



# アクセスリストおよびプレフィックスリストの実装

アクセスコントロールリスト (ACL) は、ネットワークトラフィックプロファイルをまとめて定義する 1 つ以上のアクセスコントロールエントリ (ACE) です。このプロファイルはその後、トラフィックフィルタリング、ルートフィルタリング、QoS 分類、アクセスコントロールなど、Cisco IOS XR ソフトウェアの機能で参照できます。各 ACL には、送信元アドレス、宛先アドレス、プロトコル、およびプロトコルに固有のパラメータなどの基準に基づく、アクション要素（許可または拒否）やフィルタ要素が含まれています。

プレフィックスリストはルートマップおよびルートフィルタリング操作に使用されるほか、ボーダーゲートウェイプロトコル (BGP) の多くのルートフィルタリングコマンドではアクセスリストの代わりに使用できます。プレフィックスは IP アドレスの一部であり、左端のオクテットの左端のビットから始まります。アドレスの何ビットがプレフィックスに属するかを正確に指定すると、プレフィックスを使用してアドレスを集約し、そのアドレスに対して再配布（フィルタルーティングアップデート）などの機能を実行できるようになります。

この章では、次の製品にアクセスリストおよびプレフィックスリストを実装するのに必要な新規のタスクおよび改訂されたタスクについて説明します：Cisco ASR 9000 シリーズルータ



(注)

この章に記載されているアクセスリストおよびプレフィックスリストのコマンドの詳細については、『Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference』を参照してください。この章で使用される他のコマンドの説明については、コマンドリファレンスのマスター索引を参照するか、またはオンラインで検索してください。

## アクセスリストおよびプレフィックスリストの実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

- [アクセスリストおよびプレフィックスリストの実装の前提条件、2 ページ](#)

- [アクセスリストおよびプレフィックスリストの実装の制約事項, 2 ページ](#)
- [ハードウェアの制限, 3 ページ](#)
- [アクセスリストおよびプレフィックスリストの実装に関する情報, 3 ページ](#)
- [ACL ベース転送の実装に関する情報, 14 ページ](#)
- [アクセスリストおよびプレフィックスリストの実装方法, 14 ページ](#)
- [ACL ベース転送を実装する方法, 36 ページ](#)
- [IPv6 ACL 用のピュア ACL ベース転送の設定, 46 ページ](#)
- [アクセスリストおよびプレフィックスリストの実装の設定例, 48 ページ](#)
- [その他の参考資料, 50 ページ](#)

## アクセスリストおよびプレフィックスリストの実装の前提条件

アクセスリストおよびプレフィックスリストの実装には、次の前提条件が適用されます。

すべてのコマンドタスク ID は、それぞれのコマンドリファレンスと、『Cisco IOS XR Task ID Reference Guide』に記載されています。タスクグループの割り当てについて支援が必要である場合は、システム管理者にお問い合わせください。

## アクセスリストおよびプレフィックスリストの実装の制約事項

アクセスリストおよびプレフィックスリストの実装には、次の制約事項が適用されます。

- IPv4 ACL は、ループバック インターフェイスおよびインターフレックス インターフェイスではサポートされません。
- IPv6 ACL は、ループバック、インターフレックス、および L2 イーサネット フローポイント (EFP) のメインまたはサブインターフェイスではサポートされません。

ACL ベース転送 (ABF) の実装には、次の制約事項が適用されます。

- ネクストホップオプションを持つ ACL を出方向に接続する設定、ネクストホップを持ち出方向に接続された ACL を変更する設定、ネクストホップを持つ ACE を拒否する設定のネクストホップ設定はサポートされていません。
- リリース 4.2.0 では、A9K-SIP-700 LC および ASR 9000 Enhanced Ethernet LC は ABFv4 および ABFv6 をサポートします。リリース 4.2.0 では、ASR 9000 Ethernet LC は ABFv6 をサポートせず、ABFv4 のみをサポートします。



(注) これには例外が1つあります。IP to TAG の場合、入力 LC が (ABF ネクスト ホップに基づいて) ラベルを提供するため、パケットはタグパケットとしてファブリックを横断します。このようなパケットは、A9K-SIP-700 によって問題なく処理されます。

- 低速パスでは ABF がサポートされないため、NPU から LC CPU へと入力方向にパントされたパケットは ABF では処理されません。
- フラグメンテーションを必要とする IP パケットは、ABF で処理されません。そのようなパケットは、従来の方法で転送されます。フラグメント化されたパケットは受信後、ABF によって処理されます。

## ハードウェアの制限

- ABF のサポートは、IPv4 およびイーサネット ラインカードのみが対象です。IPv6 とその他のインターフェイスはサポートされません。
- ABF は入力ラインカードの機能であるため、出力ラインカードは ABF に対応している必要があります。

## アクセスリストおよびプレフィックスリストの実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

## アクセスリストおよびプレフィックスリスト機能のハイライト

ここでは、アクセスリストとプレフィックスリストの機能のハイライトを示します。

- Cisco IOS XR ソフトウェア 特定のシーケンス番号を指定して、アクセスリストまたはプレフィックスリストのカウンタをクリアできます。
- Cisco IOS XR ソフトウェア 既存のアクセスリストまたはプレフィックスリストの内容を別のアクセスリストまたはプレフィックスリストにコピーできます。
- Cisco IOS XR ソフトウェア permit ステートメントまたは deny ステートメントにシーケンス番号を適用して、名前付きのアクセスリストまたはプレフィックスリストでこのようなステートメントの並べ替え、追加、または削除を実行できます。



(注) 並べ替えは、IPv4 プレフィックス リストのみが対象です。

- Cisco IOS XR ソフトウェア 標準アクセス リストと拡張アクセス リストとを区別しません。標準アクセス リストをサポートしているのは、下位互換性を確保するためです。

## IP アクセス リストの目的

アクセスリストは、パケットフィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。この処理は、ネットワークトラフィックを制限したり、ユーザやデバイスによるネットワークへのアクセスを制限したりするのに役立ちます。アクセスリストの用途は多様なので、多くのコマンドの構文でアクセス リストが参照されます。アクセス リストを使用して、次のようなことを実行できます。

- インターフェイスでの着信パケットのフィルタリング
- インターフェイスでの発信パケットのフィルタリング
- ルーティング アップデートの内容の制限
- アドレスまたはプロトコルに基づくデバッグ出力の制限
- vty へのアクセスの制御
- 輻輳回避、輻輳管理、プライオリティ キューイング、カスタム キューイングなどの高度な機能に使用されるトラフィックの特定または分類

## IP アクセス リストの機能

アクセス リストは、`permit` ステートメントと `deny` ステートメントで構成される順次リストです。これらのステートメントは、IP アドレス、場合によっては上位層 IP プロトコルに適用されます。アクセス リストには、参照に使用される名前があります。多くのソフトウェア コマンドは、構文の一部としてアクセス リストを受け取ります。

アクセス リストを設定して名前を付けることは可能ですが、アクセス リストを受け取るコマンドによってアクセス リストが参照されるまで、有効にはなりません。複数のコマンドから同じアクセス リストを参照できます。アクセス リストで、ルータに到達するトラフィック、またはルータ経由で送信されるトラフィックは制御できますが、ルータが送信元のトラフィックは制御できません。

## IP アクセス リストのプロセスとルール

IP アクセス リストを設定するときは、次のプロセスとルールを使用してください。

- アクセスリストの条件に対してフィルタリングされる各パケットの送信元アドレスや宛先アドレス、またはプロトコルがテストされます。一度に1つの条件（**permit** ステートメントまたは **deny** ステートメント）がテストされます。
- パケットがアクセスリストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセスリストのステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。つまり、一致すると、それ以降のエントリは考慮されません。
- アクセスリストでアドレスまたはプロトコルが拒否されると、パケットは廃棄され、インターネット制御メッセージプロトコル（ICMP）ホスト到達不能メッセージが返されます。ICMP は、Cisco IOS XR ソフトウェアで設定できます。
- 各アクセスリストの最後には暗黙の **deny** ステートメントがあるため、一致する条件がない場合は、パケットはドロップされます。つまり、各ステートメントに対してテストするときまでにパケットを許可または拒否しないと、パケットは拒否されます。
- アクセスリストには **permit** ステートメントを1つ以上含める必要があります。そうしないと、パケットはすべて拒否されます。
- 最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。同じ **permit** ステートメントまたは **deny** ステートメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。
- コマンドでアクセスリストを名前によって参照したときに、そのアクセスリストが存在しない場合は、すべてのパケットが通過します。
- 1つのインターフェイス、1つのプロトコル、1つの方向につき、許可されるアクセスリストは1つだけです。
- インバウンドアクセスリストは、ルータに到達するパケットを処理します。着信パケットの処理後に、アウトバウンドインターフェイスへのルーティングが行われます。インバウンドアクセスリストが効率的なのは、フィルタリングテストで拒否されたことでパケットが廃棄される場合、ルーティング検索のオーバーヘッドが抑えられるためです。パケットがテストで許可されると、そのパケットに対してルーティングの処理が実施されます。インバウンドリストの場合、**permit** とは、インバウンドインターフェイスでパケットの受信後に処理が継続されることを示します。**deny** とは、パケットが廃棄されることを示します。
- アウトバウンドアクセスリストの場合、パケットの処理後にルータから送信されます。着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンドアクセスリストで処理されます。アウトバウンドリストの場合、許可とは、出力バッファに対して送信されることを示し、拒否とは、パケットが廃棄されることを示します。
- アクセスリストは、使用中のアクセスグループによって適用されている場合には削除できません。アクセスリストを削除するには、まずアクセスリストを参照しているアクセスグループを削除してから、アクセスリストを削除します。

