



スコープ、プレフィックス、リンク テンプレートの管理

この章では、スコープ、プレフィックス、およびリンクのテンプレートを設定する方法について説明します。

- [スコープ テンプレートの作成と適用](#) (1 ページ)
- [プレフィックス テンプレートの作成と編集](#) (4 ページ)
- [リンク テンプレートの作成と編集](#) (8 ページ)
- [スコープ テンプレートでの式の使用](#) (11 ページ)
- [プレフィックス テンプレートでの式の使用](#) (16 ページ)
- [リンク テンプレートでの式の使用](#) (20 ページ)

スコープ テンプレートの作成と適用

スコープテンプレートは、特定の共通属性を複数のスコープに適用します。これらの共通属性には、式、ポリシー、アドレス範囲、および式に基づく埋め込みポリシーオプションに基づくスコープ[スコープテンプレートでの式の使用](#) (11 ページ) 名が含まれます (を参照)。

ローカルアドバンスドおよびリージョン Web UI

ローカルクラスタから追加またはプルするスコープテンプレートは、[DHCP スコープ テンプレートの一覧表示] ページに表示されます。そこに移動するには、[デザイン] メニューの `ScopeTemplatesDHCPv4` [サブメニュー] からを選択します。この機能は、地域の中央 `cfg-admin` ロールまたはローカル `ccm-admin` ロールの `dhcp-management` サブロールが割り当てられた管理者のみが使用できます。

スコープテンプレートを明示的に作成するには、[Add Scope Templates スコープテンプレート] ウィンドウをクリックします。[DHCP スコープ テンプレートの追加] ダイアログ ボックスが開き、テンプレート名が表示されます。スコープテンプレートに既存のポリシーを選択することもできます。その他のフィールドには、式の値が必要です (これらのフィールドについて説明する「スコープテンプレートの作成」セクション [Cisco プライム ネットワーク レジストラ 11.0 管理ガイド](#) を参照)。

関連項目

- [スコープ テンプレートでの式の使用 \(11 ページ\)](#)
- [追加のスコープ テンプレート属性 \(2 ページ\)](#)
- [スコープ テンプレートの編集 \(2 ページ\)](#)
- [スコープ テンプレートのスコープへの適用 \(2 ページ\)](#)
- [スコープ テンプレートの複製 \(3 ページ\)](#)

CLI コマンド

`scope-template` 名前 `create`[属性=値..] を使用してスコープ テンプレートを作成します。次に例を示します。

```
nrcmd> scope-template example-scope-template create
```

スコープ テンプレートにポリシーを関連付けることもできます。

```
nrcmd> scope-template example-scope-template set policy=examplepolicy
```

追加のスコープ テンプレート属性

オプションの追加属性は、機能カテゴリに表示されます。各属性の説明を表示するには、属性名をクリックしてヘルプ ウィンドウを開きます。たとえば、スコープの動的 DNS 更新を有効にしたり、メインおよびバックアップの DHCP フェールオーバー サーバーを設定したりできます。

これらのフィールドに入力したら、Add Scope Templateをクリックします。

スコープ テンプレートの編集

スコープ テンプレートを編集するには、[スコープ テンプレート] ウィンドウで名前を選択します。[DHCP スコープ テンプレートの編集] ページは、属性の設定解除機能を除く [スコープ テンプレートの作成と適用 \(1 ページ\)](#) き、[DHCP スコープ テンプレートの追加] ページ (を参照) と基本的に同じです。必要な変更を行い、[保存 (Save)] をクリックします。

CLI で、`scope-template` 名前 `set`属性 を使用してスコープ テンプレート属性を編集します。次に例を示します。

```
nrcmd> scope-template example-scope-template set policy=default
```

スコープ テンプレートのスコープへの適用

スコープ テンプレートは、いくつかの方法でスコープに適用できます。



注意 既存のスコープにスコープテンプレートを適用する際には注意が必要です。テンプレートは、すべてのスコープ属性を独自の属性で上書きします。

ローカルアドバンスド Web UI

- **テンプレートがターゲットに適用される場合**- スコープテンプレートに埋め込みポリシーがある場合、そのテンプレートはスコープにコピーされます。この埋め込みポリシーには、オプションが含まれている場合と、使用できない場合があります。スコープテンプレートの埋め込みポリシー全体が使用されている場合は、スコープ内の既存のオプションが消去されます。スコープテンプレートに埋め込みポリシーがない場合、スコープの埋め込みポリシーは保持されます。次に、スコープテンプレートのオプション式が評価され、オプションがスコープ内の埋め込みポリシー オプションに追加されます (埋め込みポリシーが存在しない場合は、1つが作成されます)。
- **スコープの作成中に、その名前をテンプレートから派生させる** — [リスト/DHCP スコープテンプレートの [スコープテンプレートでの式の使用 \(11 ページ\)](#) 追加] ページでスコープテンプレートのスコープを設定する (「DHCP スコープテンプレートの追加」ページを参照) 場合は、[一覧/追加 DHCP スコープ] ページでスコープを追加するときに、スコープの名前を省略し、サブネットとマスクを追加してから、[テンプレート] ドロップダウンリストからスコープテンプレートを選択します。[DHCPスコープの追加] をクリックすると、スコープ名式から合成された名前で作成されます。テンプレートにスコープ名式を設定せず、スコープの名前を指定せずにスコープに適用すると、エラーが発生します。(基本モードでは、この機能は提供されません)。
- **名前付きスコープの作成後**- [DHCP スコープの編集] ページで、下までスクロールして [テンプレートの適用] ボタンを見つけます。ドロップダウンリストから事前設定テンプレートを選択し、ボタンをクリックします。次に、[保存 (Save)] をクリックします。(テンプレート属性がスコープの既存の属性を上書きするという以前の警告に注意してください)。

CLI コマンド

スコープの作成中にテンプレートをスコープに適用するには、**スコープ名作成アドレスマスク [テンプレート=テンプレート名] [属性=value .]** を使用します。次に例を示します。

```
nrcmd> scope example-scope create 192.168.50.0 24 template=example-scope-template
```

スコープの作成中にテンプレートからスコープ名を取得するには、**スコープテンプレート名適用先 {すべて | scope1、スコープ 2、...}** を使用します。次に例を示します。

```
nrcmd> scope-template example-scope-template apply-to examplescope-1,examplescope-2
```

