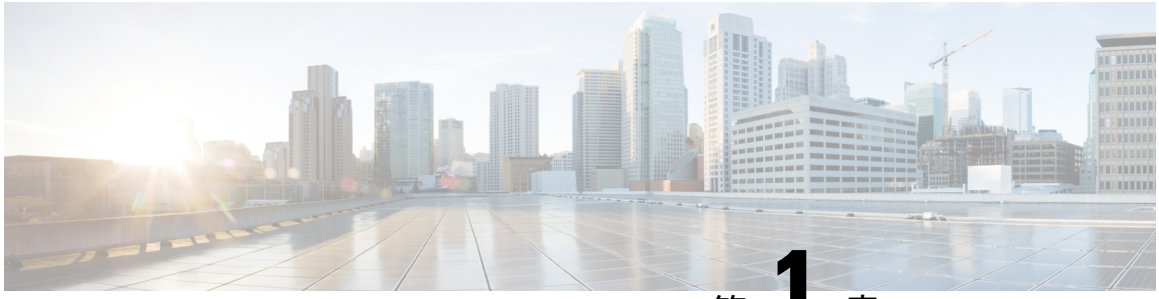
Cisco Prime Network Registrar 11.0 DHCP ユーザーガイド

初版：2021 年 12 月 9 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

目 次



第 1 章

ダイナミックホストコンフィギュレーションの概要

インターネットアクセスを求めるすべてのホストは、IPアドレスを持っている必要があります。インターネット管理者は、新しいユーザーおよびコンピュータが別のサブネットに移動したすべてのユーザーに対して、次の操作を実行する必要があります。

1. 正当な IP アドレスを選択します。
2. アドレスを個々のデバイスに割り当てます。
3. デバイス構成パラメーターを定義します。
4. DNS データベースを更新し、デバイス名を IP アドレスにマッピングします。

これらのアクティビティは時間がかかり、エラーが発生しやすいため、動的ホスト構成プロトコル (DHCP) が発生します。DHCP を使用すると、IP アドレスを個別に割り当てる負担から解放されます。これは、TCP/IP の使用時に必要な設定の量を減らすため、Internet Engineering Task Force（IETF）によって設計されました。DHCP はホストに IP アドレスを割り当てます。また、接続しているインターネットネットワークの情報をホストが操作および交換するために必要なすべてのパラメータを提供します。

DHCP は TCP/IP 構成情報をローカライズします。また、DHCP を使用するように構成されたシステムに IP アドレスを自動的に割り当てることによって、TCP/IP 構成データの割り当てを管理します。したがって、各ホストを個別に構成しなくても、ホストがインターネットにアクセスできることを確認できます。

この章は、次の項で構成されています。

DHCP の仕組み

DHCP は、デバイス構成をサーバー レベルでグローバル アドレス プールに移行することで、動的アドレス割り当てを可能にします。DHCP はクライアント サーバー モデルに基づきます。クライアント ソフトウェアはデバイスで実行され、サーバー ソフトウェアは DHCP サーバーで実行されます。

関連項目

サンプル DHCP ユーザー （2 ページ）

標準 DHCP 管理 （3 ページ）

Leases （4 ページ）

スコープとポリシー （4 ページ）

サンプル DHCP ユーザー

Beth のワークステーション (bethpc) が DHCP で構成された後、次のアクションは、最初に起動したときに発生します。

1. 彼女の PC はネットワーク上の DHCP サーバーから IP アドレスを自動的に要求します。
2. DHCP サーバーは、IP アドレス、割り当てられたリース時間、その他インターネットを使用するために必要な構成データを含むリースを提供します。リースされたアドレスを他人が使用することはなく、彼女の PC でのみ有効です。
3. アドレスのリースが期限切れになる前に、bethpc は、リースを提供したサーバーからリース延長を要求することによってアドレスを更新できます。（通常、このプロセスは、最初に割り当てられたリース時間が約半分経過した時点で始まります）。これにより、有効期限が延長されます。リース時間の約 85% までにリースを更新できない場合、bethpc は、少し異なる要求の送信を開始して、使用可能なサーバーからリースの更新を試みます。サーバーに到達できない場合、Bethpc はリース期間が終了するまでリースを使用し続けます。

