンタックスを提供しています。これは、あなたが階層のどこにいるのかを知り、必要なものを見つけ、インタフェース内を移動し、コマンドを入力することを容易にします。

運用コマンドの階層を理解する

あなたがCLIにログインした直後、CLIは運用モードの最上位層に位置しています。

次ページの図1.2は、CLIのツリー構造を運用モードの最上位層から見たところで、例としてshowコマンドを通してカスケード型の階層を示しています。例えば、show configurationの階層にはaccess、chassis、firewall、groups、その他が含まれます。構造的にコマンドをグループ化することで、階層のパスを素早く上下したり、CLI内のどこかにある特定の機能へ素早く移動することが容易になります。

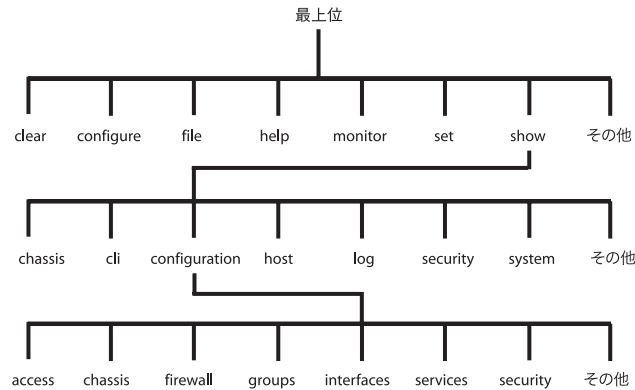


図1.2 運用モードのツリー構造の最上位

注 各モードの最上位層は、UNIX ファイルシステムの最上位 `root(/)` によく似ており、運用モードも設定モードも、その階層構造は UNIX システム、PC、Mac のディレクトリ構造に似ています。運用モードについては第 3 章でさらに詳しく学習します。

設定ステートメントの階層を理解する

設定モードには、関連する設定ステートメントを論理的にグループ化する階層構造があります。この構造で設定のセットアップ、レビュー、修正などの作業が容易になりますが、それは関連するステートメントを見つけて表示することがずっと楽になるからです。図 1.3 は設定モードのツリー構造の一部で、第 2 層には `system` や `interfaces` などのノードがあります。

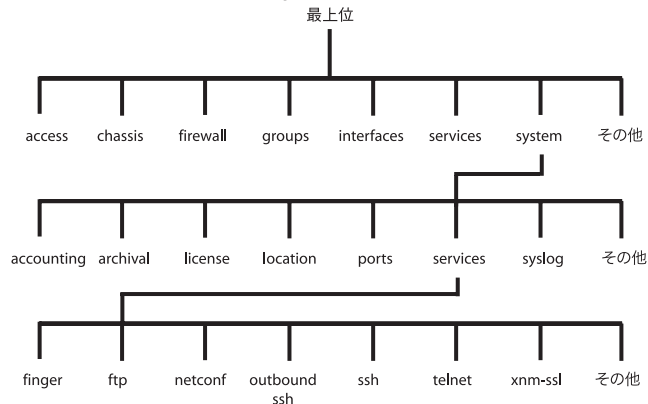


図1.3 設定モードのツリー構造の最上位

- 設定ステートメント階層に含まれるステートメントは2種類に分けられます。
- コンテナステートメント:他のステートメントを含む。つまり、下位の設定階層がある。
 - 単独ステートメント:他のステートメントを含まず、ある階層パスの終端にある。

設定シンタックス

JUNOS CLIは、設定モードの階層構造を固有のシンタックスで表示します。次の例は、CLIの設定リスティングを読む上で知っておく必要がある事項に注目しています。

```
[edit]
system {
  services {
    ftp;
  }
}
```

1. [edit] バナー(見出し)は、リスティングを始める階層を示します。
2. CLIは、設定の階層構造を下位の階層ごとに字下げして表示します。
3. CLIは、一對のブレース({ })を付けてコンテナステートメントであることを示します。上の例では、**system**と**services**がカスケード型のコンテナステートメントです。
4. CLIは、セミコロン(;)を付けて単独ステートメントであることを示します。上の例では、**ftp;**が単独ステートメントです。

注 設定に見られる構成的な構造(コンテナステートメントと単独ステートメントとを構成要素とする構造)は、C言語などの構造化プログラミングに似ていますが、設定ファイルを理解するためにプログラミングを理解する必要はありません。これは設定の見取り図にすぎません(英語の授業の構文解釈を思い出して下さい)。この見取り図がどのように機能するのか理解してしまえば、設定は読みやすく、ナビゲートしやすいものであることが分かるでしょう。

設定コマンドのバナー

あなたは、いま自分が設定の階層構造のどこにいるのかを、常に判断することができます。それは設定コマンドのバナーを参照することで可能です。上の例では `[edit]` バナーとして示されています。もっと深い階層にいる場合、`[edit]` バナーは階層のフルパスを表示します。例えば、`[edit system services]` というバナーが示す階層の位置は、第3層の `services` の中で、それは第2層の `system` の中で、それは第1層 `root` の中にあります。

そこで、FTP サービスについての下記の2つの設定ステートメントは等価であることとなります。1番目のリスティングでは、あなたはステートメントを最上位層 `root` から見えています。そのため、このリスティングでは、`ftp;` ステートメントは `system` と `services` という2つのコンテナステートメントの中に表示されています。

```
[edit]
system {
  services {
    ftp;
  }
}
```

2番目のリスティングでは、あなたは `ftp;` ステートメントを、設定の階層構造の深いところを自分の作業場所にして見えています。具体的には、`system` 内の `services` 内を自分の作業場所としていて、`root` からは2階層深いところからです。階層構造の深いところにいるため、コマンドラインには `ftp;` ステートメントだけが表示されています。

```
[edit system services]
ftp;
```

階層構造の下位にある特定の階層で作業できるという柔軟性は、設定の中の小さな部分だけに注目したい場合に役立ちます。

設定の階層構造内をナビゲートすることについては第4章で学習します。

けれども、とりあえず CLI を使って作業を始めましょう。速く、簡単に、迷うことはありません。あなたが使っているのは JUNOS なのですから。

第2章

基本操作

ログインする	8
運用モードと設定モードを切り替える.....	9
キーボード・ショートカットを使う	9
ヘルプ機能を利用する.....	11
パイプコマンドと <i>more</i> で出力をフィルタリングする ...	15
シェルで作業する.....	20
ログアウトする.....	21

ラボ(試験室)など本番系以外の環境にあるデバイスにアクセスできるユーザーは、この章のサンプルを試しながら CLI を探究してみましょう。あなたのデバイスにコマンドやサンプルを入力できるほか、「試してみよう」のコーナーでは新しいスキルを練習することができます。

CLI にアクセスするには、まずデバイスに接続する必要があり、それからログインします。デバイスへの接続とログインについて分からないことがあれば、そのデバイスの「Quick Start」ガイドを参照するか、後述の URL にアクセスして下さい。ログインして CLI の探究を始める前に、ネットワークがどのようにセットアップされているかを知っておくか、あるいはデバイスへの物理的アクセス手段を確保しておく必要があります。

また、この Day One ブックレットの手順説明では、デバイスの管理用コンソールが設定済みで、その管理用コンソールから、事前に指定されたユーザー名とパスワードでデバイスにログインできることが前提となります。これは、あなたのデバイス上の CLI にアクセスする場合の標準的で推奨される方法です。

もっと詳しく? 工場出荷状態のデバイスにアクセスする場合の情報については、そのデバイスの「Quick Start」ガイドを次の URL で参照して下さい。
www.juniper.net/techpubs/

ログインする

以下の手順で、ネットワークに接続された Telnet 端末(PC など) からデバイスの管理用ポートにアクセスします。

1. コマンドウィンドウ(PC ではコマンドプロンプト) を開きます。
2. 必要なら、JUNOS デバイスに直に接続しているゲートウェイサーバーにログインします。

```
telnet <ゲートウェイサーバー名>  
user: <ユーザー名>  
password: <パスワード>
```

よくあることですが、ルーター、スイッチ、セキュリティデバイスなどがゲートウェイルーターの背後のサブネットにあると、これらのデバイスに対する権限のないアクセスは阻止されます。

3. デバイスにログインします。

```
telnet <ルーター名>  
user: <ユーザー名>  
password: <パスワード>
```

デバイスの IP アドレスが DNS サーバーで管理されている場合、指定されたドメイン名を使うだけでログインできます。それ以外の場合、管理用ポートの固有 IP アドレスを使えばログインできます。

運用モードと設定モードを切り替える

デバイスをモニタしたり設定したりするときは、運用モードと設定モードを相互に切り替える必要があります。設定モードに切り替えると、コマンドプロンプトも切り替わります。運用モードのプロンプトは半角の記号「>」です。設定モードのプロンプトは半角の記号「#」です。

運用モードから設定モードに切り替えるには、`configure` コマンドを発行します。

```
mike@juniper1> configure
```

ショートカット `configure` コマンドを発行するには、ただ「co」と入力すれば十分です。この2文字で始まるコマンドは他にないので、CLIはコマンドを認識して残りの文字を自動補完します。

設定モードを出て運用モードに戻るには、`exit configuration-mode` コマンドか、ずっと短い `exit` コマンドを発行します。

(訳注: `exit` が自動補完されて `exit configuration-mode` になるという意味ではありません)

```
mike@juniper1# exit
```

注 設定を変更した場合、設定モードから出る前に、その変更をコミットして有効にする必要があることを覚えておいて下さい。これについては第4章で説明します。

キーボード・ショートカットを使う

JUNOS CLIは、コマンドラインでの操作でキー入力を節約するための手段を多数提供しています。その中には、入力済み文字列の反復使用や、コマンドの自動補完などがあります。

JUNOS デバイスにログインしているときは、標準的な UNIX キーボード・ショートカットがすべて使えます。このことは、あなたがシェルのどれかを利用している場合でも、CLIを利用している場合でも同じです。ショートカットを使えばキー入力を短縮することができます。ショートカットを自然に使いこなせるようになるには2、3日かかるかもしれません。けれども、いったん指が覚えてしまえば、キー入力の大幅な時間短縮が可能です。

JUNOS CLIは、入力されたコマンドを残らずコマンド履歴に保存します。どのコマンドプロンプトでも、このコマンド履歴の中をカーソル移動キーで上下に移動することができます(VT100ターミナル互換の場合)。以前に入力したコマンドをそのまま再利用したり、必要に応じて一部を変更することができます。入力済み文字列の反復使用は、時間の節約に非常に効果があります。例えば、デバイス上の複数の類似した項目を設定したり、運用コマンドを繰り返し入力する場合で、デバッグ作業などで有効に使えます。

ショートカット 以下のキーボード・ショートカットで時間を節約しましょう。

コマンド履歴で次へ移動	↓キーまたはCtrl+n
コマンド履歴で前へ移動	↑キーまたはCtrl+p
行頭に移動	Ctrl+a
行末に移動	Ctrl+e
左に1文字分移動	Ctrl+b
右に1文字分移動	Ctrl+f
次の単語に移動	Esc+f
前の単語に移動	Esc+b
カーソル上の文字を削除	Ctrl+d
カーソル直後の単語を削除	Esc+d
カーソル直前の単語を削除	Esc+Backspace
カーソルから行末までの文字を削除	Ctrl+k
行を削除	ctrl+u
削除した文字列をカーソル位置に貼り付け	Ctrl+y