スコープテンプレートの複製

CLIでは、`scope-template clone-name create clone=template` を使用して、既存のテンプレートからスコープテンプレートを複製し、そのクローンを調整することもできます。次に例を示します。

```
nrcmd> scope-template cloned-template create clone=example-scope-template-1
ping-timeout=200
```

プレフィックス テンプレートの作成と編集

定義済みのテンプレートから接頭辞を作成できます。プレフィックステンプレートに設定できる属性は次のとおりです(式の構文については、を参照[プレフィックス テンプレートでの式の使用 \(16 ページ\)](#) してください)。

- **name** : プレフィックステンプレートのユーザー割り当て名。
- **description** : プレフィックステンプレートの説明テキスト。
- **dhcp** タイプ : プレフィックスのアドレス割り当てを DHCP が管理する方法を定義します。
 - **dhcp** (プリセット値) : ステートフルアドレス割り当てにプレフィックスを使用します。
 - **stateless**—ステートレス オプションの設定にプレフィックスを使用します。
 - **プレフィックス-委任-プレフィックス**のプレフィックスを使用して、プレフィックスの委任します。
 - **infrastructure**—プレフィックスを使用して、プレフィックスにアドレス プールがない場合に、クライアントアドレスをリンクにマップします。
 - **parent** : プレフィックスは DHCP によって使用されません。子プレフィックスをグループ化するためにテナナオブジェクトとして使用されます。
- **policy** : クライアントに返信するときに使用する共有ポリシー。
- **owner** : 名前で参照されるこのプレフィックスの所有者。
- **region** : 名前で参照されるこのプレフィックスのリージョン。
- **prefix-name-expr** : 作成されたプレフィックスの名前に使用する文字列値に対して評価する式。たとえば、**prefix-name-expr** を (`concat "CM-" prefix`) と定義した場合に **CM-** が先頭に付加したプレフィックス名を付けることができます。CLI では、ファイルに式を含め、そのファイルを指定します。


```
> type prefix-name.txt
(concat "CM-" prefix)
```

```
nrcmd> prefix-template ex-template create prefix-name-expr=@prefix-name.txt
```
- **prefix-description-expr** : テンプレートを使用するときに作成されるプレフィックスの説明に適用する文字列値に対して評価する式。
- **range-expr** : アドレス範囲を作成する IPv6 プレフィックス値に対して評価する式。CLI では、ファイル参照を使用する必要があります。次に例を示します。

```
> type subprefix-expr.txt
(create-prefix-range 1 0x1)

nrcmd> prefix-template ex-template set range-expr=@subprefix-expr.txt
```

- **options-expr** : 作成する組み込みポリシーオプションに対して評価する式。(複数のlistオプションを作成するには、この関数を使用します。
- **allocation-algorithms** : クライアントにリースする新しいアドレスやプレフィックスを選択する際にサーバーが使用する1つ以上のアルゴリズム。使用可能なアルゴリズムは次のとおりです。
 - **client-request** (プリセット値はオフ) : クライアントが要求したリースをサーバーが使用するかどうかを制御します。
 - **reservation** (プリセット値はon) : クライアントで利用可能な予約をサーバーが使用するかどうかを制御します。
 - **extension** (プリセット値はオン) : クライアントに対してアドレスまたはプレフィックスを生成するために、**generate-lease** 拡張ポイントにアタッチした拡張機能をサーバーが呼び出すかどうかを制御します。DHCPv6 フェールオーバーでリースの生成拡張機能ポイントを使用する場合、サーバーは、拡張が返すアドレスまたはデリゲートされたプレフィックスを使用し、ランダムに生成されたアドレスと同様に、このアドレスまたはプレフィックスに対してハッシュを実行しません。拡張機能がアドレスまたはデリゲートされたプレフィックスを生成するアルゴリズムメソッドを使用している場合、拡張機能はフェールオーバーに対応する必要があります(拡張機能は、フェールオーバー構成が有効になっているかどうか、およびフェールオーバーサーバーの役割を判断できます)。拡張機能の詳細については、を[拡張機能の使用](#)参照してください。
 - **interface-identifier** (プリセット値はoff) : アドレスを生成するためにサーバーがクライアント (link-local) アドレスから **interface-identifier** を使用するかどうかを制御します。一時アドレスとプレフィックスの委任では無視されます。
 - **random** (プリセット値はon) : サーバーが、RFC 3041 アルゴリズムを使用してアドレスを生成するかどうかを制御します。プレフィックスの委任では無視されます。
 - **best-fit** (プリセット値はon) : 使用可能で最も適切なプレフィックスをサーバーが最初に委任するかどうかを制御します。アドレスでは無視されます。

サーバーがクライアントに割り当てるアドレスが必要な場合、クライアント要求、予約、拡張、インターフェイス識別子、およびランダムなアドレスが見つかるまで、フラグは次の順序で処理されます。サーバーは、クライアントにプレフィックスをデリゲートする必要がある場合、クライアント要求、予約、拡張、最適なプレフィックスが見つかるまで、フラグを次の順序で処理します。

- **restrict-to-reservations** : クライアント (またはリース) 予約に対してプレフィックスが制限されるかどうかを制御します。

- **max-leases** : プレフィックスで許可されている、予約されていないリースの最大数。新しいリースを作成する必要がある場合、サーバーは制限を超えていない場合のみ作成します。制限を超えると、サーバーはクライアントに新しいリースを作成したり、新しいリースを提供したりできません。SNMP トラップも有効にした場合、**max-leases**値は使用済みアドレスと使用可能なアドレスのパーセンテージも計算します。