- **ipv4 access group** コマンドを使用するには、アクセスリストが存在している必要があります。

## IP アクセスリストを作成する際に役立つヒント

IP アクセスリストを作成する場合は、次の事項を考慮してください。

- アクセスリストは、インターフェイスに適用する前に作成します。
- 
- ネットワークまたはサブネットのより具体的な参照が、より全般的な参照よりも前に出現するように、アクセスリストを構成します。
- 個々のステートメントの用途をひと目で確認および理解しやすくするために、ステートメントの前または後に役立つ注記を書き込みます。

## 送信元アドレスと宛先アドレス

送信元アドレスと宛先アドレスは、IP パケットの最も一般的な2つのフィールドで、アクセスリストの基礎となります。送信元アドレスを指定して、特定のネットワーキングデバイスまたはホストからのパケットを制御します。宛先アドレスを指定して、特定のネットワーキングデバイスまたはホストに送信されるパケットを制御します。

## ワイルドカードマスクと暗黙のワイルドカードマスク

アドレスフィルタリングでは、アクセスリストエントリ内のアドレスビットとアクセスリストに送信されるパケットを比較するときに、ワイルドカードマスクを使用して、対応するIPアドレスビットを確認するか無視するかを指定します。管理者は、ワイルドカードマスクを慎重に設定することにより、許可または拒否のテストに1つまたは複数のIPアドレスを選択できます。

IP アドレスビット用のワイルドカードマスクでは、数値1と数値0を使用して、対応するIPアドレスビットをどのように扱うかを指定します。1と0は、サブネット（ネットワーク）マスクで意味する内容が逆になるため、ワイルドカードマスクは逆マスクとも呼ばれます。

- ワイルドカードマスクビット0は、対応するビット値を確認することを示します。
- ワイルドカードマスクのビット1は、対応するビット値を無視することを意味します。

アクセスリストステートメントでは、送信元アドレスまたは宛先アドレスにワイルドカードマスクを指定する必要はありません。**host** キーワードを使用した場合は、ワイルドカードマスクとして0.0.0.0を指定したものと見なされます。

サブネットマスクでは、ネットワークとサブネットを示す隣接ビットをマスクにする必要がありますが、それとは異なり、ワイルドカードマスクではマスクに非隣接ビットを使用できます。IPv6 アクセスリストでは、隣接ビットのみがサポートされます。

ワイルドカードビットの代わりに、CIDR形式 (/x) を使用することもできます。たとえば、アドレス1.2.3.4 0.255.255.255は1.2.3.4/8と表すことができます。

## トランスポート層の情報

トランスポート層の情報（パケットが TCP、UDP、ICMP、IGMP のいずれのパケットであるかなどの情報）に基づいてパケットをフィルタリングできます。

## IP アクセス リスト エントリ シーケンス番号

IP アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡易になります。この機能がない頃は、アクセスリスト内のエントリの位置を指定する方法はありませんでした。以前は、既存のリストの途中にエントリ（ステートメント）を挿入する場合、目的の位置の後にあるすべてのエントリを削除してから、新しいエントリを追加し、削除したすべてのエントリを再入力する必要がありました。これは手間がかかり、エラーが起こりやすい方法です。

IP アクセス リスト エントリ シーケンス番号機能を使用すると、アクセス リスト エントリにシーケンス番号を追加し、リスト内のエントリを並べ替えることができます。新しいエントリを追加する場合、アクセスリストの目的の位置に挿入されるようにシーケンス番号を選択します。必要に応じて、アクセスリストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

## シーケンス番号の動作

ここでは、シーケンス番号の動作を詳しく説明します。

- シーケンス番号のないエントリを複数適用すると、最初のエントリにシーケンス番号 10 が割り当てられ、それ以降のエントリには 10 ずつ増分したシーケンス番号が割り当てられます。最大シーケンス番号は 2147483646 です。生成したシーケンス番号がこの最大値を超えると、次のメッセージが表示されます。

Exceeded maximum sequence number.

- シーケンス番号のないエントリを 1 つ指定すると、アクセスリストの最後のシーケンス番号に 10 を加えたシーケンス番号が割り当てられ、リストの末尾に配置されます。
- ACL エントリは、トラフィック フローにもハードウェアのパフォーマンスにも影響を及ぼすことなく追加できます。
- グローバル コンフィギュレーション モードで新しいアクセス リストを入力すると、そのアクセス リストのシーケンス番号が自動的に生成されます。
- ルート プロセッサ (RP) のエントリとラインカード (LC) のエントリのシーケンス番号を常に同期できるように、分散機能がサポートされています。
- この機能は、名前付きの標準および拡張 IP アクセス リストと連動します。アクセス リストの名前を番号として指定できるため、番号も使用できます。

## IP アクセス リスト ログ メッセージ

Cisco IOS XR ソフトウェア 標準 IP アクセス リストで許可または拒否されたパケットに関するログメッセージが表示されます。つまり、パケットがアクセスリストに一致すると、そのパケットに関するログメッセージ情報がコンソールに送信されます。ログをコンソールに送信するメッセージのレベルは、グローバルコンフィギュレーションモードの **logging console** コマンドで制御します。

最初にパケットがアクセスリストをトリガーすると、すぐにログメッセージが生成されます。その後、5分間隔でパケットが収集されて表示または記録されます。ログメッセージにはアクセスリスト番号、パケットの許可または拒否に関する状況、パケットの送信元 IP アドレス、および直前の 5 分間に許可または拒否された送信元からのパケット数が示されます。

ただし、**{ ipv4 | ipv6 } access-list log-update threshold** コマンドを使用すると、アクセスリストに一致したパケットを許可または拒否する際に、ログメッセージを生成するパケットの数を設定できます。この手順は、5分間隔よりも短い頻度でログメッセージを受信する場合に実行することを推奨します。



注意

*number-of-matches* 引数を 1 に設定すると、ログメッセージはキャッシュされずにただちに送信されます。この場合、アクセスリストに一致するすべてのパケットについてログメッセージが生成されます。大量のログメッセージでシステムが過負荷になる可能性があるため、1 に設定することは推奨されません。

**{ ipv4 | ipv6 } access-list log-update threshold** コマンドを使用する場合でも、5分タイマーは有効なままなので、各キャッシュのメッセージ数に関係なく、5分が経過すると各キャッシュは空になります。ログメッセージを送信するタイミングに関係なく、しきい値が指定されていない場合と同様に、ログメッセージのキャッシュは消去され、カウントは 0 にリセットされます。



(注)

ログメッセージが多すぎて処理できない場合や、1秒以内に2つ以上のログメッセージを処理した場合には、ログメッセージパケットの一部がドロップされることがあります。この動作により、ログを生成するパケットの数が多くなっても、ルータが CPU サイクルを過度に使用することはありません。したがって、ロギング機能は課金ツールや、アクセスリストとの一致数を正確に把握するための情報源として使用しないでください。

## フラグメント制御付き拡張アクセスリスト

この機能が導入される前、非フラグメントパケットと、パケットの先頭フラグメントは、IP 拡張アクセスリストで処理していました（このようなアクセスリストを適用した場合）が、先頭以外のフラグメントはデフォルトで許可されていました。フラグメント制御付き IP 拡張アクセスリスト機能により、先頭以外のパケットもさらにきめ細かく制御できるようになりました。IP 拡張アクセスリストを適用するときに、パケットの先頭以外の IP フラグメントを調べるかどうかを指定できます。



先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、レイヤ 3 情報のみが含まれるアクセスリスト エントリを先頭以外のフラグメントに適用できるようになりました。フラグメントにはフィルタリングに必要な情報がすべて揃っており、それでエントリをフラグメントに適用できるというわけです。

この機能により、オプションの **fragments** キーワードが、IP アクセスリスト コマンドの **deny (IPv4)**、**permit (IPv4)**、**deny (IPv6)**、**permit (IPv6)** に追加されています。アクセスリスト エントリに **fragments** キーワードを指定することにより、その特定のアクセスリスト エントリは、パケットの先頭以外のフラグメントにのみ適用されます。フラグメントは、指定内容に応じて許可または拒否されます。

**fragments** キーワードの有無に応じたアクセスリスト エントリの動作をまとめると、次のようになります。

アクセスリストエントリの状態	結果
<p><b>fragments</b> キーワードがなく、すべてのアクセスリストエントリ情報が一致する</p>	<p>アクセスリストエントリにレイヤ3情報のみが含まれている場合：</p> <ul style="list-style-type: none"> <li>• エントリは、非フラグメント パケット、先頭フラグメント、先頭以外のフラグメントに適用されます。</li> </ul> <p>アクセスリストエントリにレイヤ3情報とレイヤ4情報が含まれている場合：</p> <ul style="list-style-type: none"> <li>• エントリは、非フラグメント パケットと先頭フラグメントに適用されます。 <ul style="list-style-type: none"> <li>◦ エントリが一致し、かつ <b>permit</b> ステートメントである場合、パケットまたはフラグメントは許可されます。</li> <li>◦ エントリが一致し、かつ <b>deny</b> ステートメントである場合、パケットまたはフラグメントは拒否されます。</li> </ul> </li> <li>• エントリは、次の方法で先頭以外のフラグメントにも適用されます。先頭以外のフラグメントにはレイヤ3情報のみが含まれているため、アクセスリストエントリのレイヤ3の部分のみが適用されます。アクセスリストエントリのレイヤ3の部分一致し、 <ul style="list-style-type: none"> <li>◦ エントリが <b>permit</b> ステートメントである場合、先頭以外のフラグメントは許可されます。</li> <li>◦ エントリが <b>deny</b> ステートメントの場合は、次のアクセスリストエントリが処理されます。</li> </ul> </li> </ul> <p>(注) 先頭以外のフラグメントと非フラグメントや先頭フラグメントとでは、<b>deny</b> ステートメントの処理が異なることに注意してください。</p>

アクセスリストエントリの状態	結果
<p><b>fragments</b> キーワードがあり、すべてのアクセスリストエントリ情報が一致する</p>	<p>アクセスリストエントリは、先頭以外のフラグメントにのみ適用されます。</p> <p>(注) レイヤ4情報を含むアクセスリストエントリに <b>fragments</b> キーワードは設定できません。</p>