コマンド自動補完

CLIにはコマンド自動補完機能があり、どちらのモードでもキー入力をさらに高速化することができます。コマンド自動補完は、コマンド、ファイル名、ユーザー名などの一部を入力すると残りの文字列を自動的に補ってくれる機能であり、入力したい文字列の正確なシンタックスをいちいち思い出す必要がなくなります。コマンド自動補完は、JUNOS 初心者のユーザーには非常に便利な機能で、新しい CLI への移行が容易になります。

スペースキーでほとんどの CLI コマンドが補完されます。Tab キーでは、CLI コマンドだけでなく、ファイル名やユーザー定義の変数（ポリシー名、コミュニティ名、IP アドレスなど）も補完されます。コマンドや引数が一意的に解釈できない場合、スペースキーか Tab キーを押すとその候補 (possible completions) が一覧表示されます。

```
[edit]
mike@juniper1# show i<スペース>
'i' is ambiguous
```

Possible completions:

```
igmp    Show Internet Group Management Protocol
ike      Show Interface Key Exchange Information
interfaces Show Interface Information
ipsec    Show IP Security Information
isis     Show Intermediate System-to Intermediate
```

ショートカット 他のオペレーティングシステムで一般的な省略形(sh や int など) は JUNOS でも使えます。例えば、次のように入力します。

```
mike@juniper1> sh<スペース>ow int<Enter>
```

試してみよう: スペースキーとタブキーを使う

以下のコマンドを、スペースキーを使って補完しながら入力しましょう。

```
sh<スペース>ow ro<スペース>ute
sh<スペース>ow ch<スペース>assis h<スペース>ardware
sh<スペース>ow conf<スペース>iguration
cl<スペース>ear rip s<スペース>tastics
res<スペース>tart ro<スペース>uting g<スペース>racefully
```

ヘルプ機能を利用する

JUNOS CLI にはヘルプ機能を利用する方法がいくつかあり、ユーザーがどうすればよいか分からなくなったとき、あるいは記憶を確かめたいときに、いつでも利用することができます。あらゆるユーザーが、この CLI の総合的で体系的なオンラインヘルプを利用しています。ジュニパーのデバイスを長年使っているベテランもヘルプを活用しています。

コンテキスト依存型ヘルプ

運用モードと設定モードのどの階層でも、コマンドラインに疑問符「?」を付けて質問すると、そのとき使用できるコマンドと簡単な説明が一覧表示されます。コマンドを途中まで入力して疑問符を付ければ、候補となるコマンドの補完の結果がすべて一覧表示されます。このような 2 種類の疑問符の利用法は、JUNOS ではコンテキスト依存型ヘルプと呼ばれます。

```
[edit system]
mike@juniper1# set s?
Possible completions:
  saved-core-context      Save context information for core files
  saved-core-files        Number of saved core files per executable (1..64)
```

```

> services System          services
> static-host-mapping     Static hostname database mapping
> syslog                   System logging facility

```

試してみよう：疑問符でヘルプを利用する

下記の運用モードのコマンドで、補完候補を表示してみましょう。

```

show ?
show chassis ?
show interfaces ?
show system ?
request ?
request support ?
restart ?
ping ?
traceroute ?

```

下記のようにコマンドを途中まで入力して、そこから補完候補を表示してみましょう。

```

s?
show i?
request system s?
restart s?

```

引数としてファイル名が必要なコマンドの場合は、疑問符を付けるとワーキングディレクトリにあるファイルが一覧表示されます。

```

mike@juniper1> request system license add ?
Possible completions:
<filename>      Filename (URL, local, remote, or floppy)
file1           Size: 19701, Last changed: Feb 23 21:56:52
file2           Size: 1835, Last changed: Apr 09 09:51:57
log1            Size: 1215, Last changed: Feb 16 13:07:49
log2            Size: 1135, Last changed: Apr 09 11:05:16
terminal        Use login terminal

```

パスを指定すると、そのディレクトリ内のファイルが表示されます。

```

mike@juniper1> request system license add /cf/ ?
Possible completions:
<[Enter]>       Execute this command
<filename>     Filename (URL, local, remote, or floppy)
/cf/boot/      Last changed: Apr 16 11:08:56
/cf/dev/        Last changed: Apr 08 2004
/cf/etc/        Last changed: Apr 30 08:40:09
/cf/kernel     Size: 32797835, Last changed: Apr 15
/cf/kernel.old Size: 32715591, Last changed: Nov 09

```



```
/cf/opt/          Last changed: Nov 09 02:08:43
/cf/packages/    Last changed: Apr 16 11:08:57
/cf/root/        Last changed: Apr 16 11:08:56
/cf/sbin/        Last changed: Apr 16 11:08:56
/cf/usr/         Last changed: Nov 09 02:11:23
/cf/var/         Last changed: Nov 09 02:11:23
```

オンボードの技術資料

コンテキスト依存ヘルプよりも詳しい情報が必要な場合は、**help** コマンドで、あなたのデバイスに搭載されている JUNOS ソフトウェア技術資料を参照して下さい。ジュニパーは技術資料をデバイスに搭載して出荷し、アップグレードのビルドの一部としても提供しています。

ヘルプファイルは 5 つの大分類に分かれています。ヘルプファイルは、運用モードからでも設定モードからでもアクセスできます。

- **help apropos:** ステートメントやコマンド名に含まれる文字列についてのヘルプを表示
- **help reference:** 設定ステートメントのサマリー情報を表示して、設定シンタックスに関する支援を提供
- **help syslog:** 指定したシステムログイベントの情報を表示
- **help tip:** CLI の使用上のヒントをランダムに表示
- **help topic:** 設定ステートメントの使用法のガイドラインを表示

ヘルプを要求する場合は、上記のコマンドに続けて、情報を必要としている文字列やトピックを入力します。

help apropos コマンド

help apropos コマンドは、コマンドの一部は覚えているけれども完全な形のステートメントを覚えていない場合に便利です。このコマンドは、一致するステートメントやコマンド名を探し、それに対応するヘルプ文字列と共に表示します。

```
[edit]
mike@junper1# help apropos host-name
set system host-name <host-name>
    Hostname for this router
set system static-host-mapping <host-name>
    Fully qualified name of system
set system services dhcp static-binding <mac-address> host-name <host-name>
```

```

    Hostname for this client
set system syslog host
    Host to be notified
set interfaces <interface_name> services-options syslog host <host-name>
    Name of host to notify
set accounting-options routing-engine-profile <profile-name> fields host-name
    Hostname for this router
set services l2tp tunnel-group <name> syslog host <host-name>
    Name of host to notify
set services service-set <service-set-name> syslog host <host-name>
    Name of host to notify

```

文字列の中にスペースが含まれるときは、文字列を二重引用符 (" ") で囲みます。

help topic コマンド

設定ステートメントを指定して、その使用法のガイドラインを知りたいときは、この `help topic` コマンドを使います。

```

mike@juniper1> help topic interfaces address
Configuring the Interface Address

```

You assign an address to an interface by specifying the address when configuring the protocol family. For the `inet` family, you configure the interface's IP address. For the `iso` family, you configure one or more addresses for the loopback interface. For the `ccc`, `tcc`, `mpls`, `tnp`, and `vpls` families, you never configure an address.

訳注:上記のヘルプ記述は以下のような意味です。「プロトコルファミリーを設定する場合、あるアドレスを指定してインタフェースのアドレスとして割り当てます。`inet`ファミリーでは、インタフェースのIPアドレスを設定します。`iso`ファミリーでは、ループバック・インタフェース用に1個以上のアドレスを設定します。`ccc`、`tcc`、`mpls`、`tnp`、`vpls`ファミリーでは、アドレスは設定しないで下さい。」

help reference コマンド

あるコマンドが、何をするもので、いつ使えばよいか分かったら、`help reference` コマンドで、実際のシンタックスと指定できるオプションを確認しましょう。上記と同じコマンドを例とします。

```

mike@juniper1> help reference interfaces address
address
    Syntax

address address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
    destination address;
    destination-profile name;
    eui-64;
    multipoint-destination address dlci dlci-identifier;

```

...
Hierarchy Level]

```
[edit interfaces interface-name unit logical-unit-number family family],  
[edit logical-routers logical-router-name interfaces interface-name unit  
logical-unit-number family family]
```

Description

Configure the interface address.

<...>

注 help reference コマンドは、UNIX のオンラインマニュアル (man page) や、他のオペレーティングシステムのマニュアルコマンドに似ています。

シンタックスヘルプ

JUNOS ソフトウェアは、あなたが設定ステートメントの最後で Enter キーを押すまで待たずに、シンタックスを単語ごとにチェックします。あなたがコマンドラインの中に単語を 1 つ入力してスペースキーを押すごとに、CLI はその単語がコマンドの一部として正当かどうか、使い方が正しいかどうかを確認します。間違いを見つけると、CLI は修正を求めます。

さらに JUNOS ソフトウェアは、特定の階層で必要とされるステートメントが省略されていないかどうかもチェックします。このチェックは、あなたがその階層から移動しようとするとき、あるいは設定モードで show コマンドを発行するときに行われます。

```
[edit]  
mike@juniper1# show  
protocols {  
  pim {  
    interface so-0/0/0 {  
      priority 4;  
      version 2;  
      # Warning: missing mandatory  
statement(s): 'mode'  
    }  
  }  
}
```

訳注:この例では、interfaceステートメントの下位に、必須のmodeステートメントがないことが警告されています。

パイプコマンドと more で出力をフィルタリングする

CLI が出力をどのように表示するかをパイプ文字 (|) と more プロンプトで変更することができます。

パイプ文字

パイプ文字 (|) は、運用モードと設定モードの両方で、出力のフィルタリングに使えます。パイプは、指定した情報をコマンドの 1 ステップで表示できるようにする機能です。あるコマンドの出力を別のコマンドの入力として送るか、あるいはリダイレクトしてファイルに出力します。パイプ文字の左側のコマンドの出力が、右側のコマンドやファイルの入力となります。

CLI に質問すれば、コマンドをパイプで処理する正当な方法を知ることができます。下記のリスティングは運用モードでの使用例です。

```
mike@juniper1> show route | ?
```

Possible completions:

```
count      Count occurrences
display    Show additional kinds of information
except     Show only text that does not match a pattern
find       Search for first occurrence of pattern
hold       Hold text without exiting the --More-- prompt
last       Display end of output only
match      Show only text that matches a pattern
no-more    Don't paginate output
request    Make system-level requests
resolve    Resolve IP addresses
save       Save output text to file
trim       Trim specified number of columns from start of line
```

訳注:この例ではshow routeの出力をパイプで何に渡せるかを表示しています。

パイプを使う

以下は設定済みのデバイスによるサンプルであり、コマンドを微調整していく上でパイプがどれだけ役に立つかを示しています。

コマンドの出力をフィルタリングしてファイルに送る

第3章の「fileコマンドを使う」のセクションで、作成したファイルへのアクセスの仕方を学習します。

運用モードの `request support information` コマンドの出力を保存するファイルを新規作成するため、出力をパイプで指定のファイル名にリダイレクトします。

```
mike@juniper1> request support information | save <ファイル名>
Wrote 1143 lines of output to 'ファイル名'
```

情報を追加して表示、あるいは情報をホールドする

CLI に、情報を追加してリスティングをしたり、情報をホールドすることを要求することができます。

出力をXMLで表示

設定情報やステート情報を他のシステムとやり取りする場合は、出力をXML形式で渡すと便利です。XML形式の出力は標準RPC(リモートプロシージャコール)形式でフォーマットされます。