注 SNMP アドレス トラップが意味のある結果を返すことができるように、最大リース値を予想される最大値に設定してください。

- **ignore-declines** : IPv6 アドレスを参照する DHCPv6 DECLINE メッセージまたは、このプレフィックスからの委任されたプレフィックスにサーバーが応答するかどうかを制御します。有効にすると、サーバーはこのプレフィックスのリースに関するすべての拒否を無視します。無効(プリセット値)または未設定の場合、サーバーは、クライアントにリースされている場合、DECLINE メッセージで要求されるすべてのアドレスまたは委任されたプレフィックスを UNAVAILABLE に設定します。
- **deactivated** : プレフィックスがクライアントへのリースを延長するかどうかを制御します。非アクティブ化されたプレフィックスは、リースをクライアントに拡張せず、範囲内のすべてのアドレスを個別に非アクティブ化されたかのように扱います。プリセット値は **false** (アクティブ化) です。
- **expiration-time** : プレフィックスの有効期限が切れる日時。この日時以降、サーバーは新しいリースを許可せず、このプレフィックスから既存のリースを更新することもしません。[平日]月の日 hh:mm[:ss]年"の形式"で値を入力します。たとえば"Dec31, 23:59"などです。2006" 有効期限が切れる理由は、ネットワークの番号変更イベントをサポートするためです。一般的な考え方は、新しいプレフィックスが追加され、古いものは、有効期限の後に、いつか取り除かれます。クライアントには、両方のプレフィックスにリースが与えられます。有効期限に達する前に、サーバーは、構成された有効な有効期間が経過すると、新しいクライアントにリースを自動的に与えることを停止します。この時点では、新しいクライアントはプレフィックスのリースを取得しません。既存のクライアントは引き続き既存のリースを使用できますが、有効期間が短くなります(優先および有効)。優先と有効の間のデルタは常に維持されます。したがって、優先が1日で有効な2日の場合、新しいクライアントは有効期限の2日前にリースの取得を停止し、既存のクライアントは、1日未満の優先有効期間と2日を超える有効な有効期間でリースを更新し続けます。有効期限の1日前に、クライアントは0の優先有効期間を取得します。
- **free-address-config** : プレフィックス上の予期しない空きアドレスイベントをキャプチャするトラップ。
- **reverse-zone-prefix-length** : ip6.arpa 更新の逆引きゾーンのプレフィックス長 (詳細については、[DNS 更新のための逆引きゾーンの決定](#)を参照してください)。
- **max-pd-balancing-length** : **prefix-delegation** プレフィックスのバランスをとる際にフェールオーバー プール バランシングが考慮する、**prefix-delegation** プレフィックスの最大長を制

御します。既定値は 64 で、プレフィックスの委任で許可されている最長プレフィックス長を超えることはありません。

- **selection-tags** : プレフィックスに関連付けられた選択タグのリスト。
- **allocation-group** : プレフィックスが属している割り当てグループ。
- **allocation-group-priority** : 同じ割り当てグループ内の他のプレフィックスに対するこのプレフィックスの優先順位。デフォルト値は 0 です。
- **range-start-expr** : プレフィックスの **range-start** に対して評価する式を定義します。
- **range-end-expr** : プレフィックスの **range-end** に対して評価する式を定義します。
- **embedded-policy** : 組み込まれたポリシー。テンプレートが適用されると、プレフィックスに埋め込まれたポリシー全体が置き換えられます。

ローカルアドバンスドおよびリージョン Web UI

- ステップ 1** Design メニューで、DHCPv6 サブメニューから **Prefix Templates** を選択します。[DHCP v6 プレフィックス テンプレートの一覧/追加] ページに、既存のテンプレートが表示されます。
- ステップ 2** [プレフィックス テンプレート] **Templates** ウィンドウのアイコンをクリックして、[プレフィックス テンプレートの追加] ダイアログ ボックスを開きます。 **Add Prefix**
- ステップ 3** プレフィックス テンプレート名を入力し **Add Prefix Template**、 をクリックします。
- ステップ 4** プレフィックス テンプレートを編集するには、[プレフィックス テンプレート] ウィンドウで名前を選択します。式を必要とするテンプレートの属性を設定し、式を追加 [プレフィックス テンプレートでの式の使用 \(16 ページ\)](#) します (「」を参照)。
- ステップ 5** [DHCP v6 プレフィックス テンプレートの編集] ページで、選択タグの追加、グループの割り当て、優先順位の設定などのテンプレート **Save** 属性を編集し、 をクリックします。
- ステップ 6** 地域 Web UI では、レプリカプレフィックス テンプレートをプルしたり、テンプレートをローカルクラスターにプッシュしたりできます。
 - クリック **Pull Data** すると、[プルするレプリカプレフィックス テンプレート データの選択] ページが開きます。クラスターのプルモード (確認、置換または完全一致) を選択し、 **Pull All Prefix Templates** をクリックします。[レポートプル DHCPv6 プレフィックス テンプレート OK] ページで、 をクリックします。
 - 特定 **Push** のテンプレート (または **Push All**) をクリックして、[ローカルクラスターへのデータのプッシュ] ページを開きます。データ同期モード (確実、置換、または正確) を選択し、目的のクラスターを [選択] テーブルに移動して **Push Data to Clusters**、 をクリックします。
 - クリック **Reclaim** して [プレフィックス テンプレートの再利用] ページを開きます。[選択済み (Selected)] テーブルに目的のクラスターを移動させ、 **Reclaim Data from Clusters** をクリックします。

CLI コマンド

プレフィックス テンプレートを作成するには、`prefix-template` 名前 `create`[属性=値..] を使用します。次に例を示します。

```
nrcmd> prefix-template example-prefix-template create [attribute=value]
```

前述の属性は通常の方法で設定および有効化でき、接頭辞テンプレートの表示とリスト表示を行うことができます。また、次の点に注意してください。

- プレフィックステンプレートのクローンを作成`prefix-template`するには、名前を`createclone=`使用します。
- テンプレートを 1 つ以上のプレフィックスに適用するには`prefix-template`、`name apply-to {all|プレフィックス[,プレフィックス,..]}`
- プレフィックス テンプレートには、埋め込みポリシー オブジェクトが含まれます。プレフィックス テンプレート ポリシー CLI コマンドおよび Web UI は、プレフィックス テンプレート ページに埋め込まれたポリシーをサポートします。
- 地域クラスターに接続すると、次のプル、プッシュ、および再利用のコマンドを使用できます。プッシュおよび再利用の場合は、クラスターのリストまたは「すべて」を指定できます。
 - <名前|`prefix-template`すべて>プル<確認する|置き換える|正確な>クラスター名[-レポートのみ|-レポート]
 - <名前|`prefix-template`すべて>プッシュ<確認する|置き換える|正確な>クラスターリスト[-レポートのみ|-レポート]
 - 名前再利用クラスターリスト [-レポートのみ | `prefix-template` -レポート]