すべてのアクセスリストエントリに **fragments** キーワードを追加しないでください。IPパケットの先頭フラグメントは非フラグメントと見なされ、それ以降のフラグメントとは独立して扱われるためです。先頭フラグメントは **fragments** キーワードが含まれているアクセスリスト **permit** エントリまたは **deny** エントリとは一致しないため、パケットは次のアクセスリストエントリと比較されます。この比較は、**fragments** キーワードが含まれていないアクセスリストエントリによってパケットが許可または拒否されるまで続きます。したがって、**deny** エントリごとに、2つのアクセスリストエントリが必要になる場合があります。ペアの最初の **deny** エントリには **fragments** キーワードは含まれず、初期フラグメントに適用されます。ペアの2番めの **deny** エントリには **fragments** キーワードは含まれ、以降のフラグメントに適用されます。同じホストに複数の **deny** アクセスリストエントリがあり、それぞれのレイヤ4ポートが異なる場合、そのホストに追加する必要があるのは、**fragments** キーワードを指定した **deny** アクセスリストエントリ1つだけです。このように、パケットのすべてのフラグメントは、アクセスリストによって同様に扱われます。

IPデータグラムのパケットフラグメントは個々のパケットと見なされ、各フラグメントはアクセスリストアカウントとアクセスリスト違反カウントの1つのパケットとして個別にカウントされます。



(注) アクセスリストおよびIPフラグメントに関するあらゆるケースを **fragments** キーワードで解決できるわけではありません。

## ポリシールーティング

ポリシールーティングが **match ip address** コマンドに基づくものであり、アクセスリストのエントリがレイヤ4～レイヤ7の情報に一致した場合、フラグメンテーションとフラグメント制御機能はポリシールーティングに影響を及ぼします。先頭フラグメントがポリシールーティングされなかった場合でも、先頭以外のフラグメントがアクセスリストを通過し、ポリシールーティングされることがあります。その逆もまた同じです。

前に説明したようにアクセスリストエントリに **fragments** キーワードを使用すると、先頭フラグメントと先頭以外のフラグメントに対するアクションの照合を改善できるため、ポリシールーティングが想定どおりに機能する可能性が高くなります。

## アクセスリストのエントリに関するコメント

**remark** アクセスリスト コンフィギュレーション コマンドを使用すると、名前付き IP アクセスリストのエントリに関するコメント（注釈）を含めることができます。コメントを含めると、ネットワーク管理者がアクセスリストを理解し、精査しやすくなります。1つのコメント行の最大長は255文字です。

コメントは、**permit** ステートメントまたは **deny** ステートメントの前後どちらにでも配置できます。コメントがどの **permit** ステートメントまたは **deny** ステートメントの説明であるのかが明確になるように、コメントの位置に関して一貫性を保つようしてください。たとえば、あるコメントは対応する **permit** または **deny** ステートメントの前にあり、他のコメントは対応するステートメントの後ろにあると、混乱を招きます。コメントに順番を付けることができます。

アクセスリストの作成後、アクセスリストをインターフェイスまたは端末回線に適用することを忘れないでください。詳細については、[アクセスリストの適用](#)、(18 ページ) を参照してください。

## アクセスコントロールリストカウンタ

Cisco IOS XR ソフトウェアでは、ACL カウンタがハードウェアとソフトウェアの両方で維持されます。ハードウェアカウンタは、アクセスグループをインターフェイスに適用するなど、パケットフィルタリングの用途に使用します。ソフトウェアカウンタは、主にソフトウェアパケット処理に関するあらゆる用途に使用できます。

パケットフィルタリングでは、ACE ごとに 64 ビットのハードウェアカウンタが使用されます。同じラインカードにある所定の方向のインターフェイスに同じアクセスグループを適用した場合、ACL のハードウェアカウンタは2つのインターフェイス間で共有されます。

特定のアクセスグループのハードウェアカウンタを表示するには、EXEC モードで **show access-lists ipv4** [*access-list-name hardware* {*ingress* | *egress*}] [*interface type interface-path-id*] {*location node-id*} コマンドを使用します。

ハードウェアカウンタをクリアするには、EXEC モードで **clear access-list ipv4** *access-list-name* [*hardware* {*ingress* | *egress*}] [*interface type interface-path-id*] {*location node-id*} コマンドを使用します。

わずかながらパフォーマンスが低下するため、IPv4 ACL に対するハードウェアカウンタはデフォルトでは無効になっています。ハードウェアカウンタをイネーブルにするには、インターフェイス コンフィギュレーション モードで **ipv4 access-group** *access-list-name* {*ingress* | *egress*} [*hardware-count*] コマンドを使用します。このコマンドは必要に応じて使用できるため、カウンタは指定したインターフェイスに対してのみイネーブルになります。

ソフトウェアカウンタは、ソフトウェアがパケットを処理すると更新されます。たとえば、例外パケットを LC CPU にパントして処理した場合や、ルーティングプロトコルが ACL を使用した場合などです。維持されるソフトウェアカウンタというのは、その ACL を使用するすべてのソフトウェアアプリケーションの集合体です。ソフトウェア専用の ACL カウンタを表示するには、EXEC モードで **show access-lists ipv4** *access-list-name* [*sequence number*] コマンドを使用します。

ここに挙げた情報は、ハードウェア カウントが常にイネーブルになっていることを除いて、すべて IPv6 にも当てはまります。IPv6 アクセス グループのコマンドライン インターフェイス (CLI) には **hardware-count** オプションがありません。

## プレフィックス リストを使用した BGP フィルタリング

プレフィックス リストは、BGP ルート フィルタリング コマンドの多くでアクセス リストの代わりに使用できます。プレフィックス リストを使用した場合の利点は次のとおりです。

- サイズの大きなリストをロードしてルート ルックアップを実施する場合のパフォーマンスが大幅に向上します。
- 差分更新がサポートされます。
- CLI の使い勝手が向上します。アクセス リストを使用して BGP 更新をフィルタリングするための CLI は、パケット フィルタリング形式を使用しているため、わかりにくく使い勝手もよくありません。
- 柔軟性が高まります。

コマンドでプレフィックス リストを使用するには、あらかじめプレフィックス リストをセットアップしておく必要があります。プレフィックス リストのエントリには、シーケンス番号を割り当ててください。

## プレフィックス リストでトラフィックをフィルタリングする仕組み

プレフィックス リストによるフィルタリングでは、ルートのプレフィックスが、プレフィックス リストに記載されているプレフィックスと照合されます。一致すると、一致したルートが使用されます。具体的には、プレフィックスを許可するか、拒否するかは次のルールに基づきます。

- 空のプレフィックス リストはすべてのプレフィックスを許可します。
- 特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、暗黙の **deny** が適用されます。
- プレフィックス リストの複数のエントリが特定のプレフィックスと一致したときは、最も長く、最も具体的な一致が選択されます。

シーケンス番号は自動的に生成されます。ただし、この自動生成をディセーブルにしている場合を除きます。シーケンス番号の自動生成をディセーブルにしている場合は、IPv4 または IPv6 のプレフィックス リスト コンフィギュレーション コマンドの **permit** コマンドおよび **deny** コマンドで *sequence-number* 引数を使用して、各エントリのシーケンス番号を指定する必要があります。プレフィックス リストのエントリを削除するには、*sequence-number* 引数を指定した **permit** コマンドまたは **deny** コマンドの **no** 形式を使用してください。

**show** コマンドの出力には、シーケンス番号が含まれます。

## ACL ベース転送の実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

### ACL ベース転送の概要

統合ネットワークは、音声、ビデオ、およびデータを伝送します。トラフィックによっては、ルーティングプロトコルが算出したパスを使用するのではなく、特定のパスにルーティングすることが必要になる場合があります。これを実現するための簡単なソリューションは、ACL 設定にネクストホップアドレスを指定することです。これで、パケットベースで宛先アドレスをルックアップするのではなく、ACL に設定したネクストホップアドレスを使用して指定の宛先にパケットを転送できるようになります。ACL 設定でネクストホップを使用して転送するというこの機能は、ACL ベース転送 (ABF) と呼ばれます。

ACL ベース転送を使用すると、ブロードキャスト TV over IP、IP テレフォニー、データなどを対象としたサービスを複数のプロバイダーから選択することが可能になり、カフェテリア形式でインターネットにアクセスできます。サービスプロバイダーは、ユーザトラフィックをさまざまなコンテンツプロバイダーに迂回させることができます。

### ABF-OT

ユーザが適切なネクストホップを柔軟に選択できるようにするため、ABF の機能が強化され、オブジェクトトラッキング (OT) と情報をやり取りできるようになりました。これは、次の機能に影響を及ぼします。

- CEF でのプレフィックスのトラッキング
- ラインステートプロトコルのトラッキング
- IPSLA (IP サービス レベル契約)

### オブジェクトトラッキングでの IPSLA のサポート

OT モジュールは、IPSLA モジュールとやり取りして到達可能性情報を取得します。ルータは、IPSLA を使って定期的に測定を実施します。

## アクセスリストおよびプレフィックスリストの実装方法

Cisco ASR 9000 SIP 700 ラインカードおよび ASR 9000 イーサネット ラインカードで IPv6 ACL をサポートするようになりました。これに関連する基準は次のとおりです。

- ACL 対応のインターフェイス：1000（各方向 500 ずつ）、ASR 9000 イーサネットラインカードの場合は 4000
- 一意の ACL：512（それぞれに 5 個の ACE）、ASR 9000 イーサネットラインカードの場合は 2000
- ACL あたりの最大 ACE 数：8000（ASR 9000 イーサネットラインカードの場合は、LC モデルに基づいて 16000、8000、4000 のいずれか）
- IPv6 ACL ログも、今後サポートする予定です。