- | **count**: 出力の行数を渡す

```
mike@juniper1> show interfaces terse | count
Count: 22 lines
```
- | **display detail**: 設定の内容に関する追加の情報を渡す(設定モードでのみ使用可能)
- | **display xml**: 出力をXML形式で渡す

```
mike@juniper1> show cli directory | display xml
<rpc-reply xmlns:junos=?http://xml.juniper.net/
junos/9.410/junos>
  <cli>
    <working-directory>/var/home/username</working-
directory>
  </cli>
  <cli>
    <banner>master:0</banner>
  </cli>
</rpc-reply>
```
- | **hold**: クリアされるまで出力をバッファに保持

コマンド出力を条件で絞り込む

パイプ文字(|)のもっとも一般的な使用法は、条件を指定して出力を絞り込むことです。

- | **match**: あなたが、まさに表示したい情報を指定

```
mike@juniper1 > show configuration | match at
at-2/1/0 {
at-2/1/1 {
at-2/2/0 {
at-5/2/0 {
at-5/3/0 {
```

注 match は UNIX の grep コマンドと同等です。

- | **except**: 出力中の指定した文字列は除外して渡す

```
mike@juniper1> show system users | except root
8:28PM up 1 day, 13:59, 2 users, load averages:
  0.01, 0.01, 0.00
USER  TTY FROM          LOGIN@  IDLE WHAT
sheep p0 baa.juniper.net 7:25PM - cli
```

訳注:この例では2つのユーザー名のうち、rootは除外されてsheepだけが表示されています。

- | **find**: 指定の文字列が見つかった位置を先頭として出力を渡す

```
mike@juniper1> show ethernet-switching interfaces detail | find "Index: 80"
Interface: ge-0/0/16.0 Index: 80
```

```
State: down
VLANs:
  default untagged blocked - blocked by STP
```

```
Interface: ge-0/0/17.0 Index: 81
State: down
VLANs:
  default untagged blocked - blocked by STP
```

■ | last: リスティングの最後のページだけを表示

注 `find` や `match` で検索文字列にスペース、演算子、ワイルドカード文字が含まれる場合は、文字列を引用符で囲みます。

多重パイプ

JUNOS ソフトウェアは、多重パイプを論理積 (AND) とみなし、すべてのパイプを通過した出力だけを表示します。多重パイプに入力するコマンドは、同じコマンドでも違うコマンドでもよく、多数をつなげることができます。例えば、アクティブな設定の中で高速 Ethernet インタフェースが何個設定されているかを数えるには次のようにします。

```
mike@juniper1> show interfaces terse | match fe- | count
Count: 12 lines
```

次のサンプルでは、同じパイプコマンドを 1 行に並べて、サブネットマスク「/32」で、文字列「10.0」を含む route (ネットワーク経路) をすべて表示します。

```
mike@juniper1> show route | match /32 | match 10.0
10.0.15.2/32    *[Local/0] 03:18:28
10.0.16.1/32    *[Local/0] 03:20:49
10.0.0.4/32     *[Local/0] 08:54:55
192.168.10.0/32 *[Local/0] 08:57:26
```

More プロンプト

CLI は出力を自動的にページ建てします。各ユーザーアカウントごとの 1 ページの最大行数は CLI の設定で決まり、既定値は 24 行です。デバイスが改ページ位置で表示を停止すると、CLI は `more` プロンプトを表示し、同時に表示済みの出力が表示されるべき全体の何 % にあたるのかを数字で表示します。 `more` プロンプトが表示されているときはいつでも、`h` キーを押して表示オプションをリスト表示することができます。この中には、出力の中を前方や後方に移動すること、検索、保存などが含まれます。

```
mike@juniper1> show ethernet-switching interfaces detail
```

```
Interface: ge-0/0/0.0 Index: 64
```

```
State: down
```

```
VLANs:
```

```
default          untagged   blocked - blocked by STP
```

```
*// 途中のデータは省略 //*
```

```
Interface: ge-0/0/12.0 Index: 76
```

```
State: down
```

```
VLANs:
```

```
default          untagged   blocked - blocked by STP
```

```
---<more>--- h
```

```
---(Help for CLI automore)---
```

```
Clear all match and except strings(matchとexceptの文字列をすべてクリア):      c or C
```

```
Display all line matching a regexpz(正規表現に合致する行をすべて表示):
```

```
      m or M <string>
```

```
Display all lines except those matching a regexp(正規表現に合致しない行をすべて表示):
```

```
      e or E <string>
```

```
Display this help text(このhelp記述を表示):
```

```
      h
```

```
Don't hold in automore at bottom of output(最終ページではmoreプロンプトを表示しない):
```

```
      N
```

```
Hold in automore at bottom of output(最終ページでもmoreプロンプトを表示):      H
```

```
Move down half display(表示を1/2ページ進める):      TAB, d, or ^D
```

```
Move down one line(表示を1行進める):
```

```
      Enter, j, ^N, ^X, ^Z, or Down-Arrow
```

```
Move down one page(表示を1ページ進める):
```

```
      Space, f, ^F, or Right-Arrow
```

```
Move to bottom of output(表示を最終ページまで進める):
```

```
      G, ^E, or End
```

```
Move to top of output(表示を先頭ページに戻す):
```

```
      g, ^A, or Home
```

```
Move up half display(表示を1/2ページ戻す):
```

```
      u or ^U
```

```
Move up one line(表示を1行戻す):
```

```
      k, Delete, Backspace, ^P, or Up-Arrow
```

```
Move up one page(表示を1ページ戻す):
```

```
      b, ^B, or Left-Arrow
```

```
Quit automore(moreプロンプトを終了):
```

```
      q, Q, ^K
```

```
Redraw display(表示をリフレッシュ):
```

```
      ^L or ^R
```

```
Repeat a keystroke command 1 to 9 times(キー入力コマンドを1~9回繰り返す):
```

```
      Meta-1..9
```

```
Repeat last search(最新の検索を再度実行):
```

```
      n
```

```
Save output to a file(出力をファイルに保存):
```

```
      s or S <filename/url>
```

```
Search backwards thru the output(出力内を逆方向に検索):
```

```
      ?<string>
```

```
Search forwards thru the output(出力内を順方向に検索):
```

```
      /<string>
```

```
---(End of Help)---
```

ヒント `set cli screen-length` コマンドで画面に表示する最大行数を修正できます。出力を一括表示するには、コマンドにパイプ「| no-more」を追加するという方法もあります。

シェルで作業する

シェルとは一般に、UNIX や UNIX 類似のオペレーティングシステムで使われる CLI を指します。この種のインタフェースは、高水準のコマンドをデバイスが理解できる低水準の命令に変換します。JUNOS のデバイスにログインすると、2 つのシェルのどちらかの中に置かれることになります。つまり、C シェルか JUNOS CLI の運用モードのどちらかです。

利用できるシェルは以下の通りです。

シェル	コマンドプロンプト
C シェル	%
Bourne シェル	\$
CLI	mike@juniper1> (運用モード) mike@juniper1# (設定モード)

適切なコマンドを発行して、あるシェルから別のシェルへ移動することができます。このコマンドは新しくシェルのインスタンスを生成し、あなたをそのインスタンスの中に置きます。例えば、あなたが CLI の運用モードにいるとき、新しく C シェルのインタフェースを生成することができ、それには `start shell` コマンドを発行します。

警告! デバイスからログアウトするときは、セッション中に入ったすべてのシェルから出たことを確認して下さい。

CLI にログインする

CLI インタフェースにログインするには、いずれかのシェルのプロンプトで `cli` コマンドを発行します。

```
% cli
```

CLI は常に運用モードで開きます。

C シェルにログインする

CLI の運用モードから C シェルインタフェースにログインするには、`start shell` コマンドを発行します。

```
mike@juniper1> start shell
```


CLI の設定モードから C シェルインタフェースにログインするには、`run start shell` コマンドを発行します。

```
mike@juniper1# run start shell
```

ショートカット `run` コマンドにより、CLI の設定モードで運用モードのコマンドを発行することができます。設定モードの中にも、実行したい運用モードのコマンドの前に `run` というキーワードを付けるだけです。

Bourne シェルインタフェースから C シェルインタフェースにログインするには、`csch` コマンドを発行します。

```
$ csch
```

Bourne シェルにログインする

CLI の運用モードから Bourne シェルインタフェースにログインするには、`start shell sh` コマンドを発行します。

```
mike@juniper1> start shell sh
```

CLI の設定モードから Bourne シェルインタフェースにログインするには、`run start shell sh` コマンドを発行します。

```
mike@juniper1# run start shell sh
```

C シェルインタフェースから Bourne シェルインタフェースにログインするには、`sh` コマンドを発行します。

```
% sh
```

ログアウトする

ログアウトするときは、あなたが開いたシェルからすべてログアウトした上で、デバイスから完全にログアウトする必要があります。つまり、デバイスにログインして CLI の運用モードに置かれた後、CLI の設定モードを開いた場合、両方のシェルから出る必要があります。

休憩時間になったら、あなたが開いたシェルのそれぞれからログアウトして、デバイスから完全にログアウトします。デバイスから完全にログアウトすると、次のメッセージが表示されます。

```
"Connection closed by foreign host."
```

```
mike@juniper1> exit  
logout
```

```
Connection closed by foreign host.  
$
```

設定モードにいるときにログアウトしたい場合は、設定セッションを出て運用モードに入ります。

```
[edit protocols ospf]  
mike@juniper1# exit configuration-mode  
Exiting configuration mode
```

```
mike@juniper1> exit  
logout  
Connection closed by foreign host.  
$
```

おすすめ ログインし続ける理由がない場合や、たとえ数分でも端末から離れる場合は、ログアウトしてデバイスのセキュリティを守りましょう。そうすれば誰かが間違っ（あるいは故意に）、あなたのワークステーションの席に着き、デバイスにアクセスするのを防げます。

第3章

運用モードを理解する

運用モードを詳しく見る.....	24
デバイスステータスを表示する.....	25
運用管理の基本機能.....	27
ファイル操作コマンドを使用する.....	28
オペレーティングシステム・ソフトウェアを管理する ...	29

JUNOS CLI の運用モードは、デバイスをモニタし、管理し、メンテナンスするためのコマンドを提供します。あなたはデバイスステータスを確認し、診断を行い、その他の運用上の作業をこなすことができ、デバイスを稼働させているソフトウェアの管理も可能です。

運用モードの主な機能は以下の通りです。

- デバイスのモニタリングとトラブルシューティング
- 他のネットワークシステムとの接続
- ソフトウェアプロセスのリスタート
- 設定モードへの遷移
- 設定情報の表示
- CLI 環境の制御
- システムレベルの操作（デバイスの停止とリブート、JUNOS ソフトウェア・イメージのロード、など）

JUNOS ソフトウェアには機能豊富なオンボードのデータ収集能力があり、重大な運用ステータス、統計データ、その他の情報を収集します。このツールはトラブルの事前通知を行い、イベント発生時の問題解決を速めます。

あなたが行う設定セットアップの一環として、トラッキングしたいイベントのタイプ、イベントの深刻度、データ保存先のファイル、その他のオプションを指定することができます。ジュニア機器には重大な運用データを収集し、保存するだけの十分な処理能力があります。収集されるデータには、SNMP 管理、システムロギング、トレースオプションなどがあり、平常時の機器の運用状況と、どこで、いつ、なぜ変化が起こるのかを理解するのに役に立ちます。