リンク テンプレートの作成と編集

定義済みテンプレートからリンクを作成できます。リンク テンプレートに設定できる属性は次のとおりです (式の構文については、[を参照してくださいリンク テンプレートでの式の使用 \(20 ページ\)](#))。

- `name` : リンクテンプレートのユーザー割り当て名。
- `description` : リンクテンプレート自体の説明。
- `policy` : クライアントに応答する際にリンクに適用される共有ポリシー。
- `owner` : リンクの所有者。
- `region` : このリンクのリージョン。
- `link-name-expr` : テンプレートが適用された後にリンクの名前を定義する式。
- `link-description-expr` : 適用された後にリンクに関する説明を定義する式。

- **prefix-expr** : テンプレートが適用された後に、関連付けられたプレフィックスのリストを作成する式。たとえば、この式を含むファイル@link-prefix-expr.txtを指すようにprefix-exprを定義した場合に、プレフィックスを作成するように指定できます (cm-prefix、cpe-address-prefix、および cpe-pd-prefix テンプレートが存在すると仮定します)。

```
(list
  (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
  (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
  (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1))
)
```

- **options-expr** : リンクを使用して作成する組み込みポリシーのオプションのリストを定義する式。
- **free-address-config** : このリンク上の予期しない空きアドレスイベントをキャプチャするトラップ。
- **type** : リンクのタイプ (トポロジ、ロケーション非依存、ユニバーサル) 。
- **group-name** : リンクが属しているリンクのグループ。
- **embedded-policy** : 組み込まれたポリシー。テンプレートが適用されると、リンク内の埋め込みポリシー全体が置き換えられます。

ローカル アドバンスドおよびリージョン アドバンスド Web UI

- ステップ 1** メニューからDesignDHCPv6 サブメニューの下で選択Link Templatesします。[DHCP v6 リンク テンプレートの一覧/追加] ページが表示されます。ページには既存のテンプレートが表示されます。
- ステップ 2** [リンク テンプレート]ペインTemplatesのアイコンをクリックして、[リンク テンプレートの追加]ダイアログ ボックスを開きます。 Add Link
- ステップ 3** リンク テンプレート名を入力し、[リンク テンプレートの追加] をクリックします。
- ステップ 4** オプションの説明を入力し、オプションでドロップダウン リストから事前設定済みポリシーを選択します。
- ステップ 5** リンク名-expr、リンク記述-expr、プレフィックス-expr、またはオプション-exprフィールド属性の式を追加します ([リンク テンプレートでの式の使用 \(20 ページ\)](#) を参照)。
- ステップ 6** リンク テンプレートが [接頭辞の安定性] の場合は、リンクの種類 (種類) を選択し、リンク グループ名 (group-name) を指定します。これらの属性は、[DHCP v6 リンク テンプレートの追加] ページの [プレフィックス安定性プレフィックス安定性ブロック] に表示されます (リンクタイプとリンクグループの詳細についてはを参照してください)。
- ステップ 7** Save をクリックします。
- ステップ 8** 地域 Web UI では、レプリカ リンク テンプレートをプルしたり、ローカル クラスターにテンプレートをプッシュしたり、リンク テンプレートを再利用したりできます。
 - Pull クリック Data すると、[プルするレプリカ リンク テンプレート データの選択] ページが開きます。クラスターのプルモードを選択し (確認、置換、または正確) Pull All Link Templates をクリックします。[レポート プル DHCPv6 リンク テンプレート OK] ページで、 をクリックします。

- 特定Pushのテンプレート (またはPush All) をクリックして、[ローカルクラスタへのデータのプッシュ] ページを開きます。データ同期モード(確実、置換、または正確)を選択し、目的のクラスタを[選択] テーブルに移動してPush Data to Clusters、 をクリックします。
- クリックReclaimして[リンク テンプレートの再利用] ページを開きます。[選択済み (Selected)] テーブルに目的のクラスタを移動させ、Reclaim Data from Clusters をクリックします。

CLI コマンド

リンク テンプレートを作成するには、link-template 名前 create[属性=値..] を使用します。次に例を示します。

```
nrcmd> link-template example-link-template create [attribute=value]
```

上記の式設定属性は通常の方法で設定および有効化でき、リンクテンプレートの表示とリスト表示が可能です。たとえば、リンク テンプレートのプレフィックス式を設定するには、次のファイル定義とファイルへのポインターを使用します (cm-prefix、cpe-address-prefix、および cpe-pd-prefix テンプレートが存在すると仮定します)。

```
> type link-prefix-expr.txt
(list (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
 (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
 (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1)) )
```

```
nrcmd> link-template example-link-template set prefix-expr=@link-prefix-expr.txt
```

また、次の点に注意してください。

- リンク テンプレートを複製するには、link-template 名前 create clone=を使用します。
- 1つまたは複数のリンクにテンプレートを適用するにはlink-template、名前apply-to{all | リンク,[リンク,..] }link-template名前apply-toリンク[prefix] を使用してプレフィックスを作成できますが、指定されたリンクは1つのみになります。
- リンクテンプレートには、埋め込みポリシーオブジェクトが含まれています。リンクテンプレートポリシー CLI コマンドおよび Web UI は、リンクテンプレートページの埋め込みポリシーをサポートします。
- 地域クラスターに接続すると、次のプル、プッシュ、および再利用のコマンドを使用できます。プッシュおよび再使用の場合は、クラスターのリストまたは「すべて」を指定できます。
 - <名前| link-templateすべて>プル<確認する |置き換える|正確な>クラスター名[-レポートのみ|-レポート]
 - <名前| link-templateすべて>プッシュ<確認する |置き換える|正確な>クラスターリスト [-レポートのみ|-レポート]
 - 名前再利用クラスターリスト [-レポートのみ | link-template -レポート]

スコープ テンプレートでの式の使用

スコープ テンプレートで式を指定して、スコープを作成するときに、スコープ名、IP アドレス範囲、および埋め込みオプションを動的に作成できます。式には、コンテキスト変数と操作を含めることができます。