ここでは、次の手順について説明します。

## 拡張アクセスリストの設定

このタスクでは、拡張 IPv4 または IPv6 アクセスリストを設定します。

### 手順の概要

1. **configure**
2. **{ipv4 | ipv6} access-list name**
3. **[ sequence-number ] remark remark**
4. 次のいずれかを実行します。
  - **[ sequence-number ] {permit | deny} source source-wildcard destination destination-wildcard [precedence precedence] [dscp dscp] [fragments] [packet-length operator packet-length value] [log | log-input]**
  - **[ sequence-number ] {permit | deny} protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator {port | protocol-port}] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address} [operator {port | protocol-port}] [dscp value] [routing] [authen] [destopts] [fragments] [packet-length operator packet-length value] [log | log-input]**
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no sequence-number** コマンドを使用します。
6. 次のいずれかのコマンドを使用します。
  - **end**
  - **commit**
7. **show access-lists {ipv4 | ipv6} [access-list-name hardware {ingress | egress}] [interface type interface-path-id] {sequence number | location node-id} | summary [access-list-name] | access-list-name [sequence-number] | maximum [detail] [usage {pfilter location node-id}]**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>configure</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# configure</pre>	<p>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。</p>
ステップ 2	<p><b>{ipv4   ipv6} access-list name</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2</pre>	<p>IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、名前付きアクセス リストを設定します。</p>
ステップ 3	<p><b>[ sequence-number ] remark remark</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out</pre>	<p>(任意) 名前付きのアクセス リストに <b>permit</b> ステートメントまたは <b>deny</b> ステートメントに関するコメントを書くことができます。</p> <ul style="list-style-type: none"> <li>注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。</li> <li><b>permit</b> ステートメントまたは <b>deny</b> ステートメントの前後どちらにも設定できますが、どちらかの位置に統一することを推奨します。</li> </ul>
ステップ 4	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li><b>[ sequence-number ] { permit   deny } source source-wildcard destination destination-wildcard [ precedence precedence ] [ dscp dscp ] [ fragments ] [ packet-length operator packet-length value ] [ log   log-input ]</b></li> <li><b>[ sequence-number ] { permit   deny } protocol { source-ipv6-prefix/prefix-length   any   host source-ipv6-address } [ operator { port   protocol-port } ] { destination-ipv6-prefix/prefix-length   any   host destination-ipv6-address } [ operator { port   protocol-port } ] [ dscp value ] [ routing ] [ authen ] [ destopts ] [ fragments ] [ packet-length operator packet-length value ] [ log   log-input ]</b></li> </ul>	<p>IPv4 アクセス リスト <b>acl_1</b> に許可または拒否の条件を 1 つ以上指定します。</p> <ul style="list-style-type: none"> <li>任意の <b>log</b> キーワードを指定すると、エントリに一致したパケットに関する情報ログメッセージがコンソールに送信されます。</li> <li>任意の <b>log-input</b> キーワードは、ログメッセージに入力インターフェイスも含まれることを除いて、<b>log</b> キーワードと同じように機能します。</li> </ul> <p>または</p> <p>IPv6 アクセス リスト <b>acl_2</b> に許可または拒否の条件を 1 つ以上指定します。</p> <ul style="list-style-type: none"> <li>IPv6 オプションヘッダーおよび任意の上位層プロトコルタイプ情報に基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、<b>deny</b> (IPv6) コマンドおよび <b>permit</b> (IPv6) コマンドを参照してください。</li> </ul>



	コマンドまたはアクション	目的
	<p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255 RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000</pre>	<p>(注) どのIPv6アドレスリストにも、ネイバーアドバタイズメントおよび送信要求に使用される暗黙の <b>permit 2</b> つあります。それは暗黙的ネイバー探索ネイバーアドバタイズメント (NDNA) と暗黙的ネイバー探索ネイバー送信要求 (NDNS) です。</p> <p>(注) どのIPv6アクセスリストにも最後の一致条件として暗黙の <b>deny ipv6 any any</b> ステートメントがあります。1つのIPv6アクセスリストには、暗黙の <b>deny ipv6 any any</b> ステートメントを有効にするために少なくとも1つのエントリが含まれる必要があります。</p>
<p>ステップ5</p>	<p>必要に応じてステップ4を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、<b>no sequence-number</b> コマンドを使用します。</p>	<p>アクセスリストは変更できます。</p>
<p>ステップ6</p>	<p>次のいずれかのコマンドを使用します。</p> <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# end</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
<p>ステップ7</p>	<p><b>show access-lists {ipv4   ipv6} [access-list-name hardware {ingress   egress} [interface type interface-path-id] {sequence number   location</b></p>	<p>(任意) 現在のIPv4またはIPv6アクセスリストの内容を表示します。</p>

コマンドまたはアクション	目的
<p><code>node-id</code>   <b>summary</b> [<i>access-list-name</i>]   <i>access-list-name</i> [<i>sequence-number</i>]   <b>maximum</b> [<b>detail</b>] [<b>usage</b> {<i>pfilter location node-id</i>}]</p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1</pre>	<ul style="list-style-type: none"> <li>特定のアクセスリストの内容を表示するには、<i>access-list-name</i> 引数を使用します。</li> <li>方向（入力または出力）とアクセスリストを指定して、それを使用するすべてのインターフェイスのハードウェアの内容とカウンタを表示するには、<b>hardware</b>、<b>ingress</b> または <b>egress</b>、および <b>location</b> または <b>sequence</b> の各キーワードを使用します。インターフェイスのアクセスグループを設定するには、イネーブルにするアクセスリストハードウェアカウンタに対して <b>ipv4 access-group</b> コマンドを使用します。</li> <li>現在の IPv4 または IPv6 アクセスリストをまとめたサマリーを表示するには、<b>summary</b> キーワードを使用します。</li> <li>インターフェイスの統計情報を表示するには、<b>interface</b> キーワードを使用します。</li> </ul>

### 次の作業

アクセスリストを作成したら、回線またはインターフェイスに適用する必要があります。アクセスリストを適用する方法については、[アクセスリストの適用](#)、(18 ページ) を参照してください。

一意のアクセスリストエントリ (ACE) の追加または削除中に、ACL コミットが失敗します。これは、割り当てられたマネージャプロセスが存在しないために発生します。config-ipv4-acl モードを終了してコンフィギュレーションモードに戻り、再び config-ipv4-acl モードを開始してから、最初の ACE を追加してください。

## アクセスリストの適用

作成したアクセスリストを機能させるには、そのアクセスリストを参照する必要があります。アクセスリストは、発信インターフェイスまたは着信インターフェイスに適用できます。ここでは、端末回線とネットワークインターフェイスの両方に対してこのタスクを実行するためのガイドラインを示します。

すべての仮想端末回線にユーザが接続する可能性があるため、すべての仮想端末回線に同じ制約を設定する必要があります。

着信アクセスリストの場合、パケットの受信後、Cisco IOS XR ソフトウェアはアクセスリストに照らしてそのパケットの送信元アドレスをチェックします。アクセスリストがアドレスを許可している場合は、パケットの処理を継続します。アクセスリストがアドレスを拒否している場合

は、パケットを廃棄し、ICMPホスト到達不能メッセージを返します。ICMPメッセージは設定可能です。

発信アクセスリストの場合、パケットを受信して管理下のインターフェイスに転送した後、アクセスリストに照らしてパケットの送信元アドレスをチェックします。アクセスリストがアドレスを許可している場合は、パケットを送信します。アクセスリストがアドレスを拒否している場合は、パケットを廃棄し、ICMPホスト到達不能メッセージを返します。

まだ定義されていないアクセスリストをインターフェイスに適用すると、アクセスリストがまだインターフェイスに適用されていないものと解釈し、すべてのパケットを容認します。ネットワークで未定義のアクセスリストをセキュリティの手段として使用する場合は、この動作に留意してください。

## インターフェイスへのアクセスの制御

このタスクでは、アクセスリストをインターフェイスに適用して、そのインターフェイスへのアクセスを制限します。

アクセスリストは、発信インターフェイスまたは着信インターフェイスに適用できます。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のいずれかを実行します。
  - **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
  - **ipv6 access-group** *access-list-name* {**ingress** | **egress**} [**interface-statistics**]
4. 次のいずれかを実行します。
  - end
  - commit

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>configure</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# configure</pre>	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。

	コマンドまたはアクション	目的
ステップ 2	<p><b>interface</b> <i>type interface-path-id</i></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/2</pre>	<p>インターフェイスを設定し、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。</p> <ul style="list-style-type: none"> <li>• <i>type</i> 引数には、インターフェイス タイプを指定します。インターフェイス タイプの詳細については、疑問符 (?) オンラインヘルプ機能を使用してください。</li> <li>• <i>instance</i> 引数には、物理インターフェイス インスタンスまたは仮想インスタンスを指定します。 <ul style="list-style-type: none"> <li>◦ 物理インターフェイス インスタンスの表記方法は <i>rack/slot/module/port</i> です。値を区切るスラッシュ (/) は、表記の一部として必要です。</li> <li>◦ 仮想インターフェイス インスタンスの数値範囲は、インターフェイス タイプによって異なります。</li> </ul> </li> </ul>
ステップ 3	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <b>ipv4 access-group</b> <i>access-list-name</i> {<b>ingress</b>   <b>egress</b>} [<b>hardware-count</b>] [<b>interface-statistics</b>]</li> <li>• <b>ipv6 access-group</b> <i>access-list-name</i> {<b>ingress</b>   <b>egress</b>} [<b>interface-statistics</b>]</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-in-filter in  RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-out-filter out</pre>	<p>インターフェイスへのアクセスを制御します。</p> <ul style="list-style-type: none"> <li>• 特定の IPv4 または IPv6 アクセス リストを指定するには、<i>access-list-name</i> 引数を使用します。</li> <li>• 着信パケットをフィルタリングするには <b>in</b> キーワードを使用し、発信パケットをフィルタリングするには <b>out</b> キーワードを使用します。</li> <li>• IPv4 アクセス グループのハードウェア カウンタをイネーブルにするには、<b>hardware-count</b> キーワードを使用します。 <ul style="list-style-type: none"> <li>◦ IPv6 アクセス グループのハードウェア カウンタは、自動的にイネーブルになります。</li> </ul> </li> <li>• ハードウェアにインターフェイスごとの統計情報を指定するには、<b>interface-statistics</b> キーワードを使用します。</li> </ul> <p>この例では、GigabitEthernet 0/2/0/2 から発着信されるパケットにフィルタを適用します。</p>
ステップ 4	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-if)# end</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</pre>