もっと詳しく？ あなたのデバイスの基本的なモニタリング機能の設定について、近刊予定の JUNOS ソフトウェア基本シリーズのブックレットで、さらに深く学びましょう。次の URL で入手可能になるので、ダウンロードして下さい。

www.juniper.net/dayone

運用モードを詳しく見る

運用モードを階層の最上位から探究しましょう。以下はごく普通に使用されるコマンドのリステイング（一部省略）です。

```
mike@juniper1> ?
```

```
Possible completions:
```

```
clear          Clear information in the system (システム内の情報をクリア)
```

configure	Manipulate software configuration information (ソフトウェア設定情報を操作)
file	Perform file operations (ファイル操作を実行)
help	Provide help information (ヘルプ情報を提供)
monitor	Show real-time debugging information (リアルタイムのデバッグ情報を表示)
ping	Ping remote target (リモートのターゲットへのping)
quit	Exit the management session (管理セッションを終了)
request	Make system-level requests (システムレベルの要求を発行)
restart	Restart software process (ソフトウェアプロセスをリスタート)
set	Set CLI properties, date/time, craft interface message (CLIのプロパティと日付/時刻を設定、インタフェースメッセージを編集)
show	Show system information (システム情報を表示)
ssh	Start secure shell on another host (別のホスト上でセキュアシェルを開始)
start	Start shell (シェルを開始)
telnet	Telnet to another host (別のホストにtelnetで接続)
test	Perform diagnostic debugging (診断機能によるデバッグを実行)
traceroute	Trace route to remote host (リモートホストへの経路をトレース)

<以下省略>

デバイスステータスを表示する

運用モードで提供される **show** コマンドのグループは多様で、デバイス上のほとんどあらゆることについて、ステータスと統計データ表示することができます。

```
mike@juniper1> show?
```

Possible completions:

accounting	Show accounting profiles and records
aps	Show Automatic Protection Switching information
arp	Show system Address Resolution Protocol table entries
as-path	Show table of known autonomous system paths
bfd	Show Bidirectional Forwarding Detection information
bgp	Show Border Gateway Protocol information
chassis	Show chassis information
class-of-service	Show class-of-service (CoS) information
cli	Show command-line interface settings
configuration	Show current configuration
connections	Show circuit cross-connect connections
dialer	Show dialer information
dls	Show DLSw information
dvmp	Show Distance Vector Multicast Routing Protocol information
dynamic-tunnels	Show dynamic tunnel information information
esis	Show end system-to-intermediate system information
firewall	Show firewall information
helper	Show port-forwarding helper information
host	Show hostname information from domain name server
igmp	Show Internet Group Management Protocol information
ike	Show Internet Key Exchange information
interfaces	Show interface information
ipsec	Show IP Security information

ipv6	Show IP version 6 information
isdn	Show Integrated Services Digital Network information
isis	Show Intermediate System-to-Intermediate System info...
l2circuit	Show Layer 2 circuit information
l2vpn	Show Layer 2 VPN information
ldp	Show Label Distribution Protocol information
llc2	Show LLC2 protocol related information
log	Show contents of log file
mld	Show multicast listener discovery information
mpls	Show Multiprotocol Label Switching information
msdp	Show Multicast Source Discovery Protocol information
multicast	Show multicast information
ntp	Show Network Time Protocol information
ospf	Show Open Shortest Path First information
ospf3	Show Open Shortest Path First version 3 information
pfe	Show Packet Forwarding Engine information
pgm	Show Pragmatic Generalized Multicast information
pim	Show Protocol Independent Multicast information
policer	Show interface policer counters and information
policy	Show policy information
pppoe	Show PPP over Ethernet information
rip	Show Routing Information Protocol information
ripng	Show Routing Information Protocol for IPv6 information
route	Show routing table information
rsvp	Show Resource Reservation Protocol information
sap	Show Session Announcement Protocol information security
services	Show services information
snmp	Show Simple Network Management Protocol information
system	Show system information
task	Show routing protocol per-task information
ted	Show Traffic Engineering Database information
version	Show software process revision levels
vrrp	Show Virtual Router Redundancy Protocol information

ヒント Cisco IOS の利用経験がある読者にとって JUNOS の基本的な違いは、キーワード「IP」を使用しないことです。そのため、既にご存知の show コマンドの多くは「IP」を省略するだけで使えます。例えば、IOS コマンドの show ip route は、JUNOS では単純に show route になります。

show コマンドには他にも引数があって、出力を加工することができます。例えば、fe-1/1/1 Fast Ethernet インタフェースに対する show interfaces コマンドでは、以下の引数を使用可能です。

```
mike@juniper1> show interfaces fe-1/1/1 ?
```

```
Possible completions:
```

```
<[Enter]>      Execute this command
brief          Display brief output
descriptions   Display interface description strings
```

```

detail          Display detailed output
extensive       Display extensive output
media           Display media information
snmp-index      SNMP index of interface
statistics      Display statistics and detailed output
terse           Display terse output
<. . .>

```

これらのオプションを追加すれば、必要な情報をリスティング出力するように調整することができます。2つのオプション **brief** と **terse** を上の例の **show** コマンドに追加したときの出力を比較してみましょう。

```
mike@juniper1> show interfaces fe-1/1/1 brief
```

```

Physical interface: fe-1/1/1 Enabled, Physic link is Down
Link-level type:Ethernet, MTU: 1514, Spped: 100mbps, Loopback:
                    Disabled, Source filtering: Disabled
Flow control : Enabled
Device flags  : Present Running Down
Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
Link flags    : None

```

```
mike@juniper1> show interfaces fe-1/1/1 terse
```

```

Interface   Admin Link Proto Local Remote
fe-1/1/1    up up
at-1/3/0.0 up up  inet 1.0.0.1 --> 1.0.0.2
iso

```

ヒント `clear` コマンドにより、デバイスの統計データをゼロにリセットすることができます。

運用管理の基本機能

JUNOS ソフトウェアは標準的なネットワーク・ユーティリティと管理用のリモートアクセスをサポートしています。これらの基本的なコマンドのいくつかは UNIX や他のオペレーティングシステムで見覚えがあるでしょう。

- **ping:** この標準的な IP コマンドは他のデバイス、インタフェースカード、あるいはノードがネットワーク上で到達可能かどうかをテストします。
- **traceroute:** このネットワーク・ユーティリティは、あなたのデバイスから IP ネットワーク上の宛先までパケットが通るパスを報告します。
- **ssh:** この標準的な UNIX のセキュアシェル・プログラムは、ネットワーク上の他のデバイスやホスト上でユーザーシェルを開始します。
- **telnet:** この管理プロトコルは、ネットワーク上の他のデバイスやホストとの間でターミナル接続を確立します。

ファイル操作コマンドを使用する

ファイル操作コマンドを使えば、ファイルの内容を表示したり、あなたのデバイス上のある場所から他の場所へ、あなたのデバイスからリモートシステム（サーバーなど）へ、あるいはリモートシステムからあなたのデバイスへ、ファイルをコピーすることができます。デバイス上の設定ファイルをセーブ/ロードすることは、以下の場合に効果的です。

- 設定情報をアーカイブに保存したり、バックアップしたりする
- 複数のデバイスで設定情報を共有する
- 設定ファイルを部分的にセーブ/ロードする。対象となるのはネットワーク内の多数のデバイスに共通している可能性のある部分（ルートフィルタなど）

ファイルの内容を表示するには、`file show` コマンドを使います。

```
mike@juniper1> file show <ファイル名>
```

file copy コマンド

`file copy` コマンドを使用すれば手動でファイルをアーカイブに保存できます。`file copy` コマンドのシンタックスは標準的な UNIX の `cp` コマンドと同じです。

```
file copy /target-directory/target-filename<スペース>  
/destinationdirectory/destination-filename
```

訳注:1番目の引数がコピー元、2番目の引数がコピー先です。

例えば、現在アクティブな設定ファイル (`/config/juniper.conf.gz`) を、`backup.gz` としてデバイスの `/var/home/user` ディレクトリにコピーする場合は次のようにします。

```
mike@juniper1> file copy /config/juniper.conf.gz<スペース>  
/var/home/user/backup.gz
```

おすすめ 本番用の設定ファイルについてはレスキュー（緊急用）設定を作成しましょう。万一、アクティブな設定ファイルが壊れた場合、デバイスは自動的に、`/config` ディレクトリにある `rescue.gz` という名前のファイルをアクティブな設定としてロードします。

おすすめ 設定ファイルを新しい場所にコピーした後、常にそのファイルをリネームするようにします。こうすれば、後になって同じファイルを更新したものをコピーするとき、誤って上書きすることはありません。

同じコマンドで、サーバーからデバイスのホームディレクトリに設定ファイルを戻すことができます。

```
mike@juniper1> file copy /config/juniper.conf.gz<スペース>  
/var/home/user/juniper.conf.gz-20090123
```


file list コマンド

`file list` コマンドを使用すれば、ファイルがあなたのホームディレクトリに作成されたことを確認できます。

```
mike@juniper1> file list

/var/home/user/:
.ssh/
juniper.conf.gz-20090123
```

注 第4章では、デバイスのアクティブな(本番用)設定ファイルとしてファイルをロードする手順を学習します。

オペレーティングシステム・ソフトウェアを管理する

運用モードには、オペレーティングシステム・ソフトウェアを管理するためのコマンドがあり、デバイスのアップグレードとリポート、さらに個々のプロセスのリスタートやリセットができます。JUNOSソフトウェアはモジュラー型のオペレーティングシステムで、個々のプロセスは独立して保護されたメモリー空間で実行されます。そこで、これらのプロセス(デーモンと呼ばれます)は、個別に管理することができます。

restart コマンド

JUNOSソフトウェアのプロセスの大半は運用モードからリスタートできます(少数のデーモンはシェルに任せる必要があります)。オペレーティングシステムのデーモンを個別に停止し、そのあとリスタートする必要がある場合は、`restart` コマンドを使用します。

警告! 個々のプロセスは完全に独立していますが、`restart`コマンドを使用するときは十分に注意してください。SNMPプロセスでの`restart`はSNMPを停止させるだけですが、`routing`デーモンの`restart`はあなたのネットワークに重要な影響を及ぼす恐れがあります。

ヒント 特定のルーティングプロトコル(OSPFなど)をリスタートするには、設定モードでそれを非アクティブ化してから再アクティブ化するという方法があります。トラブルが1つのプロトコルだけに関わる場合、これがうまい方法です。JUNOSの`routing`デーモン全体をリスタートすれば、全てのルーティングプロトコルに影響を及ぼすことになります。

request コマンド

`request` コマンドはシステム全体に関わる機能、例えばデバイスのリブート、アップグレード、シャットダウンなどを実行します。このコマンドグループは、デバイス全体をリブートすることなく、個々のコンポーネントをオンラインにしたり、オフラインにしたり、リスタートしたりすることもできます。

```
mike@juniper1> request chassis fpc slot 0 restart
```

```
Restart initiated, use "show chassis fpc" to verify
```

```
user@host> show chassis fpc
```

Slot State	Temp	CPU Utilization (%)		Memory Utilization (%)		
	(C)	Total	Interrupt	DRAM (MB)	Heap	Buffer
0 Starting	32	0	0	0	0	0
1 Online	30	0	0	8	11	14
2 Empty						
3 Empty						

もっと詳しく? オンライン技術資料「*JUNOS Software Installation and Upgrade*」には、あなたのデバイスのソフトウェアバージョンのアップグレードに関する詳細情報が掲載されています。最新のパッケージは、ソフトウェア・ダウンロード・ページ (www.juniper.net/support) からダウンロードできます。新しいソフトウェアをダウンロードするには、現在有効のサービス契約とログインアカウントが必要です。

もっと詳しく? 運用モードのコマンドについてもっと深く学びたい方は、オンラインで「*JUNOS Software CLI User Guide*」と「*JUNOS System Basics and Services Command Reference*」を参照して下さい。URLは次の通りです。
www.juniper.net/techpubs/