- (注) 式は DHCP 拡張と同じではありません。式は、クライアント ID の作成やクライアントの検索に一般的に使用されます。拡張 ([拡張ポイントの使用](#)を参照) は、要求パケットまたは応答パケットを変更するために使用されます。既に範囲が定義されているスコープにテンプレートを適用すると、そのスコープテンプレートのアドレス範囲式は評価されません。

次の表は、スコープ式関数の一覧です。これらの関数では大文字と小文字が区別されないことに注意してください。

表 1: スコープ テンプレート式関数

式関数	説明
Context Variables	
bcast-addr	サブネット内のブロードキャスト アドレスから導出されます (192.168.50.255 など)。任意の式フィールドで使用します。
first-addr	192.168.50.64/26 の最初のアドレスなど、サブネットの最初のアドレスから派生したアドレスは 192.168.50.65 です。式フィールドで使用します。
last-addr	192.168.50.64/26 の最後のアドレスなど、サブネットの最後のアドレスから派生した 192.168.50.127 です。式フィールドで使用します。
mask-addr	サブネット内のネットワーク マスク アドレス (255.255.255.0 など) から派生します。式フィールドで使用します。
mask-count	24 など、サブネットのネットワーク アドレスのビット数から派生します。[スコープ名式] フィールドまたは [埋め込みポリシー オプションの式] フィールドで使用します。
naddrs	サブネット内の IP アドレスの数 (255 など) から派生します。[スコープ名の式] フィールドで使用します。
nhosts	サブネット内の使用できるホストの派生数 (254 など)。式フィールドで使用します。
subnet	192.168.50.0/24 などのサブネットの IP アドレスとマスクから派生します。スコープ名の式または埋め込みポリシーオプション式フィールドで使用します。

式関数	説明
subnet-addr	192.168.50.0 などのサブネット アドレスから派生します。式フィールドで使用します。
template.attribute	template.ping タイムアウトなどのスコープテンプレートの属性。[埋め込みポリシー オプション式] フィールドで使用します。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。
this.attribute	スコープの属性。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。
Arithmetic (符号なし整数Operations引数のみ)	
(+ arg1 arg2)	(+ 2 3) などの 2 つの引数値を加算します。
(- arg1 arg2)	100 として定義された ping タイムアウト(-の場合は、テンプレート.ping タイムアウト10)など、最初の引数の値から 2 番目の引数値を減算すると 90 になります。
(* arg1 arg2)	2 つの引数の値を乗算します。
(/ arg1 arg2)	最初の引数の値を 2 番目の引数の値で除算します (0 にすることはできません)。
Concatenation Operation	
(concat arg1 ... argn)	引数を文字列に連結し、[スコープ名の式] フィールドで使用します。例: サブネット=192.168.50.0/24 および template.ping-timeout=100: <code>(concat "ISP-" subnet)</code> --> ISP-192.168.50.0/24 <code>(concat subnet "-" (+ template.ping-timeout 10))</code> --> 192.168.50.0/24-110 <code>(concat "ISP-" subnet "-" (+ template.ping-timeout 10))</code> --> ISP-192.168.50.0/24-110 スコープ名の式の例 (15 ページ) も参照してください。
Create Option Operation	

式関数	説明
(create-option opt val)	<p>[埋め込みポリシー オプション式] フィールドで create-option を使用して、スコープの新しい DHCP オプションを作成します。最初の引数には、オプション番号または名前を表す整数または文字列を指定できます。2 番目の引数は、オプションに値を与える文字列または BLOB です。</p> <p>カスタム定義オプションおよび不明オプションを指定することもできます。未定義のオプションの場合は、オプション番号を指定し、データを (BLOB データとして) 使用する必要があります。データが文字列の場合、文字列は、データが数値またはアドレスである場合は、その文字列を使用します。</p> <p>次に、例を示します。</p> <pre>(list (create-option "domain-name" "example.com") (create-option 3 "10.10.10.1")) (create-option "routers" "10.10.10.1,10.10.10.2,10.10.10.3") (create-option "routers" (create-ipaddr subnet 10))</pre> <p>埋め込みポリシー オプション式の例 (16 ページ) も参照してください。</p>
Create Vendor Option Operation	
(create-vendor-option set-name opt val)	<p>[組み込みポリシー オプション式] フィールドの [ベンダー作成オプション] を使用して、DHCP ベンダー オプションを作成します。set-name は、ベンダー オプションに設定されているオプション定義を指定します。opt は、セット内のベンダー・オプションを識別するリテラル・ストリングまたは整数にすることができます。val はオプション値を表します。</p> <p>次に、例を示します。</p> <pre>(list (create-option "routers" (create-ipaddr subnet 1)) (create-vendor-option "dhcp-cablelabs-config" 125 (concat "(tftp-servers 2 " (create-ipaddr subnet 2)"))))</pre>
Create Range Operation	

式関数	説明
(create-range start end)	<p>[範囲式] フィールドでこの操作を使用します。スコープの IP アドレス範囲を作成します。最初の引数はアドレス範囲の先頭で、整数または IP アドレス文字列を指定できます。2 番目の引数は範囲の終わり、整数または IP アドレスの文字列を指定できます。範囲内のマスク (0 と 255 など/24 サブネット) によって決定されるローカル ホストまたはブロードキャスト アドレスを含めないでください。検証では、範囲がテンプレートで定義されているサブネット内に存在し、最初の引数値が 2 番目の値より小さくなくてはなりません。整数値は、指定されたサブネット内のアドレスの位置を決定します。</p> <p>例 (サブネット= 192.168.50.0/26):</p> <pre>(create-range "192.168.50.65" "192.168.50.74") --> 192.168.50.65 - 192.168.50.74 (create-range 1 10) --> 192.168.50.65 - 192.168.50.74</pre> <p>範囲の式の例 (15 ページ) も参照してください。</p>
Create IP Operation	
(create-ipaddr ネット ホスト)	<p>この操作は、埋め込みポリシーオプション式または範囲式フィールドで使用します。IP アドレス文字列を作成します。net 引数は文字列または変数です。ホスト引数は整数です。</p> <p>例:</p> <pre>(create-ipaddr subnet 4)</pre>
List Operation	
(list oper1 ... opern)	<p>引数はすべて、作成オプションまたは範囲の作成操作である必要があります。ネスティングはサポートされていません。</p> <p>例:</p> <pre>(list (create-option "routers" "10.10.10.1") (create-option "domain-name" "example.com")) (list (create-range 1 5) (create-range 10 20))</pre>