	コマンドまたはアクション	目的
	または RP/0/RSP0/CPU0:router (config-if) # commit	<ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## 回線へのアクセスの制御

このタスクでは、回線にアクセスリストを適用して、その回線へのアクセスを制御します。

### 手順の概要

1. **configure**
2. **line {aux | console | default | template *template-name*}**
3. **access-class *list-name*{ingress | egress}**
4. 次のいずれかのコマンドを使用します。
  - **end**
  - **commit**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b>  例：  RP/0/RSP0/CPU0:router# configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	<p><b>line {aux   console   default   template template-name}</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# line default</pre>	<p>補助、コンソール、デフォルト、またはユーザ定義の回線テンプレートを指定し、回線テンプレート コンフィギュレーションモードを開始します。</p> <ul style="list-style-type: none"> <li>• ラインテンプレートは、物理端末回線接続（コンソールポートおよびAUXポート）およびVTY接続を設定して管理するために使用する属性のコレクションです。Cisco IOS XR ソフトウェアでは、次のテンプレートを使用できます。 <ul style="list-style-type: none"> <li>◦ 補助回線テンプレート：補助回線に適用される回線テンプレート。</li> <li>◦ コンソールラインテンプレート：コンソール回線に適用されます。</li> <li>◦ デフォルトラインテンプレート：物理および仮想端末回線に適用されます。</li> <li>◦ ユーザ定義ラインテンプレート：仮想端末回線の範囲に適用できます。</li> </ul> </li> </ul>
ステップ 3	<p><b>access-class list-name {ingress   egress}</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-line)# access-class acl_2 out</pre>	<p>IPv4 または IPv6 アクセスリストを使用して、着信接続および発信接続を制限します。</p> <ul style="list-style-type: none"> <li>• 例では、IPv6 アクセスリスト <code>acl_2</code> を使用して、デフォルトの回線テンプレートの発信接続をフィルタリングしています。</li> </ul>
ステップ 4	<p>次のいずれかのコマンドを使用します。</p> <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# end</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## プレフィックスリストの設定

このタスクでは、IPv4 または IPv6 プレフィックスリストを設定します。

### 手順の概要

1. `configure`
2. `{ipv4 | ipv6} prefix-list name`
3. `[ sequence-number ] remark remark`
4. `[ sequence-number ] {permit | deny} network/length [ge value] [le value] [eq value]`
5. 必要に応じてステップ 4 を繰り返します。 エントリを削除するには、**no sequence-number** コマンドを使用します。
6. 次のいずれかを実行します。
  - `end`
  - `commit`
7. 次のいずれかを実行します。
  - `show prefix-list ipv4 [name] [sequence-number]`
  - `show prefix-list ipv6 [name] [sequence-number] [summary]`
8. `clear {ipv4 | ipv6} prefix-list name [sequence-number]`

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>  例 : <code>RP/0/RSP0/CPU0:router# configure</code>	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。

	コマンドまたはアクション	目的
ステップ 2	<p><b>{ipv4   ipv6} prefix-list name</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv4 prefix-list pfx_1 または RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2</pre>	<p>IPv4 または IPv6 プレフィックス リスト コンフィギュレーションモードを開始し、名前付きプレフィックスリストを設定します。</p> <ul style="list-style-type: none"> <li>プレフィックスリストを作成するには、少なくとも1つの <b>permit</b> 句または <b>deny</b> 句を入力する必要があります。</li> <li>プレフィックスリストのエントリをすべて削除するには、<b>no {ipv4   ipv6} prefix-list name</b> コマンドを使用します。</li> </ul>
ステップ 3	<p><b>[ sequence-number ] remark remark</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 10 remark Deny all routes with a prefix of 10/8 RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 20 deny 10.0.0.0/8 le 32</pre>	<p>(任意) 名前付きのプレフィックス リストに次の <b>permit</b> ステートメントまたは <b>deny</b> ステートメントに関するコメントを書くことができます。</p> <ul style="list-style-type: none"> <li>注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。</li> <li><b>permit</b> ステートメントまたは <b>deny</b> ステートメントの前後どちらにも設定できますが、どちらかの位置に統一することを推奨します。</li> </ul>
ステップ 4	<p><b>[ sequence-number ] {permit   deny} network/length [ge value] [le value] [eq value]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 20 deny 128.0.0.0/8 eq 24</pre>	<p>名前付きプレフィックスリストに許可または拒否の条件を1つ以上指定します。</p> <ul style="list-style-type: none"> <li>この例では、プレフィックスリスト <b>pfx_2</b> の 128.0.0.0/8 の /24 に一致するプレフィックスをすべて拒否します。</li> </ul>
ステップ 5	<p>必要に応じてステップ 4 を繰り返します。 エントリを削除するには、<b>no sequence-number</b> コマンドを使用します。</p>	<p>プレフィックス リストは変更できます。</p>
ステップ 6	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li><b>end</b></li> <li><b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end または RP/0/RSP0/CPU0:router(config-ipv6_pfx)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li><b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</pre> <ul style="list-style-type: none"> <li><b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレー</li> </ul>



	コマンドまたはアクション	目的
		<p>セッションセッションが終了して、ルータが EXEC モードに戻ります。</p> <ul style="list-style-type: none"> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
<p><b>ステップ 7</b></p>	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <b>show prefix-list ipv4</b> [<i>name</i>] [<i>sequence-number</i>]</li> <li>• <b>show prefix-list ipv6</b> [<i>name</i>] [<i>sequence-number</i>] [<i>summary</i>]</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show prefix-list ipv4 pfx_1</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2 summary</pre>	<p>(任意) 現在の IPv4 または IPv6 プレフィックスリストの内容を表示します。</p> <ul style="list-style-type: none"> <li>• 特定のプレフィックスリストの内容を表示するには、<i>name</i> 引数を使用します。</li> <li>• プレフィックスリスト エントリのシーケンス番号を指定するには、<i>sequence-number</i> 引数を使用します。</li> <li>• プレフィックスリストの内容のサマリーを表示するには、<b>summary</b> キーワードを使用します。</li> </ul>
<p><b>ステップ 8</b></p>	<p><b>clear {ipv4   ipv6} prefix-list</b> <i>name</i> [<i>sequence-number</i>]</p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# clear prefix-list ipv4 pfx_1 30</pre>	<p>(任意) IPv4 または IPv6 プレフィックスリストのヒットカウントをクリアします。</p> <p>(注) ヒットカウントは、特定のプレフィックスリスト エントリに一致する数を示す値です。</p>

## 標準アクセスリストの設定

このタスクでは、標準 IPv4 アクセスリストを設定します。

標準アクセスリストでは、照合操作に送信元アドレスを使用します。

## 手順の概要

1. **configure**
2. **ipv4 access-list name**
3. [ *sequence-number* ] **remark remark**
4. [ *sequence-number* ] { **permit** | **deny** } *source* [*source-wildcard*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no sequence-number** コマンドを使用します。
6. 次のいずれかを実行します。
  - end
  - commit
7. **show access-lists [ipv4 | ipv6] [access-list-name hardware {ingress | egress} [interface type interface-path-id] {sequence number | location node-id} | summary [access-list-name] | access-list-name [sequence-number] | maximum [detail] [usage {pfilter location node-id}]**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b>  例： RP/0/RSP0/CPU0:router# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>ipv4 access-list name</b>  例： RP/0/RSP0/CPU0:router# ipv4 access-list acl_1	IPv4 アクセスリスト コンフィギュレーション モードを開始し、アクセスリスト <code>acl_1</code> を設定します。
ステップ 3	[ <i>sequence-number</i> ] <b>remark remark</b>  例： RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out	(任意) 名前付きのアクセスリストに次の <b>permit</b> ステートメントまたは <b>deny</b> ステートメントに関するコメントを書くことができます。 <ul style="list-style-type: none"> <li>• 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。</li> <li>• <b>permit</b> ステートメントまたは <b>deny</b> ステートメントの前後どちらにも設定できますが、どちらかの位置に統一することを推奨します。</li> </ul>

	コマンドまたはアクション	目的
<p>ステップ 4</p>	<p>[ <i>sequence-number</i> ] {<b>permit</b>   <b>deny</b>} <i>source</i> [<i>source-wildcard</i>] [<b>log</b>   <b>log-input</b>]</p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255 または RRP/0/RSP0/CPU0:routerrouter(config-ipv4-acl)# 30 deny 192.168.34.0 0.0.0.255</pre>	<p>パケットの通過またはドロップを決定する許可または拒否の条件を1つ以上指定します。</p> <ul style="list-style-type: none"> <li>• パケットの送信元のネットワークまたはホストの番号を指定するには、<i>source</i> 引数を使用します。</li> <li>• 送信元に適用するワイルドカードビットを指定するには、任意の <i>source-wildcard</i> 引数を使用します。</li> <li>• 任意の <b>log</b> キーワードを指定すると、エントリに一致したパケットに関する情報ログメッセージがコンソールに送信されます。</li> <li>• 任意の <b>log-input</b> キーワードは、ログメッセージに入力インターフェイスも含まれることを除いて、<b>log</b> キーワードと同じように機能します。</li> </ul>
<p>ステップ 5</p>	<p>必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、<b>no sequence-number</b> コマンドを使用します。</p>	<p>アクセスリストは変更できます。</p>
<p>ステップ 6</p>	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# end または RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
ステップ 7	<p><b>show access-lists</b> [<b>ipv4</b>   <b>ipv6</b>] [<i>access-list-name hardware</i> {<b>ingress</b>   <b>egress</b>} [<b>interface type interface-path-id</b>] {<b>sequence number</b>   <b>location node-id</b>}   <b>summary</b> [<i>access-list-name</i>]   <i>access-list-name</i> [<i>sequence-number</i>]   <b>maximum</b> [<b>detail</b>] [<b>usage</b> {<b>pfilter location node-id</b>}]]</p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1</pre>	<p>(任意) 名前付き IPv4 アクセスリストの内容を表示します。</p> <ul style="list-style-type: none"> <li>IPv4 標準アクセスリストの内容は、拡張アクセスリスト形式で表示されます。</li> </ul>