第4章

設定モードを探索する

設定プロセスの概要	32
設定モードに入る.....	33
設定モードの基礎を理解する	35
設定を編集する	40
設定候補をコミットする.....	42
日常作業を自動化する.....	47
設定をロールバックする	48
事前にシステム変更を準備する	49
設定を簡単にするテクニック.....	52

設定モードとは、その名前の通り、デバイスの「設定」(*configuration*)を定義するモードです。設定モードには、管理用コンソールのネットワーク設定項目を設定して使えるようにすること、ユーザーアカウントを登録してデバイスにアクセスできるようにすること、セキュリティ対策を指定してデバイスとネットワークを保護すること、スイッチングとルーティングのプロトコルをセットアップすること、などが含まれます。各ステートメントは、それぞれデバイスの別々の機能を設定し、ネットワーク内でのデバイスの特定のプロパティを指定します。

設定プロセスの概要

JUNOS CLI は、設定をプロセス(一連の手続き)とみなすよう、慎重に設計されています。そこで、セーフガードが導入され、新しい設定はセットアップ後にチェックしてから運用を開始できるようになっています。例えば、JUNOS は設定「候補」に加えられたすべての変更を捕捉します。この設定候補は、完成した後でコミットされることで、初めてアクティブな設定ファイルになります。

この方式は、設定変更を行単位で入力して即座にアクティブ化する他のシステムとは、まさに対照的です。他のシステムで行単位の変更をした経験がある方は、自分が干渉のリスクを引き起こしてしまう(例えば、あるインタフェースのファイアウォールを削除してしまうというような)ことがある、ということに気づかれたのではないのでしょうか。おそらく、ある 1 行の変更を入力することで、自分が意図しない、予想外の結果を生じさせ、簡単に元に戻せなかったという経験をお持ちでしょう。

JUNOS CLI は、このような設定に関する厄介な問題からあなたを解放します。ジュニパーのエンジニアはその目的を目指して、早くからお客様の協力を得ながら、多段階の設定プロセスを設計しました。このプロセスは、デバイス設定における予期しないミスや、その他のよくある難問によって引き起こされるトラブルを回避するさまざまな手段を提供します。

図 4.1 は、JUNOS ソフトウェアによって稼働するデバイスを設定する 3 段階の基本的なステップを概説しています。図中に数字で示されているステップ 1 ~ 3 を以下に説明します。

設定候補はどこにある?

設定はファイルだと考えがちですが、実は設定候補に対応するファイルは存在しません。設定候補はシステムメモリーにあります。

1. 設定候補に変更を加える

設定候補はアクティブな設定のコピーです。あなたは CLI を介して設定候補に対する設定変更を入力できます。具体的にはテキストファイルをカット & ペーストしたり、ロードしたり、更新済の設定変更とマージしたり、あるいは CLI から変更を直に手入力することもできます。設定候補にすべての変更を加え終えたら、作業結果を確認することができます。設定候補をアクティブな(本番用)ファイルと比較するのも 1 つの方法です。

2. 変更をコミットする

設定候補を移動してアクティブな設定にするには、`commit` コマンドまたは `commit confirmed` コマンドを入力します。ソフトウェアは最終的に変更を反映する前に設定候補内の特定のステートメントをチェックし、その他のコンテキスト検証を行います。デバイス上にロード済みのコミットスクリプトが存在する場合は、そのスクリプトも設定候補内のエラーをチェックし、おそらくはエラーを修正します。

3. 設定候補がアクティブになる

設定候補は、すべての検証チェックに合格するとアクティブになります。設定候補がアクティブな設定ファイルになり、`/config/juniper.conf.gz` として保存されます。デバイスは、変更直前の `juniper.conf.gz` ファイルをリネームして、`juniper.conf.1.gz` にします。

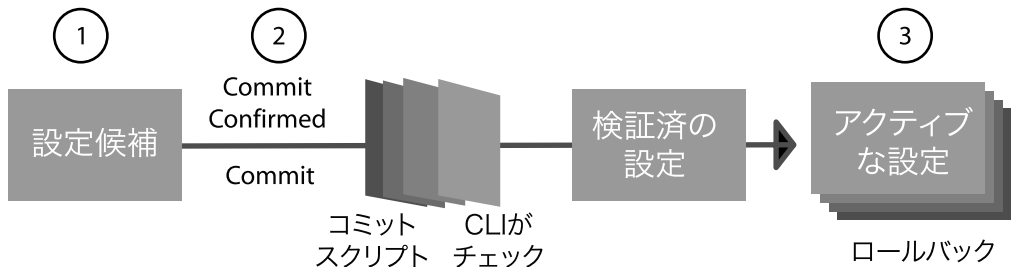


図4.1 JUNOS CLI設定プロセスの概要

注 JUNOS デバイスは、過去にアクティブだった設定を最大 49 個まで保存します。`rollback [1 - 49]` コマンドを発行することによって、これらのバックアップ設定のうちの任意の 1 つにロールバックできます。これについては、この章でこのあと学習します。

設定モードに入る

設定の変更を行えるユーザーアカウントが 1 つだけではないデバイス環境では、誰が変更作業中か、いつ変更を行うのかを管理する柔軟性が不可欠です。そこで、JUNOS ソフトウェアは、設定モードへの入り方として以下の 3 種類のオプションを提供します。

- **標準 (standard)**: 設定候補を同時に編集できるユーザー数に制限はなく、ひとりのユーザーが加えた変更は視覚的に共有され、すべてのユーザーが変更を確認できる。
- **排他 (exclusive)**: 排他ユーザーが排他状態を終了するまで、他のすべてのユーザーは設定モードから締め出される。

- **個人 (private):** 個人別の設定候補を提供。デバイスが持つ個人別の設定候補のコピーは、そのユーザーが加えた変更だけを保持。

configure コマンド

標準の設定モードに入るには、`configure` コマンドを発行します。

```
mike@juniper1> configure  
Entering configuration mode
```

警告! 標準の設定モードを出るときは、あなたがセッション中に加えたコミットされていない変更は、すべて設定候補の中に残ります。この変更を無効にするためには、明示的に削除するか、`rollback 0` コマンドを発行して、アクティブな設定を設定候補としてリロードする必要があります。ユーザーが標準モードでログインおよびログアウトするときは、以下の警告メッセージが表示されます。

```
mike@juniper1> configure  
Entering configuration mode  
The configuration has been changed but not committed
```

```
mike@juniper1# exit  
The configuration has been changed but not committed  
Exit with uncommitted changes? [yes,no]
```

configure exclusive コマンド

設定候補をロックして他のユーザーを締め出すには、`configure` コマンドに `exclusive` スイッチを追加します。あなたが排他モードのセッションを終了すると、デバイスは、コミットされなかった設定変更をすべて破棄します。

```
mike@juniper1> configure exclusive  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```

configure private コマンド

`configure` コマンドに `private` スイッチを追加すれば、個人モードで自分専用の設定候補を作成することができます。

```
mike@juniper1> configure private  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```

個人モードのユーザーが変更をコミットするとき、JUNOS ソフトウェアはそのユーザーが設定候補に加えた変更だけをアクティブな(本番用)設定に統合します。JUNOS ソフトウェアは、他のユーザーが設定候補に加えた変更は何も実装しません。つまり、何人かのユーザーが `configure private` コマンドを使用して、アクティブな設定に対する互いに競合しない変更を、同時に加えることが可能です。個人モードのユーザーが `rollback 0` コマンドを発行した場合、デバイスはそのユーザーの設定候補に加えられた変更だけを破棄します。

注 あるユーザーが個人モードで設定セッションを開始済の場合、他のユーザーは、標準モード、または自分自身の個人モードでセッションにログインできます。あるユーザーがすでにログインしている場合、他のユーザーは「別のユーザーが設定を修正中である」という警告メッセージを通知されます。

おすすめ 複数のユーザーアカウントが設定を変更できる状況では、常に `configure exclusive` コマンドか `configure private` コマンドを使用します。これを習慣にすれば、誰もが不注意によるミスを防げます。例えば、ジュニパーのエンジニアはこんな事例を聞きました。あるネットワーク管理者が、間違っ `delete interfaces` コマンドを入力してしまいました。その管理者はミスに気がつきましたが、そのステートメントを削除する代わりに、単に設定モードを出るだけで済ませてしまいました。ところがその後、別のユーザーがログインして設定をコミットした結果、デバイスのインタフェースがすべて削除されてしまったのです。幸い JUNOS ソフトウェアだったので、以前の設定にロールバックすることができました。

設定モードの基礎を理解する

設定モードのいくつかのオプションを使用して、設定候補を表示したり、階層を移動したりすることができます。

設定候補の内容を表示する

`show` コマンドは、デバイスの設定候補を表示します。設定の最上位で入力すると、CLI は設定候補の全体を表示します。以下のサンプルは、ある設定済みのデバイスのリスティングの一部です。

```
[edit]
mike@juniper1# show
version "9.2R1.3"
```

```

groups
{
  re0 {
    system {
      host-name juniper1;
    }
  }
}
<...>

```

設定に何も変更を加えていない場合、設定候補とデバイスのアクティブな（本番用）設定とは同一です。

階層がもっと深くなると、**show** コマンドはカレントの階層以下の設定を表示します。

```

[edit interfaces ge-5/0/0]
mike@juniper1# show
gigether-options {
  flow-control;
  auto-negotiation;
}
unit 0 {
  family inet {
    address 1.2.3.4/28;
  }
}

```

注 設定モードの **show** コマンドの使い方が運用モードとは違うことに気づかれたことでしょうか。各モードのコマンドは互いに独立しているので、**show** コマンドの動作は各モードで異なります。

設定の階層を移動する

階層の最上位から設定を編集することもできますが、たいいていは、あなたが変更作業をしている設定の特定の領域に移動してから、コマンドを追加したり、削除したりするほうが簡単です。例えば、設定に新しいサービスを追加したい場合、以下のように一連の **set** コマンドを発行することは間違いではありません。

```

[edit]
mike@juniper1# set system services finger
mike@juniper1# set system services ftp
mike@juniper1# set system services ssh
mike@juniper1# set system services telnet

```


しかし、`system services` ディレクトリに移動してから、以下のコマンドを発行するほうが簡単です。

```
[edit system services]
mike@juniper1# set finger
mike@juniper1# set ftp
mike@juniper1# set ssh
mike@juniper1# set telnet
```

どちらの場合も、最終的に以下の行が設定候補に追加されます。

```
[edit]
system {
  services {
    finger;
    ftp;
    ssh;
    telnet;
  }
}
```

CLIには設定モードの階層を移動するためのコマンドが4つあります。`edit`、`up`、`top`、`exit`です。

edit コマンド

`edit` コマンドを使用して、設定候補内の指定した場所にジャンプします。設定モードのバナー (`[edit ...]`) が変化して、階層内の新しい場所を示します。

```
[edit]
mike@juniper1# edit system services
```

```
[edit system services]
mike@juniper1#
```

いつも階層の最上位で `edit` コマンドを発行する必要はありません。例えば、`system syslog host log` 階層に移動する場合、階層の最上位で以下のコマンドを発行することもできます。

```
[edit]
mike@juniper1# edit system syslog host log
```

```
[edit system syslog host log]
mike@juniper1#
```

また、以下のように `edit` コマンドを順に発行して、同じ階層に移動することもできます。

```
[edit]
mike@juniper1# edit system

[edit system]
mike@juniper1# edit syslog

[edit system syslog]
mike@juniper1# edit host log

[edit system syslog host log]
mike@juniper1#
```

ある階層から `edit` コマンドを発行するときは、階層構造の中であなたがいる場所からの相対パスで指定します。

注 `edit` コマンドは UNIX の `cd` (change directory) コマンドと同様に機能し、階層ツリー内の指定した場所に直接移動できます。