ローカルアドバンスドおよびリージョン Web UI

[DHCP スコープテンプレートの追加] ページには、式を指定する必要がある次の 3 つのフィールドがあります。

- —Scope 文字列を返Name ず Expression 必要があります
- Range —IP アドレスを返Expression する必要があります
- Embedded Policy Option Expression : 要件なし

CLI コマンド

次の `scope-template` コマンド属性を使用します。

- `scope-name`
- `ranges-exp`
- `options-exp`

スコープ名の式の例

テンプレートが "ISP-" で始まり、スコープのサブネットと ping タイムアウト値の派生値が続くように、式を設定できます。[スコープ名の式] フィールドでは、次の式を使用します。

```
(concat "ISP-" subnet "-" (+ template.ping-timeout 10))
```

式の例の要素は次のとおりです。

- `(concat ...)`- 連結操作は、次のすべての値を 1 つの値に連結します。
- "ISP-"- スコープ名の開始に使用する文字列。
- `subnet`- スコープに定義された既存のサブネットを使用することを示すキーワード変数。
- "-"- 値を作成するために、このハイフンを含むように指定します。
- `-`- スコープの ping タイムアウト プロパティ値を数値 10 に追加することを示します。 (+ `template.ping-timeout 10`)

スコープサブネットが 192.168.50.0/24 で、ping タイムアウト値 100 の場合、結果として作成されるスコープ名は次のようになります。

```
ISP-192.168.50.0/24-110
```

範囲の式の例

テンプレートがスコープの特定のアドレス範囲のみを構築するように式を設定することもできます。実際の開始アドレスと終了アドレスを明示的に指定することも、サブネットに対して相対的に指定することもできます。[範囲式] フィールドで相対範囲を要求する方法は 2 つあります。

```
(create-range first-addr last-addr)  
(create-range 1 10)
```

最初 `create-range` の操作では、サブネット内の最初から最後に使用できるアドレスに基づいてアドレス範囲が作成されます。たとえば、192.168.50.0/24 サブネットの場合、アドレス範囲は 192.168.50.1 から 192.168.50.254 になります。2 番目の操作では、完全な IP アドレスではなく整数を指定するため、サブネットに対する範囲はマスクに基づいて相対的になります。テンプレートがサブネットを 192.168.50.0/26 と検出した場合、このサブネットの最初から 10 番目のアドレスを 192.168.50.65 から 192.168.50.74 とします。

CLI で範囲式を設定するには、ファイルに式を配置し、次のようなコマンドを使用する必要があります。

```
nrcmd> scope-template example-template set ranges-expr=@ file
```

ここで、file は式を使用して作成したファイルの名前です。

埋め込みポリシー オプション式の例

DHCP サーバーは、スコープの割り当てられた名前付きポリシーを参照する前に、そのポリシーを参照するため、埋め込みポリシーは重要です。通常、これはスコープに DHCP オプションを設定する場所です。テンプレートがスコープ埋め込みポリシーの DHCP オプションを構成するように式を設定することもできます。次に例を示します。

```
(create-option "domain-name" "example.com")
(create-option 3 "10.10.10.1")
(create-option "routers" (create-ipaddr subnet 10))
```

最初 create-option の操作では、値 example.com をスコープの domain-name オプションに関連付けます。2 番目の操作では、アドレス 10.10.10.1 がルーターオプション (番号 3) に関連付けられます。3 番目の操作では、サブネットの 10 番目のアドレスに基づいて、ルーターオプションの IP アドレスが作成されます。

CLI でポリシー・オプション式を設定するには、その式をファイルに入れ、次のようなコマンドを使用する必要があります。

```
nrcmd> scope-template example-template set options-expr=@ file
```

file は、式で作成したファイルの名前です。



- (注) 埋め込みスペースや引用符などの特殊文字が原因で、CLI コマンドラインで式を直接指定しようとする場合、失敗する可能性があります。CLI@コマンド・パーサーに関する潜在的な問題を回避するため、ファイル構文を使用します。しかし、WebUI は@file構文をサポートしていません。複雑な式は、Web UI に直接入力できます。

プレフィックス テンプレートでの式の使用

プレフィックス テンプレートで式を指定すると、プレフィックス名、IP アドレス範囲、および埋め込みオプションを作成して、プレフィックスを作成できます。式には、コンテキスト変数と操作を含めることができます。



- (注) 式は DHCP 拡張と同じではありません。式は、クライアント ID の作成やクライアントの検索に一般的に使用されます。拡張 ([拡張ポイントの使用](#)を参照) は、要求パケットまたは応答パケットを変更するために使用されます。

テンプレートがプレフィックスに適用される場合、prefix-template にポリシーが埋め込まれている場合、そのテンプレートはプレフィックスにコピーされます。この埋め込みポリシーには、オプションが含まれている場合と含まれていない場合があります。プレフィックステンプレートの埋め込みポリシー全体が使用されている場合は、プレフィックス内の既存のオプション

ンが消去されます。prefix-template に埋め込みポリシーがない場合、プレフィックスの埋め込みポリシーは保持されます。次に、prefix-template のオプション式が評価され、オプションがプレフィックスの埋め込みポリシー オプションに追加されます (埋め込みポリシーが存在しない場合は、そのオプションが作成されます)。

次の表は、接頭辞テンプレートの定義済み変数を示し、演算子を示しています。これらの変数と演算子は、大文字と小文字が区別されないことに注意してください。

表 2: プレフィックス テンプレート式 定義済み変数

定義済み変数	説明
prefix	リンクにリンク テンプレートを適用する場合はテンプレート ルートプレフィックスに基づくネットワーク番号と長さ、プレフィックステンプレートをプレフィックスに適用する場合はプレフィックスアドレスに基づくネットワーク番号と長さ。
vpn	プレフィックスの VPN。
prefix-addr	プレフィックスのアドレス部分。
prefix-length	プレフィックス アドレス ビットの数。
mask-length	プレフィックス マスク ビットの数。
template.attribute	プレフィックス テンプレートの属性。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。
this.attribute	プレフィックスのリンク名の this.link などのプレフィックスの属性。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。