### 次の作業

標準アクセスリストの作成後、それを回線またはインターフェイスに適用する必要があります。アクセスリストを適用する方法については、[アクセスリストの適用](#)、(18 ページ) を参照してください。

## アクセスリストのコピー

このタスクでは、IPv4 または IPv6 アクセスリストをコピーします。

### 手順の概要

1. **copy access-list** {**ipv4** | **ipv6**} **source-acl destination-acl**
2. **show access-lists** {**ipv4** | **ipv6**} [*access-list-name hardware* {**ingress** | **egress**} [**interface type interface-path-id**] {**sequence number** | **location node-id**} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location node-id**}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>copy access-list {ipv4   ipv6} source-acl destination-acl</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# copy ipv6 access-list list-1 list-2</pre>	<p>既存の IPv4 または IPv6 アクセスリストのコピーを作成します。</p> <ul style="list-style-type: none"> <li>• コピーするアクセスリストの名前を指定するには、<i>source-acl</i> 引数を使用します。</li> <li>• 送信元アクセスリストの内容のコピー先を指定するには、<i>destination-acl</i> 引数を使用します。 <ul style="list-style-type: none"> <li>◦ <i>destination-acl</i> 引数は一意の名前である必要があります。アクセスリストに <i>destination-acl</i> 引数名が存在する場合、そのアクセスリストはコピーされません。</li> </ul> </li> </ul>
ステップ 2	<p><b>show access-lists {ipv4   ipv6} [access-list-name hardware {ingress   egress} [interface type interface-path-id] {sequence number   location node-id}   summary [access-list-name]   access-list-name [sequence-number]   maximum [detail] [usage {pfilter location node-id}]]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show access-lists ipv4 list-2</pre>	<p>(任意) 名前付きの IPv4 または IPv6 アクセスリストの内容を表示します。たとえば、コピー先の内容を検証して、宛先アクセスリスト list-2 に送信元アクセスリスト list-1 の情報がすべて含まれていることを確認できます。</p>

## アクセスリストエントリの順序付けとアクセスリストの変更

このタスクでは、名前付きアクセスリストのエントリにシーケンス番号を割り当てる方法と、アクセスリストに対してエントリの追加または削除を行う方法について説明します。アクセスリストを変更することを前提に説明します。アクセスリストの並べ替えは任意です。

## 手順の概要

1. **resequence access-list {ipv4 | ipv6} name [base [increment]]**
2. **configure**
3. **{ipv4 | ipv6} access-list name**
4. 次のいずれかを実行します。
  - `[ sequence-number ] {permit | deny} source source-wildcard destination destination-wildcard [precedence precedence] [dscp dscp] [fragments] [packet-length operator packet-length value] [log | log-input]`
  - `[ sequence-number ] {permit | deny} protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator {port | protocol-port}] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address} [operator {port | protocol-port}] [dscp value] [routing] [authen] [destopts] [fragments] [packet-length operator packet-length value] [log | log-input]`
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no sequence-number** コマンドを使用します。
6. 次のいずれかを実行します。
  - **end**
  - **commit**
7. **show access-lists [ipv4 | ipv6] [access-list-name hardware {ingress | egress} [interface type interface-path-id] {sequence number | location node-id} | summary [access-list-name] | access-list-name [sequence-number] | maximum [detail] [usage {pfilter location node-id}]**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>resequence access-list {ipv4   ipv6} name [base [increment]]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# resequence access-list ipv4 acl_3 20 15</pre>	<p>(任意) 開始シーケンス番号と、シーケンス番号の増分値を使用して、指定した IPv4 または IPv6 アクセスリストを並べ替えます。</p> <ul style="list-style-type: none"> <li>• この例では、<b>acl_3</b> という名前の IPv4 アクセスリストを並べ替えます。開始シーケンス番号は 20、増分は 15 です。増分値を選択しないと、デフォルトの増分値 10 が使用されます。</li> </ul>
ステップ 2	<p><b>configure</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# configure</pre>	<p>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。</p>

	コマンドまたはアクション	目的
<p>ステップ 3</p>	<p>{<b>ipv4</b>   <b>ipv6</b>} <b>access-list name</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2</pre>	<p>IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、名前付きアクセス リストを設定します。</p>
<p>ステップ 4</p>	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>[ <i>sequence-number</i> ] {<b>permit</b>   <b>deny</b>} <i>source source-wildcard destination destination-wildcard</i> [<b>precedence precedence</b>] [<b>dscp dscp</b>] [<b>fragments</b>] [<i>packet-length operator packet-length value</i>] [<b>log</b>   <b>log-input</b>]</li> <li>[ <i>sequence-number</i> ] {<b>permit</b>   <b>deny</b>} <i>protocol</i> {<i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host source-ipv6-address</b>} [<i>operator</i> {<i>port</i>   <i>protocol-port</i>}] {<i>destination-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host destination-ipv6-address</b>} [<i>operator</i> {<i>port</i>   <i>protocol-port</i>}] [<b>dscp value</b>] [<b>routing</b>] [<b>authen</b>] [<b>destop</b>] [<b>fragments</b>] [<i>packet-length operator packet-length value</i>] [<b>log</b>   <b>log-input</b>]</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255 RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000</pre>	<p>IPv4 アクセス リスト <b>acl_1</b> に許可または拒否の条件を 1 つ以上指定します。</p> <ul style="list-style-type: none"> <li>任意の <b>log</b> キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。</li> <li>任意の <b>log-input</b> キーワードは、ログメッセージに入力インターフェイスも含まれることを除いて、<b>log</b> キーワードと同じように機能します。</li> <li>このアクセスリストでは <b>permit</b> ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、<b>deny</b> ステートメントが最初に使用される可能性もあります。</li> </ul> <p>または</p> <p>IPv6 アクセス リスト <b>acl_2</b> に許可または拒否の条件を 1 つ以上指定します。</p> <ul style="list-style-type: none"> <li>IPv6 オプションヘッダーと、ICMP、TCP、UDP などの上位層プロトコルに基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、<b>permit (IPv6)</b> コマンドおよび <b>deny (IPv6)</b> コマンドを参照してください。</li> </ul> <p>(注) どの IPv6 アクセス リストにも最後の一致条件として暗黙の <b>deny ipv6 any any</b> ステートメントがあります。1 つの IPv6 アクセス リストには、暗黙の <b>deny ipv6 any any</b> ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。</p>
<p>ステップ 5</p>	<p>必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加し</p>	<p>アクセス リストは変更できます。</p>

	コマンドまたはアクション	目的
	ます。 エントリを削除するには、 <b>no</b> <i>sequence-number</i> コマンドを使用します。	
ステップ6	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• end</li> <li>• commit</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# end</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <p>Uncommitted changes found, commit them before exiting (yes/no/cancel)?[cancel]:</p> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
ステップ7	<p><b>show access-lists</b> [<b>ipv4</b>   <b>ipv6</b>] [<i>access-list-name</i> <b>hardware</b> {<b>ingress</b>   <b>egress</b>} [<b>interface type interface-path-id</b>] {<b>sequence number</b>   <b>location node-id</b>}   <b>summary</b> [<i>access-list-name</i>]   <i>access-list-name</i> [<i>sequence-number</i>]   <b>maximum</b> [<b>detail</b>] [<b>usage</b> {<b>pfilter location node-id</b>}]]</p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1</pre>	<p>(任意) 名前付きの IPv4 または IPv6 アクセスリストの内容を表示します。</p> <ul style="list-style-type: none"> <li>• 出力をレビューして、アクセスリストに最新情報が含まれていることを確認します。</li> </ul>



### 次の作業

アクセスリストがまだインターフェイスまたは回線に適用されていないか、または他の方法で参照されている場合は、アクセスリストを適用します。アクセスリストを適用する方法については、[アクセスリストの適用](#)、(18 ページ) を参照してください。

## プレフィックスリストのコピー

このタスクでは、IPv4 または IPv6 プレフィックスリストをコピーします。

### 手順の概要

1. `copy prefix-list {ipv4 | ipv6} source-name destination-name`
2. 次のいずれかを実行します。
  - `show prefix-list ipv4 [name] [sequence-number]`
  - `show prefix-list ipv6 [name] [sequence-number] [summary]`

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><code>copy prefix-list {ipv4   ipv6} source-name destination-name</code></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# copy prefix-list ipv6 list_1 list_2</pre>	<p>既存の IPv4 または IPv6 プレフィックスリストのコピーを作成します。</p> <ul style="list-style-type: none"> <li>• コピーするプレフィックスリストの名前を指定するには <code>source-name</code> 引数を使用し、コピー元のプレフィックスリストの内容のコピー先を指定するには、<code>destination-name</code> 引数を使用します。</li> <li>• <code>destination-name</code> 引数は、一意の名前である必要があります。<code>destination-name</code> 引数名がプレフィックスリストに存在する場合、そのプレフィックスリストはコピーされません。</li> </ul>
ステップ 2	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <code>show prefix-list ipv4 [name] [sequence-number]</code></li> <li>• <code>show prefix-list ipv6 [name] [sequence-number] [summary]</code></li> </ul>	<p>(任意) 現在の IPv4 または IPv6 プレフィックスリストの内容を表示します。</p> <ul style="list-style-type: none"> <li>• 出力をレビューして、プレフィックスリスト <code>list_2</code> に <code>list_1</code> のエントリが含まれていることを確認します。</li> </ul>