あなたが移動先として設定の中に存在しない階層を指定した場合、CLI はその階層を新規作成します。しかし、後述の `set` コマンドを使用して明示的に階層を追加するほうが、あなたが何を作成したのかを正確に知ることができるので便利です。

up コマンド

`up` コマンドを使用すると階層の上位に移動できます。デフォルトでは 1 つ上の階層に移動します。コマンドの後に数字を追加すれば上位に移動する階層数を指定できます。

```
[edit interfaces fe-1/3/1 unit 0 family inet address 10.0.10.1]
mike@juniper1# up
```

```
[edit interfaces fe-1/3/1 unit 0 family inet]
mike@juniper1#
```

上記の例では、`interfaces`、`fe-1/3/1`、`unit 0`、`family inet`、`address 10.0.10.1` が、それぞれ階層構造の各層を表します。これを設定階層の最上位から見ると次のようになります。

```
[edit]
interfaces {
  fe-1/3/1 {
    unit 0 {
      family inet {
        address 10.0.10.1;
      }
    }
  }
}
```

up コマンド

`up` コマンドは、UNIX ディレクトリで `cd ..` コマンドを入力することと似ています。

topコマンド

topコマンドは、UNIXディレクトリでcd /コマンドを入力することと似ています。

top コマンド

top コマンドを使用すると階層の最上位に移動できます。

exit コマンド

exit コマンドを使用すると、階層構造内であなたが最後に編集コマンドを発行した場所に戻ります。設定階層の最上位で exit コマンドを発行すると、設定モードから出ることになります。

ショートカット 複数の移動用コマンドを組み合わせると階層を移動することもできます。例えば、top コマンドと edit コマンドを一緒に使用すれば、設定階層の別の部分に素早く移動できます。

```
[edit protocols ospf area]
mike@juniper1# top edit system login
```

```
[edit system login]
mike@juniper1#
```

top コマンドを show コマンドと併用すれば、階層構造の別のセクションにある設定の一部を表示できます。

```
[edit protocols ospf area]
mike@juniper1# top show system services
web-management {
  http {
    port 8080;
  }
}
```

試してみよう: 階層を移動する

設定の以下の階層に、edit、up、top、exit の4つのコマンドで移動してみましょう。

```
[edit interfaces]
[edit interfaces fe-0/0/0]
[edit]
[edit protocols bgp]
[edit protocols ldp interface fe-0/0/0.0]
[edit system]
[edit system services telnet]
[edit system syslog archive]
[edit system syslog]
[edit snmp v3]
[edit snmp v3 usm local-engine]
[edit snmp v3 target-parameters sample-parameters parameters]
```

設定を編集する

一連のコマンドを入力して設定候補を新規作成したり、変更したりします。設定ステートメントを追加したり、削除したりするコマンドが含まれます。

set コマンド

set コマンドは、ステートメントと値を設定候補に挿入します。例えば、デバイスに **ftp** サービスを追加したい場合、階層の最上位で次の **set** コマンドを発行します。

```
[edit]
mike@juniper1# set system services ftp
```

以下の数行が設定に追加されます。

```
system {
  services {
    ftp;
  }
}
```

setコマンド

setコマンドを使用してデバイスを完全に設定する方法を学習したい方は、JUNOSソフトウェア基本シリーズの Day One ブックレットの近刊を参照して下さい。刊行後、次のURLからダウンロードできます。

www.juniper.net/dayone/

set コマンドは、必要に応じてステートメントに値を追加する場合にも使います。例えば、デバイス名を「**juniper1**」とするには、**set** コマンドを次のように入力します。

```
[edit]
mike@devicename# set system host-name juniper1
```

以下の数行が設定に追加されます。

```
system {
  host-name juniper1;
}
```

delete コマンド

delete コマンドは、設定候補からステートメントを削除します。ステートメントを削除するということは、関連するデバイス、プロトコル、サービスを未設定の状態に戻すということです。コンテナステートメントを削除すると、その階層以下がすべて削除されます。

警告! **delete** コマンドは、下位にあるステートメントと識別子をすべて削除します。例えば、次のたった1行は、設定候補に含まれるプロトコル設定データをすべて削除することになります。

```
[edit]
mike@juniper1# delete protocols
```

おすすめ 自分が階層のどこにいるのか、`delete` コマンドで何が削除されるのかを承知した上で、コマンドを実行します。常に `[edit]` バナーで自分がいまいる階層を確認していれば、設定の中のあなたが変更したい部分だけに、コマンドの対象範囲を確実に限定することができます。

設定要素を削除した結果、設定ステートメントが空になった場合、CLI は設定候補からこの設定ステートメントを削除します。

試してみよう: 設定のセットアップと削除のコマンド

以下の手順に従って、IP アドレス「10.201.1.0/24」を持つシリアルインタフェースをセットアップし、続いて削除します。

1. 設定階層の最上位で次の `set` コマンドを入力します。

```
[edit]
mike@juniper1# set interfaces se-1/0/0 unit 0 family inet address 10.210.1.0/24
```

2. `show` コマンドで、設定に「`se`」インタフェースが追加されていることを確認します(次のサンプルでは、追加されたステートメントだけが表示されています。あなたの設定にはここに表示されている以上のデータがあるはずです)。

```
[edit]
mike@juniper1# show

interfaces {
  se-1/0/0 {
    unit 0 {
      family inet {
        address 10.210.1.0/24;
      }
    }
  }
}
```

3. この「`se`」インタフェースを、次の `delete` コマンドで削除します。

```
[edit]
mike@juniper1# delete interfaces se-1/0/0
```

4. `show` コマンドで、「`se`」インタフェースが設定から削除されていることを確認します。

もっと詳しく? デバイスの設定から大きな共通部分を削除する必要があるときは、ワイルドカードを使用すれば時間が節約できます。デバイスは設定候補の全体について文字列を検索して、この文字列を含むすべての行を削除することができます。ワイルドカードの使い方を詳しく学習したい方は、オンラインの「*JUNOS Software CLI User Guide*」の「Advanced Features」のセクションを参照してください。URLは次の通りです。

www.juniper.net/techpubs/

annotate コマンド

JUNOS CLI では、設定に関する注釈をリスティングの一部として残せます。この注釈は、あなた自身やチーム内の他のメンバーがトラブルシューティングをしようとするとき、あるいは設定を変更する必要があるときに大変便利です。注釈を付けたいときは、「**annotate**」に続けて注釈の文字列を記述します。

```
[edit]
mike@juniper1# annotate system this device is for training new
JUNOS users
```

設定モードで注釈を追加すると、その注釈は現在の階層でステートメントに関連付けられます。各ステートメントは、それに関連付けられた 1 行の注釈を 1 つだけ持っています。注釈を削除するには空の文字列が付いた **annotate** コマンドを使用します。

```
[edit]
mike@juniper1# annotate system ""
```

設定候補をコミットする

JUNOS CLI は、ユーザーが誤記、文字の脱落、その他のミスを検出して修正できるように支援する複数の機能を提供して、これらがトラブルになるのを防ぎます。設定候補の仕組みに加えて、これらの機能に含まれるのは、ファイル比較、設定候補のシンタックスとコンテキストのチェック、迅速なロールバックの実現、新しい設定がアクティブ化された後に切り離された元の設定をシステムにリストアすること、などです。

図 4.2 は、デバイス設定をめぐるファイル管理の詳細図です。アクティブな(本番用)設定はデバイスの運用ファイルです。これは、デバイスがブートシーケンスでロードした設定でもあります。設定候補は設定の更新を格納する作業用コピーです。**commit** コマンドはデバイスによる以下の遷移を発生させます(対象となる設定候補は検証チェックをパスしなければなりません)。

1. 設定候補をアクティブな設定にコピーします。この時点で、アクティブな設定と設定候補は同一です。
2. すべてのロールバック設定ファイルを1つずつデクリメントした上で、アクティブな設定を「rollback 0」として保存します。

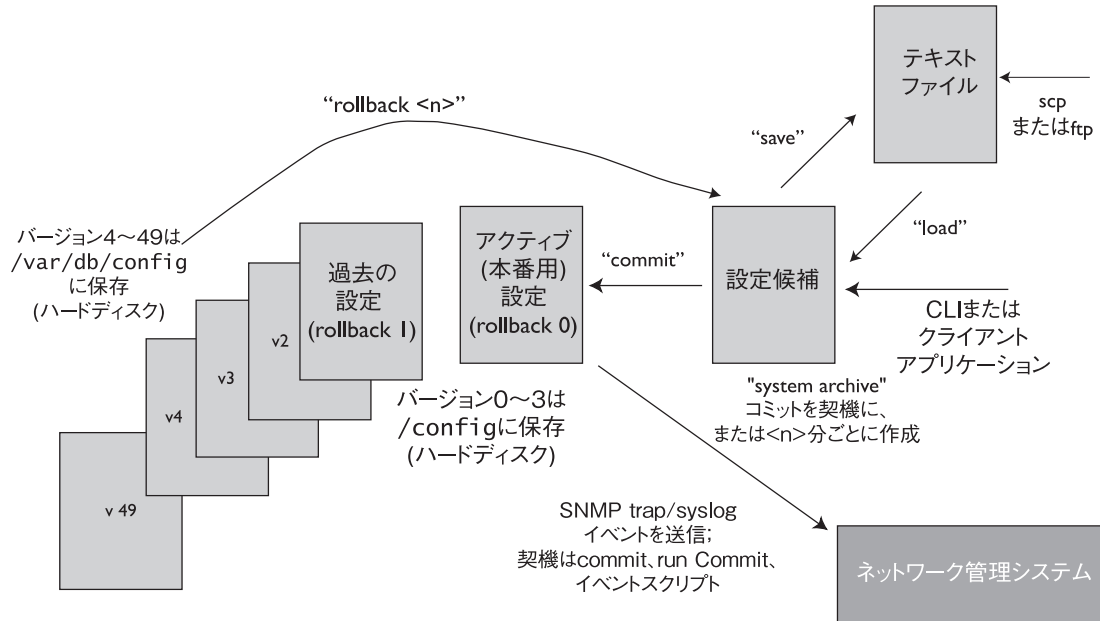


図4.2 JUNOSの設定ファイル管理

アクティブな(本番用)設定ファイル(rollback 0)と、新しい順に過去3世代のロールバック設定ファイル(rollback 1～3)は、/configディレクトリに保存されます。デバイスは、アーカイブされた残りのロールバック設定ファイルを、/var/db/configディレクトリに保存します。

注 アクティブな設定ファイルのファイル名は `juniper.conf.gz` で、ロールバック設定ファイルのファイル名は `juniper.conf.1.gz` から `juniper.conf.49.gz` までです(合計50個のアクティブな設定がアーカイブされます)。

compare コマンド

設定モードには、2つの設定の差分を表示する便利な `show | compare` コマンドがあります。

以下のサンプルでは設定候補を修正して、Telnet アクセスを有効化し、SSHとJ-Webアクセスを削除しています。

```
[edit system]
mike@juniper1# set services telnet
```

```
[edit system]
mike@juniper1# delete services web-management
```

```
[edit system]
mike@juniper1# delete services ssh
```

それでは、設定候補の変更結果とアクティブな設定を比較して、表示してみましょう。

```
[edit system services]
mike@juniper1# show | compare
- ssh;
+ telnet;
- web-management {
-   http {
-     port 8080;
-   }
- }
```