表 3: 接頭辞テンプレート式演算子

式の演算子	説明
Arithmetic (符号なし整数Operators引数のみ)	
(+ arg1 arg2)	(+ 2 3) などの 2 つの引数値を加算します。
(- arg1 arg2)	100 として定義された ping タイムアウト(-の場合は、テンプレート ping タイムアウト10)など、最初の引数の値から 2 番目の引数値を減算すると 90 になります。
(* arg1 arg2)	2 つの引数の値を乗算します。
(/ arg1 arg2)	最初の引数の値を 2 番目の引数の値で除算します (0 にすることはできません)。

式の演算子	説明
(% arg1 arg2)	剰余算術演算子は、最初の引数の結果の残りの部分を 2 番目の引数で除算した値を求めます。
Concatenation Operator	
(concat arg1 ... argn)	引数を文字列に連結します。
List Operator	
(list oper1 ... opern)	オプションリストまたはプレフィックスのリストを作成します。プレフィックスに対して複数のオプションが必要な場合は必須です。すべての引数は、 <code>create-v6-option</code> または <code>create-prefix-range</code> 操作である必要があります。ネスティングはサポートされてません。
Create IP Operator	
(create-prefix-addrプレフィックス名インターフェイス ID)	プレフィックス名とインターフェイス ID(文字列として指定できる IPv6 アドレス)に基づいて IPv6 アドレス文字列を作成します。範囲-exprおよびオプション-expr属性で使用されます。
Create Range Operator	
(create-prefix-rangeサイズ n)	<p>範囲 expr属性で使用されるプレフィックスのアドレス範囲 (子) を作成します。関数の基になっているプレフィックス値は、リンク テンプレートをリンクに適用する場合はテンプレート ルート プレフィックス、プレフィックス テンプレートをプレフィックスに適用する場合はプレフィックス アドレスのいずれかです。</p> <p>範囲値 - プレフィックス長の増加。</p> <p>サイズ - プレフィックス長を増やすことができるビット数。1 から 32 までの値を指定する必要があります。親プレフィックスの長さより小さい値にする必要があります。</p> <p>n - 子プレフィックスの n 番目の出現。値は 0 にできますが、サイズの累乗に対して 2 未満に制限されます。サイズ以下にする必要があります。</p> <p>サイズと n にはゼロより大きな値を指定する必要があります。n はサイズ以下にする必要があります、サイズは親プレフィックス長よりも小さくなければなりません。</p> <p>例 :</p> <p>(create-prefix-range 32 0x1)</p>
Create Option Operation	

式の演算子	説明
(create-option opt val)	<p>オプション <code>expr</code>属性で使用される DHCPv6 オプションを作成します。 <code>opt</code> は、オプションを識別するリテラル文字列または整数にすることができます。<code>val</code> は、オプション TLV 値で定義されたオプション値のストリング表現です。</p> <p>カスタム定義オプションと不明オプションを使用できます。未定義のオプションの場合は、オプション番号を指定し、データを (BLOB データとして) 使用する必要があります。データが文字列の場合、文字列は、データが数値またはアドレスである場合は、その文字列を使用します。</p> <p>例 :</p> <pre>(list (create-option " dns-servers" (create-prefix-addr prefix ":::2")) (create-option " domain-list" " sales.example.com,example.com"))</pre> <p>(注) (create-v6-option opt val)は(create-option)のシノニムであり、代わりに使用することができます。</p>
Create Vendor Option Operator	

式の演算子	説明
(create-vendor-option set-name opt val)	<p>オプション-expr属性で使用される DHCPv6 ベンダー オプションを作成します。set-name は、ベンダー オプションに設定されているオプション定義を指定します。opt は、セット内のベンダー・オプションを識別するリテラル・ストリングまたは整数にすることができます。val はオプション値を表します。</p> <p>例：</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::2")) set-name opt val (create-vendor-option "dhcp6-cablelabs-config" 17 " (enterprise-id 4491((tftp-servers 32 3800:0:0:180::6) (config-file-name 33 modem_ipv6.bin) (syslog-servers 34 3800:0:0:180::8) (rfc868-servers 37 3800:0:0:180::6) (time-offset 38 -5h) (cablelabs-client-configuration 2170 (primary-dhcp-server 1 10.38.1.5) (secondary-dhcp-server 2 10.38.1.6))))))")</pre> <p>(注) ((create-v6-vendor-option opt val))は、(create-vendor-option)のシノニムであり、代わりに使用できます。</p>



(注) v4 と v6 の場合は、作成オプションとベンダー作成オプションを使用することをお勧めします。

リンク テンプレートでの式の使用

リンクテンプレートで式を指定して、リンクを作成するときにプレフィックス名、IPアドレス範囲、および埋め込みオプションを動的に作成できます。式には、コンテキスト変数と操作を含めることができます。



(注) 式は DHCP 拡張と同じではありません。式は、クライアント ID の作成やクライアントの検索に一般的に使用されます。拡張 ([拡張ポイントの使用](#)を参照) は、要求パケットまたは応答パケットを変更するために使用されます。

リンクにテンプレートを適用すると、リンクテンプレートにポリシーが埋め込まれている場合、リンクテンプレートはリンクにコピーされます。この埋め込みポリシーには、オプションが含まれている場合と含まれていない場合があります。リンクテンプレートの埋め込みポリシー全体が使用されている場合は、リンク内の既存のオプションが消去されます。リンクテンプレートに埋め込みポリシーがない場合、リンクの埋め込みポリシーは保持されます。次に、リンクテンプレートのオプション式が評価され、オプションがリンク内の埋め込みポリシーオプションに追加されます(埋め込みポリシーが存在しない場合は、1つが作成されます)。

次の表は、リンク テンプレートの定義済み変数を示し、表 5: リンク テンプレート式演算子はリンク テンプレート演算子を示しています。これらの変数と演算子では大文字と小文字が区別されません。表 3: 接頭辞テンプレート式演算子に、接頭辞テンプレート演算子を示します。リンク テンプレート演算子テーブルとプレフィックス テンプレート操作テーブルの両方に同じ演算子が含まれますが、リンクテンプレートだけが[プレフィックス演算子の作成]を使用でき、プレフィックス テンプレートでは演算子を使用できません。