	コマンドまたはアクション	目的
	例 : <pre>RP/0/RSP0/CPU0:router# show prefix-list ipv6 list_2</pre>	

## プレフィックス リスト エントリの順序付けとプレフィックス リストの変更

このタスクでは、名前付きプレフィックスリストのエントリにシーケンス番号を割り当てる方法と、プレフィックスリストに対してエントリの追加または削除を行う方法について説明します。プレフィックスリストを変更することを前提に説明します。プレフィックスリストの並べ替えは任意です。

はじめる前に



(注) IPv6 プレフィックス リストの並べ替えはサポートされません。

### 手順の概要

1. **resequence prefix-list ipv4** *name* [*base* [*increment*]]
2. **configure**
3. **{ipv4 | ipv6} prefix-list** *name*
4. [*sequence-number*] **{permit | deny}** *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no sequence-number** コマンドを使用します。
6. 次のいずれかを実行します。
  - **end**
  - **commit**
7. 次のいずれかを実行します。
  - **show prefix-list ipv4** [*name*] [*sequence-number*]
  - **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>resequence prefix-list ipv4 name [base [increment]]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# resequence prefix-list ipv4 pfx_1 10 15</pre>	<p>(任意) 開始シーケンス番号と、シーケンス番号の増分値を使用して、指定したIPv4プレフィックスリストを並べ替えます。</p> <ul style="list-style-type: none"> <li>この例では、pfx_1というプレフィックスリストを並べ替えます。開始シーケンス番号は10、増分は15です。</li> </ul>
ステップ 2	<p><b>configure</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# configure</pre>	<p>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。</p>
ステップ 3	<p><b>{ipv4   ipv6} prefix-list name</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2</pre>	<p>IPv4 または IPv6 プレフィックス リスト コンフィギュレーション モードを開始し、名前付きプレフィックス リストを設定します。</p>
ステップ 4	<p><b>[ sequence-number ] {permit   deny} network/length [ge value] [le value] [eq value]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 15 deny 128.0.0.0/8 eq 24</pre>	<p>名前付きプレフィックス リストに許可または拒否の条件を1つ以上指定します。</p>
ステップ 5	<p>必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、<b>no sequence-number</b> コマンドを使用します。</p>	<p>プレフィックス リストは変更できます。</p>
ステップ 6	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>end</li> <li>commit</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end または RP/0/RSP0/CPU0:router(config-ipv6_pfx)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li><b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</pre> <ul style="list-style-type: none"> <li><b>yes</b> と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
<p>ステップ 7</p>	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• <b>show prefix-list ipv4</b> [<i>name</i>] [<i>sequence-number</i>]</li> <li>• <b>show prefix-list ipv6</b> [<i>name</i>] [<i>sequence-number</i>] [<b>summary</b>]</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2</pre>	<p>(任意) 現在の IPv4 または IPv6 プレフィックスリストの内容を表示します。</p> <ul style="list-style-type: none"> <li>• 出力をレビューして、プレフィックスリスト <b>pfx_2</b> に新しい情報がすべて含まれていることを確認します。</li> </ul>

## ACL ベース転送を実装する方法

ここでは、次の手順について説明します。

### セキュリティ ACL での ACL ベース転送の設定

セキュリティ ACL で ACL ベース転送を設定するには、次のタスクを実行します。

### 手順の概要

1. configure
2. **ipv4 access-list** *name*
3. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [*precedence precedence*] [**default nexthop1** [*ipv4 ipv4-address1*] **nexthop2**[*ipv4 ipv4-address2*] **nexthop3**[*ipv4 ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [[**track** *track-name*] [**ttl** *ttl* [*value1* ... *value2*]]]
4. 次のいずれかを実行します。
  - end
  - commit
5. **show access-list ipv4** [[*access-list-name* **hardware** {**ingress** | **egress**}] [**interface** *type interface-path-id*] {**sequence number** | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter** *location node-id*}] ]]

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure  例： RP/0/RSP0/CPU0:router# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>ipv4 access-list</b> <i>name</i>  例： RP/0/RSP0/CPU0:router(config)# ipv4 access-list security-abf-acl	IPv4 アクセス リスト コンフィギュレーション モードを開始し、指定したアクセス リストを設定します。
ステップ 3	[ <i>sequence-number</i> ] <b>permit</b> <i>protocol source source-wildcard destination destination-wildcard</i> [ <b>precedence</b> <i>precedence</i> ] [ <b>default nexthop1</b> [ <i>ipv4 ipv4-address1</i> ] <b>nexthop2</b> [ <i>ipv4 ipv4-address2</i> ] <b>nexthop3</b> [ <i>ipv4 ipv4-address3</i> ]] [ <b>dscp</b> <i>dscp</i> ] [ <b>fragments</b> ] [ <i>packet-length operator packet-length value</i> ] [ <b>log</b>   <b>log-input</b> ] [[ <b>track</b> <i>track-name</i> ] [ <b>ttl</b> <i>ttl</i> [ <i>value1</i> ... <i>value2</i> ]]]  例： RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 10.0.0.0 0.255.255.255 any nexthop 50.1.1.2 RP/0/RSP0/CPU0:router(config-ipv4-acl)# 15 permit ipv4 30.2.1.0 0.0.0.255 any RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit ipv4 30.2.0.0 0.0.255.255 any nexthop 40.1.1.2 RP/0/RSP0/CPU0:router(config-ipv4-acl)# 25 permit ipv4 any any	IPv4 アクセス リストの条件を設定します。設定例では、セキュリティ ACL で ACL ベース転送を設定する方法を示しています。  • <b>nexthop</b> キーワードは、このエントリに指定されたネクスト ホップに転送します。

	コマンドまたはアクション	目的
ステップ 4	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li>• end</li> <li>• commit</li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# end</pre> <p>または</p> <pre>RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)?[cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>
ステップ 5	<p><b>show access-list ipv4</b> <i>[[access-list-name hardware {ingress   egress} [interface type interface-path-id] {sequence number   location node-id}   summary [access-list-name] access-list-name [sequence-number]   maximum [detail] [usage {pfilter location node-id}]]</i></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router# show access-lists ipv4 security-abf-acl</pre>	<p>ACL ソフトウェアに関する情報を表示します。</p>

## IPSLA-OT の実装

ここでは、次の手順について説明します。

- [トラック モードのイネーブル化, \(39 ページ\)](#)

- [トラックタイプの設定](#), (40 ページ)
- [トラッキングタイプの設定 \(回線プロトコル\)](#), (40 ページ)
- [トラックタイプ \(リスト\) の設定](#), (42 ページ)
- [トラッキングタイプ \(ルート\) の設定](#), (43 ページ)
- [トラッキングタイプの設定 \(rtr\)](#), (44 ページ)

## トラックモードのイネーブル化

### 手順の概要

1. `configure`
2. `track track-name`
3. 次のいずれかのコマンドを使用します。
  - `end`
  - `commit`

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>  例： RP/0/RSP0/CPU0:router# <b>configure</b>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<code>track track-name</code>  例： RP/0/RSP0/CPU0:router(config)# <b>track t1</b>	トラック コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかのコマンドを使用します。 <ul style="list-style-type: none"> <li>• <code>end</code></li> <li>• <code>commit</code></li> </ul> 例： RP/0/RSP0/CPU0:router(config)# end	設定変更を保存します。 <ul style="list-style-type: none"> <li>• <code>end</code> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> </ul>

	コマンドまたはアクション	目的
	または  RP/0/RSP0/CPU0:router(config)# commit	<ul style="list-style-type: none"> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## トラックタイプの設定

ネクストホップデバイスの可用性をトラッキングするメカニズムには、さまざまな種類があります。トラッキングタイプには4つのタイプがあり、次のものを使用します。

- 回線プロトコル
- リスト
- ルート
- IPSLA

## トラッキングタイプの設定（回線プロトコル）

回線プロトコルは、オブジェクトトラッカーコンポーネントがトラッキングできるオブジェクトタイプの1つです。このオブジェクトタイプでは、インターフェイスからの状態変化通知をトラッキングするためのオプションを利用できます。インターフェイス状態変化通知に基づいて、トラック状態を UP にするか、DOWN にするかを決定します。

### 手順の概要

1. `configure`
2. `track track-name`
3. `type line-protocol state interface type interface-path-id`
4. 次のいずれかのコマンドを使用します。
  - `end`
  - `commit`



手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>configure</p> <p>例： RP/0/RSP0/CPU0:router# <b>configure</b></p>	<p>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。</p>
ステップ 2	<p><b>track track-name</b></p> <p>例： RP/0/RSP0/CPU0:router (config)# <b>track t1</b></p>	<p>トラック コンフィギュレーション モードを開始します。</p>
ステップ 3	<p><b>type line-protocol state interface type interface-path-id</b></p> <p>例： RP/0/RSP0/CPU0:router (config-track)# <b>type line-protocol state interface tengige 0/4/4/0</b></p>	<p>状態変化通知のためにトラッキングする必要があるインターフェイスを設定します。</p>
ステップ 4	<p>次のいずれかのコマンドを使用します。</p> <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> <p>例： RP/0/RSP0/CPU0:router (config)# end または RP/0/RSP0/CPU0:router (config)# commit</p>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <p>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</p> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## トラックタイプ（リスト）の設定

リストは、ブールオブジェクトタイプです。ブールとは、オブジェクトトラッカーでサポートされているさまざまなオブジェクトタイプの組み合わせに対して、ブール AND 演算またはブール OR 演算を実行する機能のことです。

### 手順の概要

1. `configure`
2. `track track-name`
3. `type list boolean and`
4. 次のいずれかのコマンドを使用します。
  - `end`
  - `commit`