CLIは設定候補に追加された行をプラス記号(+)で、そこから削除された行をマイナス記号(-)で示します。

ショートカット 運用モードの `show configuration` コマンドは、現在のアクティブな(本番用)設定を表示します。このコマンドは、次のようにキーワード「run」を追加すれば、設定モードでも使用できます。

```
[edit]
mike@juniper1# run show configuration
```

commit check コマンド

CLIには、システムが設定候補を処理可能かどうかをチェックするコマンドもあります。`commit check` コマンドは、設定候補のロジックと完全性を検証しますが、このときの変更も有効化されることはありません。検証そのものは設定候補をコミット(`commit`)したときに実行される検証と同じです。システムが設定候補の中に問題を発見すると、それはあなたに通知されます。


```
[edit]
mike@juniper1# commit check
[edit interfaces lo0 unit 0 family inet]
  'address 192.168.69.1/24'
    Loopback addresses' prefix must be 32 bits
error: configuration check-out failed
```

おすすめ 設定候補を本番用設定としてアクティブ化する前に、入力結果を必ずチェックします。**show | compare** コマンドを使用して、予定した設定の要素とパラメタのすべてが設定候補の一部になっていることを確認します。**commit check** コマンドを入力して、変更を有効化せずに、システムに設定候補を検証させましょう。

commit コマンド

設定候補は設定になる予定のものにすぎず、あなたが **commit** コマンドを発行するまでは、デバイスがこの設定候補をたとえ一部でも使用することはありません。必要な変更をすべて入力し終わり、入力結果をシステムに再確認させて、ようやく設定候補をアクティブな(本番用)設定としてアクティブ化する用意が整います。

設定候補をアクティブ化するには **commit** コマンドを入力します。

```
[edit]
mike@juniper1# commit
commit complete
```

設定候補を実際にアクティブ化する前に、JUNOS ソフトウェアは基本的なシンタックスとセマンティクスをチェックします。例えば、JUNOS ソフトウェアは、ポリシーが定義された上で参照されていることを確認します。シンタックスとセマンティクスに関して問題が見つかった場合、**commit** コマンドはエラーを返します。

```
[edit]
mike@juniper1# commit
error: Policy error: Policy my-policy referenced but not defined
error: BGP: export list not applied
error: configuration check-out failed
```

設定候補 (または、その一部) をアクティブにするには、すべてのミスを修正する必要があります。

`commit complete` メッセージが表示されれば、新しい設定がデバイスにロードされてアクティブになっています。

```
[edit]
mike@juniper1# commit
commit complete
```

警告! デフォルトでは、複数のユーザーが設定を修正中に設定をコミットすると、すべてのユーザーの変更を保存してアクティブ化します（ただし、設定の個人モードでログインしているユーザーがいる場合はこの限りではありません。この章の「設定モードに入る」のセクションを参照して下さい）。

commit confirmed コマンド

リモートデバイスに対するセキュリティ機能の追加の仕方を間違えて、まさに自分自身がそのデバイスにアクセスするために使用しているインタフェースを、その新しいファイアウォールに拒否されてしまった、という経験を持つ人は少なくありません。あるいは、リモートデバイスをうっかり切り離してしまい、車に飛び乗って、真夜中に 4 時間もドライブして、そのデバイスをリセットしに行った、というような経験が、あなたにもあるかもしれません。JUNOS CLI の `commit confirmed` コマンドは、問題を起こした設定を自動的にロールバックで元の設定に戻すことにより、設定ミスによる労力と出費を避けることができます。

`commit confirmed` コマンドは、設定候補を 10 分間だけコミットします。これに続けて、2 回目のコミットになる `commit` コマンドが入力されない場合、デバイスは直前の設定に自動的にロールバックします。設定に潜在的な問題が含まれる可能性があるのなら、その安全策として `commit confirmed` コマンドを使用するとよいでしょう。

```
[edit]
mike@juniper1# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes
unless confirmed
commit complete
```

すべてが正常なら、新しい設定に対する 2 回目のコミットを実行して、この設定を確定する必要があります。

```
[edit]
mike@juniper1# commit
commit complete
```

2 回目のコミットになる `commit` コマンドを入力して設定を確定する、という操作がされない場合、CLI は 10 分後にデバイスを直前のアクティブな設定にロールバックします。この方法なら、デバイスを不注意で切り離してもロールバックを待つだけでよく、どうすればミスを復旧できるかをあれこれ思い悩む必要はありません。

```
Broadcast Message from root@juniper1
(no tty) at 08:10:17 UTC
Commit was not confirmed; automatic rollback complete.
```

デバイスがロールバックしたら設定候補のエラーをチェックし、再びコミットします。

デバイスがロールバックするまでの時間は変更可能で、コマンドに待機時間(単位は分)をつけて入力します。

```
[edit]
mike@juniper1# commit confirmed 2
commit confirm will be automatically rolled back in 2 minutes
unless confirmed
commit complete
```

おすすめ リモートデバイスを設定しているときは、常に `commit confirmed` コマンドで設定候補をアクティブ化するようにします。極めて経験豊富な JUNOS ユーザーでもこの作業上の“保険”は必要で、これがあつたおかげで自分のミスから救われ、余計な復旧作業をしなくて済んだという経験談はたくさんあります。

日常作業を自動化する

あなたのレパートリーに自動化スクリプトを加えれば、作業効率を最大化することができます。JUNOScript により、設定モードと運用モードのどちらの CLI コマンドでも、自分流のやりかたで自動化できます。自動化スクリプトを使用すれば、時間を節約できるだけでなく、ダウンタイムを短縮することもできます。設定ミスを防止し、問題の解決と修復をスピードアップできるからです。

「コミットスクリプト」は、設定のチェックをカスタマイズすることを可能にして、設定があなたのネットワークの標準とポリシーを確実に遵守するようにします。「op スクリプトとイベントスクリプト」は、コマンド出力を受け取り、これを検査して、次の適切なアクションを決定できるもので、問題の原因が見つかるまで、このプロセスを繰り返します。

もっと詳しく? スクリプトの実装方法については、オンラインの「*Configuration and Diagnostic Automation Guide*」を参照してください。URL は次の通りです。

www.juniper.net/techpubs/

設定をロールバックする

あなたが設定候補を新しいアクティブな設定としてコミットすると、JUNOS ソフトウェアは常に、そのとき置換されたアクティブな設定ファイルのコピーを自動的に保存します。新たに置換された設定を格納するとき、設定アーカイブの中では、以前のすべての設定ファイルのバージョン番号が 1 だけ増加します。各デバイスは以前にアクティブだったバージョンを新しい順に最大 49 個まで格納でき、同時に最新のアクティブな設定 (rollback 0) も格納しています。

この設定アーカイブには `rollback` コマンドでアクセスすることができ、戻りたいバージョン番号も指定できます。過去の設定ファイルの中で最も新しいものに戻るには、`rollback 1` コマンドを使用します。

```
[edit]
mike@juniper1# rollback 1
load complete
```

`rollback` コマンドは、要求したアーカイブファイルを設定候補としてロードします。この設定候補をすぐに使用したい場合、まず `show` コマンドで望みの設定に間違いがないことを確認し、次に `commit` コマンドで設定をアクティブ化します。

```
[edit]
mike@juniper1# show
<...>
```

```
[edit]
mike@juniper1# commit
commit complete
```

この自動バックアップメカニズムにより以前の設定に迅速に戻り、即座にそれを使用することや、素早く更新することが可能になります。

警告! 選択したロールバックファイルを本番用設定として実際にアクティブ化するには、設定候補をコミットする必要があることを忘れないでください。

ヒント アクティブな(本番用)設定とロールバックファイルの相違点がはっきりしない場合は、`show | compare` コマンドで調べます。

```
[edit interfaces]
mike@juniper1# show | compare rollback 2
[edit interfaces]
- fe-3/0/1 {
-   vlan-tagging;
-   unit 240 {
-     vlan-id 240;
-     family inet {
-       address 10.14.250.1/28;
```

rollback rescue コマンドは、レスキュー設定ファイルをロードします(詳細は、第3章、p.28の「おすすめ」を参照して下さい)。

```
-         address 10.14.250.17/28 {  
-             preferred;  
-         }  
-         address 10.14.250.33/28;  
-         address 10.14.250.49/28;  
-         address 10.14.250.65/28;  
-     }  
- }  
- }
```

ヒント `rollback` コマンドに疑問符 (?) をつけて使用すると、アーカイブ全体を一覧表示します。

```
[edit]  
mike@juniper1# rollback ?  
  
Possible completions:  
<[Enter]>   Execute this command  
0           2009-01-31 04:34:56 UTC by mike via cli  
1           2009-01-31 04:30:03 UTC by mike via cli  
2           2009-01-30 06:23:44 UTC by mike via cli  
<. . .>  
48          2008-11-03 08:00:03 UTC by mike via cli  
49          2008-11-03 07:45:21 UTC by mike via cli  
|           Pipe through a command
```

ヒント 設定候補をリセットして現在のアクティブな設定と同じにするには、`rollback` (または、`rollback 0`) コマンドを使用します。

事前にシステム変更を準備する

JUNOS CLI は、事前にシステムの変更を準備するためのオプションを提供します。これらのオプションは、設定の中のある部分を非アクティブ化して必要になるまでその状態を保ったり、あるいはアクティブ化の日時を指定したスケジュール要素になることができます。

新しいハードウェアの事前設定

他のシステムとは異なり、JUNOS ソフトウェアでは、ハードウェアを実際にインストールする前にインストールの準備をすることができます。JUNOS ソフトウェアは、本番用設定の一部に実在するハードウェア構成と対応関係がない場合、これを単に無視します。ハードウェアが利用できるようになれば、設定の新しく追加されたセクションがアクティブになります。

ハードウェアのインストール前に設定をセットアップできるオプションは非常に便利です。これは特に、ハードウェアをインストールする人とデバイスを設定する人が異なる場合に言えることで、リモートデバイスではよくあることです。以下のサンプルでは fe-3/0/0 インタフェースを設定していますが、これは明日インストールされるかもしれないと考えてください。

```
[edit]
mike@juniper1# edit interfaces fe-3/0/0 unit 0

[edit interfaces fe-3/0/0 unit 0]
mike@juniper1# set family inet address 192.168.1.254/24

[edit interfaces fe-3/0/0 unit 0]
mike@juniper1# commit
commit complete
```

commit at コマンド

設定変更のアクティブ化を指定の時刻に行うように準備したいことがあります。例えばメンテナンスの時間帯などです。commit at コマンドはこのオプションを提供します。

```
[edit]
mike@juniper1# commit at 02:00:00
commit check succeeds
commit will be executed at 2009-02-02 02:00:00 UTC
Exiting configuration mode
mike@juniper1>
```

保留中のコミット（およびコミットの履歴）を表示するには、show system commit コマンドを入力します。保留中のコミットを取り消すには clear system commit コマンドを使用します。

```
mike@juniper1>clear system commit
Pending commit cleared
```

deactivate コマンド

設定に変更を加え、その変更を使用する用意ができるまで、変更部分を非アクティブしておくようにマークを付けることもできます。デバイスは、この設定の変更部分を未定義であるかのように無視します。以下のサンプルでは、192.168.1.1 というアドレスで新しい BGP ネイバが設定されていますが、セッションへの対応が可能になるまでは非アクティブになっています。

```
[edit]
mike@juniper1# edit protocols bgp group internal

[edit protocols bgp group internal]
mike@juniper1# set neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# deactivate neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# show
type internal;
local-address 10.14.243.255;
export [ nhs accept-aggregates ];
neighbor 10.14.243.254;
inactive: neighbor 192.168.1.1;

[edit protocols bgp group internal]
mike@juniper1# commit
commit complete
```