表 4: リンク テンプレート式定義済み変数

定義済み変数	説明
mask-length	プレフィックス マスク ビットの数 (テンプレートルート プレフィックスが定義されている)。
prefix	ネットワーク番号と長さ (テンプレートルート プレフィックスが定義されている)。
prefix-addr	プレフィックスのアドレス部分 (テンプレートルート プレフィックスが定義されている)。
prefix-length	プレフィックス アドレス ビットの数 (テンプレートルート プレフィックスが定義されている)。
template.attribute	リンク テンプレートの属性。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。
this.attribute	リンクの属性。 (注) 属性は明示的に設定する必要があります。それ以外の場合、式は評価に失敗します。
vpn	リンクの VPN。

表 5: リンク テンプレート式演算子

式の演算子	説明
Arithmetic (符号なし整数Operators引数のみ)	
(+ arg1 arg2)	(+ 2 3) などの 2 つの引数値を加算します。

式の演算子	説明
(- arg1 arg2)	最初の引数から 2 番目の引数値を減算します。
(* arg1 arg2)	2 つの引数の値を乗算します。
(/ arg1 arg2)	最初の引数の値を 2 番目の引数の値で除算します (0 にすることはできません)。
(% arg1 arg2)	剰余算術演算子は、最初の引数の結果の残りの部分を 2 番目の引数で除算した値を求めます。
Concatenation Operator	
(concat arg1 ... argn)	引数を文字列に連結します。
List Operator	
(list oper1..オパーン)	<p>オプション リストまたはプレフィックスのリストを作成します。リンクまたはプレフィックスに対して複数のオプションが必要な場合、またはリンクに複数のプレフィックスが必要な場合に必要です。すべての引数は操作 <code>create-v6-option</code> である必要があります。ネストはサポートされていません。</p> <p>例 :</p> <pre>(list (create-prefix " cm-prefix" (create-prefix-range 32 0x1)) (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2)) (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1)))</pre>
Create Prefix Operator	
(create-prefix template prefix)	<p>定義済みのプレフィックス テンプレート名とプレフィックス (リンク VPN を含む) に基づいてプレフィックスを作成します (テンプレート ルート プレフィックスが定義されていると仮定します)。</p> <p>prefix 引数は、プレフィックス名にすることもできますが、<code>create-prefix-addr</code> 演算子の <code>create-prefix-range</code> 値も指定できます。この関数を <code>list</code> 使用して、複数の操作を結合できます。</p> <p>例 :</p> <pre>(create-prefix "cm-prefix" (create-prefix-range 32 0x1))</pre>
Create IP Operator	

式の演算子	説明
(create-prefix-addrプレフィックス インターフェイス ID)	プレフィックス名とインターフェイス ID(文字列として指定できる IPv6 アドレス)に基づいて IPv6 アドレス文字列(テンプレート ルートプレフィックスが定義されていると仮定して)を作成します。プレフィックス-exprおよびオプション-expr属性で使用されます。
Create Range Operator	
(create-prefix-rangeサイズ n)	<p>プレフィックスのアドレス範囲(子)を作成します。関数の基になっているプレフィックス値は、リンク テンプレートをリンクに適用する場合はテンプレートルートプレフィックス、プレフィックステンプレートをプレフィックスに適用する場合はプレフィックスアドレスのいずれかです。</p> <p>範囲値 - プレフィックス長の増加。</p> <p>サイズ - プレフィックス長を増やすことができるビット数。1 から 32 までの値を指定する必要があります。親プレフィックスの長さより小さい値にする必要があります。</p> <p>n - 子プレフィックスの n 番目の出現。値は 0 にできますが、サイズの累乗に対して 2 未満に制限されます。サイズ以下にする必要があります。</p> <p>サイズと n は 0 より大きくなければなりません。</p> <p>n はサイズ以下にする必要があります、サイズは親プレフィックス長よりも小さくなければなりません。</p> <p>例：</p> <p>(create-prefix-range 32 0x1)</p>
Create Option Operator	

式の演算子	説明
(create-option opt val)	<p>オプション <code>expr</code>属性で使用される DHCPv6 オプションを作成します。<code>opt</code> は、オプションを識別するリテラル文字列または整数にすることができます。<code>val</code> は、オプション TLV 値で定義されたオプション値のストリング表現です。</p> <p>カスタム定義オプションと不明オプションを使用できます。未定義のオプションの場合は、オプション番号を指定し、データを (BLOB データとして) 使用する必要があります。データが文字列の場合、文字列は、データが数値またはアドレスである場合は、その文字列を使用します。</p> <p>例 :</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::2")) (create-option "domain-list" "sales.example.com,example.com"))</pre> <p>(注) <code>(create-v6-option opt val)</code>は<code>(create-option)</code>のシノニムであり、代わりに使用することができます。ただし、(作成オプション)を使用することをお勧めします。</p>
Create Vendor Option Operation	
(create-vendor-option set-name opt val)	<p>オプション-<code>expr</code>属性で使用される DHCPv6 ベンダー オプションを作成します。<code>set-name</code>は、ベンダー・オプションのオプション定義セットを指定します。<code>opt</code> は、セット内のベンダー・オプションを識別するリテラル・ストリングまたは整数にすることができます。<code>val</code> はオプション値を表します。</p> <p>次に、例を示します。</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::2")) (create-vendor-option "dhcp6-cablelabs-config" 17 "(enterprise-id 4491((tftp-servers 32 3800:0:0:180::6) (config-file-name 33 modem_ipv6.bin) (syslog-servers 34 3800:0:0:180::8) (rfc868-servers 37 3800:0:0:180::6) (time-offset 38 -5h) (cablelabs-client-configuration 2170 (primary-dhcp-server 1 10.38.1.5) (secondary-dhcp-server 2 10.38.1.6))))))")</pre> <p>(注) <code>(create-v6-vendor-option opt val)</code>は、<code>(create-vendor-option)</code>のシノニムであり、代わりに使用できます。ただし、使用することをお勧めします (<code>create-vendor-option</code>)。</p>