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>  例： RP/0/RSP0/CPU0:router# <code>configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<code>track track-name</code>  例： RP/0/RSP0/CPU0:router (config)# <code>track t1</code>	トラック コンフィギュレーション モードを開始します。
ステップ 3	<code>type list boolean and</code>  例： RP/0/RSP0/CPU0:router (config-track)# <code>type list boolean and</code>	ブール AND 演算またはブール OR 演算を実行できるトラック オブジェクトのリストを設定します。
ステップ 4	次のいずれかのコマンドを使用します。 <ul style="list-style-type: none"> <li>• <code>end</code></li> <li>• <code>commit</code></li> </ul> 例： RP/0/RSP0/CPU0:router (config)# <code>end</code>	設定変更を保存します。 <ul style="list-style-type: none"> <li>• <code>end</code> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <code>yes</code> と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッ</li> </ul>

	コマンドまたはアクション	目的
	または  RP/0/RSP0/CPU0:router (config) # commit	ションが終了して、ルータが EXEC モードに戻ります。  ° <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。  ° <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。  • 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、 <b>commit</b> コマンドを使用します。

## トラッキングタイプ（ルート）の設定

ルートは、ルートオブジェクトタイプです。オブジェクトトラッカーは、FIB通知をトラッキングして、ルート到達可能性およびトラック状態を判断します。

### 手順の概要

1. **configure**
2. **track track-name**
3. **type route reachability**
4. 次のいずれかのコマンドを使用します。
  - **end**
  - **commit**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure  例： RP/0/RSP0/CPU0:router# <b>configure</b>	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。

	コマンドまたはアクション	目的
ステップ 2	<b>track track-name</b>  例： RP/0/RSP0/CPU0:router(config)# <b>track t1</b>	トラック コンフィギュレーション モードを開始します。
ステップ 3	<b>type route reachability</b>  例： RP/0/RSP0/CPU0:router(config-track)# <b>type route reachability</b>	到達可能性状態を動的に学習する必要があるルートを設定します。
ステップ 4	次のいずれかのコマンドを使用します。 <ul style="list-style-type: none"> <li>• <b>end</b></li> <li>• <b>commit</b></li> </ul> 例： RP/0/RSP0/CPU0:router(config)# end または RP/0/RSP0/CPU0:router(config)# commit	設定変更を保存します。 <ul style="list-style-type: none"> <li>• <b>end</b> コマンドを実行すると、変更をコミットするように要求されます。   <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li>◦ <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> </li> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## トラッキングタイプの設定 (rtr)

IPSLA は、`ipsla` オブジェクトタイプです。オブジェクトトラッカーは、`ipsla` 操作の戻りコードをトラッキングして、トラック状態の変化を判断します。

手順の概要

1. `configure`
2. `track track-name`
3. `type rtr ipsla operation id reachability`
4. 次のいずれかのコマンドを使用します。
  - `end`
  - `commit`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>  例： RP/0/RSP0/CPU0:router# <code>configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<code>track track-name</code>  例： RP/0/RSP0/CPU0:router (config)# <code>track t1</code>	トラック コンフィギュレーション モードを開始します。
ステップ 3	<code>type rtr ipsla operation id reachability</code>  例： RP/0/RSP0/CPU0:router# <code>type rtr 100 reachability</code>	到達可能性のためにトラッキングする必要がある ipsla 操作 id を設定します。
ステップ 4	次のいずれかのコマンドを使用します。 <ul style="list-style-type: none"> <li>• <code>end</code></li> <li>• <code>commit</code></li> </ul> 例： RP/0/RSP0/CPU0:router (config)# <code>end</code> または RP/0/RSP0/CPU0:router (config)# <code>commit</code>	設定変更を保存します。 <ul style="list-style-type: none"> <li>• <code>end</code> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>◦ <b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li>◦ <b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>° <b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> <li>• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

## IPv6 ACL 用のピュア ACL ベース転送の設定

### 手順の概要

1. `configure`
2. `{ipv6} access-list name`
3. `[ sequence-number ] permit protocol source source-wildcard destination destination-wildcard [precedence precedence] [dscp dscp] [fragments] [packet-length operator packet-length value] [log | log-input]] [ttl ttl value [value1 ... value2]][default] nexthop1 [ vrf vrf-name1 ] [ipv6 ipv6-address1] [ nexthop2 [ vrf vrf-name2 ] [ipv6 ipv6-address2 ] [nexthop3 [vrf vrf-name3 ] [ipv6ipv6-address3 ]]]`
4. 次のいずれかを実行します。
  - `end`
  - `commit`

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>  例： <code>RP/0/RSP0/CPU0:router# configure</code>	グローバルコンフィギュレーションモードを開始します。
ステップ 2	<code>{ipv6} access-list name</code>  例： <code>RP/0/RSP0/CPU0:router(config)# ipv6 access-list security-abf-acl</code>	IPv6 アクセスリストコンフィギュレーションモードを開始し、指定したアクセスリストを設定します。

	コマンドまたはアクション	目的
<p><b>ステップ 3</b></p>	<p>[ <i>sequence-number</i> ] <b>permit protocol source source-wildcard destination destination-wildcard</b> [ <b>precedence precedence</b> ] [ <b>dscp dscp</b> ] [ <b>fragments</b> ] [ <i>packet-length operator packet-length value</i> ] [ <b>log   log-input</b> ] ] [ <b>ttl ttl value [value1 ... value2]</b> ] [ <b>default</b> ] <b>nexthop1 [ vrf vrf-name1 ] [ ipv6 ipv6-address1 ] [ nexthop2 [ vrf vrf-name2 ] [ ipv6 ipv6-address2 ] [ nexthop3 [ vrf vrf-name3 ] [ ipv6 ipv6-address3 ] ] ]</b></p> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10  permit ipv6 any any default nexthop1 vrf  vrf_A ipv6 11::1 nexthop2 vrf vrf_B ipv6  nexthop3 vrf vrf_C ipv6 33::3</pre>	<p>IPv6 アクセスリストの条件を設定します。設定例では、ACL 用にピュア ACL ベース転送を設定する方法を示しています。</p> <ul style="list-style-type: none"> <li>このエントリに指定されたネクストホップに転送します。</li> </ul>
<p><b>ステップ 4</b></p>	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> <li><b>end</b></li> <li><b>commit</b></li> </ul> <p>例 :</p> <pre>RP/0/RSP0/CPU0:router(config-ipv6-acl)# end または RP/0/RSP0/CPU0:router(config-ipv6-acl)# commit</pre>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> <li><b>end</b> コマンドを実行すると、変更をコミットするように要求されます。</li> </ul> <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)?[cancel]:</pre> <ul style="list-style-type: none"> <li><b>yes</b> と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。</li> <li><b>no</b> と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。</li> <li><b>cancel</b> と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。</li> </ul> <ul style="list-style-type: none"> <li>実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、<b>commit</b> コマンドを使用します。</li> </ul>

# アクセスリストおよびプレフィックスリストの実装の設定例

ここでは、次の設定例について説明します。

## アクセスリストのエントリの並べ替え：例

次に、アクセスリストを並べ替える例を示します。並べ替え後のアクセスリストの開始値は10で、増分値は20です。後続のエントリはユーザ指定の増分値に基づいて並べられています。範囲は1～2147483646です。

シーケンス番号のないエントリが入力されると、デフォルトで、アクセスリストの最後のエントリのシーケンス番号に10を加えたシーケンス番号が割り当てられます。

```

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
  ipv4 access-list acl_1
  end
resequence ipv4 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 permit ip host 172.16.2.2 host 10.3.3.12
110 permit ip host 10.3.3.3 any log
130 permit tcp host 10.3.3.3 host 10.1.2.2
150 permit ip any any

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
  ipv6 access-list acl_1
  end
resequence ipv6 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 Dynamic test permit ip any any
110 permit ip host 172.16.2.2 host 10.3.3.12

```



```
130 permit ip host 10.3.3.3 any log
150 permit tcp host 10.3.3.3 host 10.1.2.2
170 permit ip host 10.3.3.3 any
190 permit ip any any
```

## シーケンス番号を指定したエントリの追加 : 例

次の例では、新しいエントリを IPv4 アクセスリスト `acl_5` に追加しています。

```
ipv4 access-list acl_5
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
10 permit ipv4 host 10.0.0.1 any
20 permit ipv4 host 10.0.0.2 any
configure
ipv4 access-list acl_5
 15 permit 10.5.5.5 0.0.0.255
end
ipv4 access-list acl_5
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
10 permit ipv4 host 10.0.0.1 any
15 permit ipv4 10.5.5.5 0.0.0.255 any
20 permit ipv4 host 10.0.0.2 any
```

## シーケンス番号を指定しないエントリの追加 : 例

次に、シーケンス番号が指定されていないエントリをアクセスリストの末尾に追加する方法を示します。シーケンス番号のないエントリを追加すると、自動的にシーケンス番号が割り当てられ、アクセスリストの末尾に配置されます。デフォルトの増分値は `10` であるため、エントリには、既存のアクセスリストの最後のエントリのシーケンス番号に `10` を加えたシーケンス番号が割り当てられます。

```
configure
ipv4 access-list acl_10
permit 10
.1.1.1 0.0.0.255
permit 10
.2.2.2 0.0.0.255
permit 10
.3.3.3 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
.3.3.0 0.0.0.255 any

configure
ipv4 access-list acl_10
permit 10
.4.4.4 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
```

```
.3.3.0 0.0.0.255 any
40 permit ip 10
.4.4.0 0.0.0.255 any
```

## その他の参考資料

ここでは、アクセスリストおよびプレフィックスリストの実装に関連する資料を示します。

### 関連資料

関連項目	参照先
アクセスリスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference』の「Access List Commands」の章
プレフィックスリスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference』の「Prefix List Commands」の章
端末サービス コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference』の「Terminal Services Commands」の章

### 標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

### MIB

MIB	MIB のリンク
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。 <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a>

**RFC**

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

**シスコのテクニカル サポート**

説明	リンク
シスコのテクニカルサポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	<a href="http://www.cisco.com/en/US/support/index.html">http://www.cisco.com/en/US/support/index.html</a>