実際に設定を変更する用意ができたなら、設定の変更部分をアクティブ化 (activate) してコミットするだけで、デバイスはその設定を使用するようになります。

```
[edit protocols bgp group internal]
mike@juniper1# activate neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# commit
commit complete
```

設定階層の任意の部分を指定して非アクティブにすることができます。デバイスはこの部分以下をすべて無視します。例えば、**internal** と呼ばれる BGP ネイバのグループ全体を非アクティブにすることができます。

```
[edit protocols bgp group internal]
mike@juniper1# up

[edit protocols bgp]
mike@juniper1# deactivate group internal
```

設定を簡単にするテクニック

一般的な設定には、ユーザーが名前を付けて定義した類似の要素が数多く含まれています。具体的にはインタフェース名、ポリシーステートメント、ファイアウォールフィルタなどです。JUNOS CLI は、これらのユーザー定義要素の設定を複製して、迅速に変更できるようにするコマンドを用意しています。

copy コマンド

`copy` コマンドは、指定した設定ステートメントを、その下位に設定されたすべてのステートメントを含めて複製します。`copy` コマンドを使用する場合、あるユーザー定義要素に関連付けられた設定を、新しく設定された類似の要素にコピーします。次に、この 2 番目の要素に必要な修正を加えます。

次の設定サンプルは、設定済のシリアルインタフェース「se-0/0/2」です。

```
[edit interfaces]
mike@juniper1# show
se-0/0/2 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.22.1/24;
    }
  }
}
```

ここで、`copy` コマンドを使用して、新しいインタフェース「se-0/0/1」をセットアップすることにします。複製されたインタフェースは、元のインタフェースと全く同じパラメータを持っています。次に、新しく作成されたインタフェース「se-0/0/1」の設定に必要な変更を加えます。例として、アドレスを変更することにします。

```
[edit interfaces]
mike@juniper1# copy se-0/0/2 to se-0/0/1
```

```
[edit interfaces]
mike@juniper1# delete se-0/0/1 unit 0 family inet address
10.0.22.1/24
```


コピーされたアドレスの削除を完了しました。このアドレスを新しいインタフェースの正しいアドレスに置き換えます。

```
[edit interfaces]
mike@juniper1# set se-0/0/1 unit 0 family inet address
10.0.36.2/24
```

大変簡単に新しいインタフェース「se-0/0/1」が作成されました。インタフェース「se-0/0/2」と共通のプロパティの大部分は元のままです。

```
[edit interfaces]
mike@juniper1# show

se-0/0/1 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.36.2/24;
    }
  }
}

se-0/0/2 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.22.1/24;
    }
  }
}
```

警告! copy コマンドで設定を変更するときは、必ず編集結果をチェックしてから設定候補をコミットしてください。複製したステートメントに必要な修正がもれなく加えられたことを確認してください。

忘れてはならないことは、設定の一部をコピーしただけでは設定が正しいものになるとは限らないことです。新しい設定の正しさをチェックしなければならず、必要なら設定に修正を加えます。

rename コマンド

`rename` コマンドは、ユーザー定義変数（ポリシー名、フィルタ名、IP アドレスなど）の値を変更したり、ユーザー定義要素の名前を変更するとき便利に使える簡単な手段です。

次のサンプルでは、Fast Ethernet インタフェース「fe-4/0/2」のアドレスが、間違っって「10.73.24.103/24」に設定されています。

```
[edit interfaces]
mike@juniper1# show

fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.103/24;
    }
  }
}
```

そこで、`rename` コマンドを使用して、値を「10.73.24.143/24」に変更することにします。

```
[edit interfaces]
mike@juniper1# rename fe-4/0/2 unit 0 family inet address
10.73.24.103/24 to address 10.73.24.143/24
```

変更は簡単に完了しましたが、結果を確認してみます。

```
[edit interfaces]
mike@juniper1# show

fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.143/24;
    }
  }
}
```

注 別の方法として、`rename` コマンドを使用する代わりに、`delete` コマンドでステートメントを削除してから、`set` コマンドで新しい値を追加することもできます。

ポートを切り替える: 便利な設定テクニック

テストのためだけに、ある接続を一時的に別のポートにつなぎ変えるというのを、これまでは何回も繰り返してきたのではないのでしょうか。JUNOSでは、このプロセスは簡単です。以下のサンプルに従えば、設定が「fe-2/0/1」から「fe-2/0/0」に移動します。

まず、いまのインタフェース設定を見てみましょう。

```
[edit]
mike@juniper1# show interfaces

fe-0/0/0 {
  description "MGMT INTERFACE - DO NOT DELETE";
  unit 0 {
    family inet {
      address 10.210.9.177/28;
    }
  }
}
fe-2/0/1 {
  vlan-tagging;
  unit 240 {
    vlan-id 240;
    family inet {
      address 10.14.243.238/28;
    }
  }
}
```

そこで、以下のように設定候補の中でポートを移動します。

```
[edit]
mike@juniper1# rename interfaces fe-2/0/1 to fe-2/0/0
```

設定候補には、この移動の結果が次のように反映されます。

```
[edit]
mike@juniper.net# show interfaces

fe-0/0/0 {
  description "MGMT INTERFACE - DO NOT DELETE";
  unit 0 {
    family inet {
      address 10.210.9.177/28;
    }
  }
}

fe-2/0/0 {
```

```
    vlan-tagging;  
    unit 240 {  
        vlan-id 240;  
        family inet {  
            address 10.14.243.238/28;  
        }  
    }  
}
```

もっと詳しく? 設定モードのコマンドについてもっと詳しく学習したい方は、オンラインの「*JUNOS Software CLI User Guide*」を参照してください。URL は次の通りです。

www.juniper.net/techpubs/

次にすること、次に行く場所 ...

www.juniper.net/junos

JUNOS の採用と教育に必要なものがすべてあります。

forums.juniper.net/jnet

ジュニパーが資金提供している J-Net Community フォーラムは、ジュニパーの製品、技術、ソリューションに関する情報、ベストプラクティス(最善事例)、疑問などを共有することを目的としています。メンバー登録してこの無料のフォーラムに参加しましょう。

www.juniper.net/techpubs

ジュニパーが開発した全製品の技術資料は、このサイトで無償でアクセスできます。JUNOS について知りたいことを、各製品ラインに基づいて探してください。

www.juniper.net/books

ジュニパーは複数の出版社と協力して、ネットワーク管理者に不可欠のテーマで技術書を執筆し、出版しています。各社から続々登場する新刊のリストを詳しくチェックしてみましょう。

www.juniper.net/training/fasttrack

トレーニングは、オンラインや開催地で、あるいは世界中にあるパートナーのトレーニングセンターで受講できます。ジュニパーネットワークスの技術認定プログラム (JNTCP) では、ジュニパー製品の設定やトラブルシューティングの能力を実証することで、認定証を獲得することができます。エンタープライズクラスのルーティング、スイッチング、セキュリティについて認定証を Fast Track プログラムで取得することをご希望の方は、利用可能な「Enterprise student courseware and lab guides」をご利用下さい。

この Day One ブックレットについて何かご意見があれば、メールで dayone@juniper.net までお知らせ下さい。

コマンドリファレンス

(この Day One ブックレットで解説したコマンドの要約)

設定モードのコマンド

- activate**/ 設定の一部をアクティブにする。
- annotate**/ 設定ステートメントに関する注釈を付記する。
- commit**/ 変更の設定候補セットをコミットする。
- commit at**/ 指定した時刻に、設定候補をコミットする。
- commit check**/ 変更を有効にすることなく、設定候補の妥当性を確認する。
- commit confirmed**/ ユーザーがコマンドの確認応答を返さない場合、自動的にロールバックする。
- compare**/ 2 つの設定の差分を表示する。
- copy**/ ステートメントをコピーする。
- deactivate**/ 設定の一部に非アクティブのマークを付ける。
- delete**/ 設定のステートメント(複数可) または識別子を削除する。
- edit**/ 指定された階層に移動する。
- exit**/ 設定の現在の階層から出る。最上位の階層にいる場合は設定モードから出る。
- exit configuration-mode**/ 設定モードを出る。
- help**/ オンボードヘルプを利用する。
- pipe**/ あるコマンドの出力を受け取り、それを別のコマンドの入力として使用するか、あるいはリダイレクトしてファイルに出力する。
- rename**/ 設定や識別子に新しい名前を割り当てる。
- rollback**/ 候補ファイルを、以前にコミットされた設定に戻す。
- run**/ 運用モードのコマンドを実行する。
- set**/ ステートメントの階層を 1 つ作成し、識別子に値を設定する。
- show**/ 設定候補を表示する。
- top**/ 第 1 階層(最上位) に移動する。
- up**/ 1 つ上の階層に移動する。

運用モードのコマンド

- clear**/ システム情報を削除する。
- configure**/ 設定モードに入る。
- configure exclusive**/ 設定候補に排他ロックをかけ、他のユーザーが編集できないようにする。
- configure private**/ ユーザーに自分専用の設定候補のコピーを与える。
- exit**/ 運用モードを出る。
- file copy**/ ファイルを作成してアーカイブに保存する。
- file list**/ デバイス上のファイルとディレクトリを一覧表示する。
- file show**/ ファイルの内容を表示する。
- help**/ オンボードヘルプを利用する。
- monitor**/ リアルタイムのデバッグ情報を表示する。
- ping**/ 別のホストにメッセージを送信して接続状況を検証する。
- pipe**/ あるコマンドの出力を受け取り、それを別のコマンドの入力として使用するか、あるいはリダイレクトしてファイルに出力する。
- request**/ ソフトウェアの新バージョンのインストール、リポート、シャットダウンを行う。
- restart**/ オペレーティングシステムの個々のデーモンをリスタートする。
- set**/ システムのプロパティを規定する。
- show**/ システム情報を表示する。
- ssh**/ 別のホスト上でセキュアシェルを起動する。
- start shell**/ C シェルインタフェースにログインする。
- telnet**/ ネットワーク上の別のデバイスやホストにターミナル接続する。
- traceroute**/ ある場所から別の場所にホップする IP パケットをすべて記録し、表示する。

Day One: Exploring the Junos CLI

This book is for first-time users of the Junos operating system and Juniper Networks products. It not only lays the foundation for learning Junos, it also facilitates understanding of subsequent booklets that appear in the Day One series.

The Junos CLI provides you with new tools, shortcuts, and safeguards. Learn about these new features and save yourself hours at the keyboard. The practical Day One format offers straightforward explanations, step-by-step instructions, and lots of examples. And, the “Try It Yourself” sections let you practice what you just read.

“Clear, concise, yet with enough technical detail to allow users new to Juniper Networks gear to complete the desired tasks on day one and allow more advanced users to have a handy cheat sheet for those tasks that they may not perform everyday. These guides are essential tools for all who use Juniper Networks equipment.”

Orin Blomberg, CCNP, CCIP, CCSP, JNCIS-M, JNCIS-ER, JNCIS-FWW

Day One: Exploring the Junos CLI shows you how to:

- Navigate the CLI’s operational mode and configuration mode.
- Understand the hierarchies that underlie each mode.
- Get onboard help and use keyboard shortcuts to speed up your work.
- Show device status, alarms, and other helpful information in operational mode.
- Modify, save, and load configuration files with minimal risk to operations.
- Use basic configuration mode commands such as show, set, and delete.
- Capitalize on the safety features of the Junos operating system commit model.
- Prepare system changes in advance.
- Use the shortcuts and tips of experienced users and avoid common problems.

Juniper Networks Day One booklets provide just the information you need on day one. They are written by subject matter experts and engineers who specialize in getting networks up and running. Look for other titles covering high-performance networking solutions at www.juniper.net/dayone. This book is available in both PDF and printed formats.



7100097-001-EN