   まとめると、クライアントには 3 つの重要な時間があります。

   - **リース有効期限**（Lease Expiration Time）（**有効なライフタイム**（Valid Lifetime））：リースの有効期限が切れになるタイミング。これは常にクライアントに明示的に伝達されます。
   - Renewal Time（T1）（**更新時期**）：リースが許可されたサーバー、または最後にリースが延長されたサーバーで、クライアントが更新プロセスを開始できるタイミング。DHCPv4 の更新はユニキャストです。DHCPv6 の場合、クライアントは、リースが許可されたサーバー、または最後にリースが更新されたサーバーを指定します。

     更新時期（T1）は、サーバーによって明示的に通知されるか、またはクライアントが生成します。デフォルトでは、リース時間の 50% です。

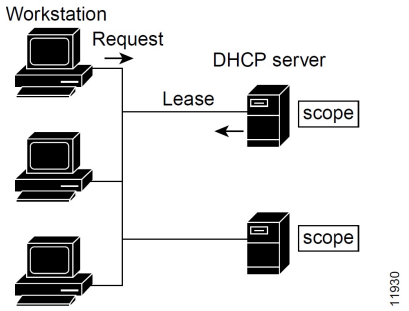
- Rebinding Time（T2）（**再バインド時期**）：クライアントが再バインドプロセスを開始できるタイミング。更新プロセスと似ていますが、単一のサーバーに制限されなくなりました。DHCPv4 の場合、これらの要求はブロードキャストされます（したがって、リレーによってピックアップされ、両方のフェールオーバーパートナーに転送されます）。DHCPv6の場合、クライアントはサーバーを指定しないため、どのサーバーも応答できます。

  再バインド時期（T2）は、サーバーによって明示的に通信されるか、またはクライアントが生成します。通常、リース時間の約 87.5%（DHCPv4 の場合）または、約 85%（DHCPv6 の場合）です。

4. ベスが別の部署に移動し、PC が別のサブネットに移った場合、現在のアドレスは期限切れになり、他のユーザーが利用できるようになります。新しい場所で自分の PC を起動すると、サブネット上の適切なDHCP サーバーからアドレスがリースされます（下の画像を参照）。

DHCP サーバーに正しい構成データが存在する限り、DHCP を使用するワークステーションまたはサーバーの構成が正しく行われなくなります。したがって、トレースが困難な、不適切に構成されたデバイスやサーバーからネットワークの問題が発生する可能性が低くなります。

図 *1:* ホストは *IP* アドレスを要求します



この例では、異なるサブネット上のアドレスを提供する一連の DHCP サーバーを含む DHCP プロトコルを示します。アドレス プールの管理をさらに簡単にするために、多くの場合、ネットワーク ルーターは、中央の DHCP サーバーにクライアント メッセージを転送する DHCP リレー エージェントとして構成されます。このサーバーは、サブネットのグループのアドレス プールで構成されています。

標準 DHCP 管理

DHCP を使用するには、ネットワーク上に少なくとも 1 つの DHCP サーバーが必要です。サーバーをインストールした後:

- DHCP サーバーが DHCP クライアントに提供できる IP アドレスのスコープを定義します。どのアドレスが使用されているか、どのアドレスが使用可能かを追跡する必要はなくなりました。
- 最初の DHCP サーバーがダウンした場合に、配布を共有したりリースを処理したりするようにセカンダリ サーバーを構成します。これは DHCP フェールオーバーと呼ばれます。

DHCPフェールオーバーの管理の詳細についてはDHCPフェールオーバーの管理（65ページ）、を参照してください。

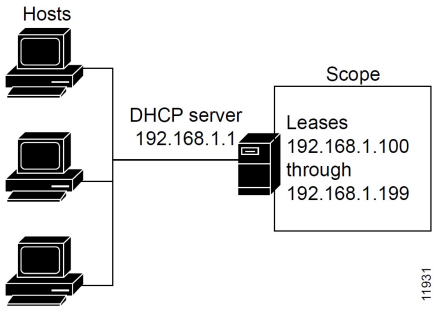
Leases

DHCPの最も大きな利点の1つは、IPアドレスを持つデバイスを動的に構成し、割り当てられたアドレスにリースを関連付けることができることです。DHCPは、ネットワーク内でアドレスを配布および再利用するための自動化された、信頼性が高く安全な方法を提供するリース メカニズムを使用しますが、管理者の介入はほとんど必要ありません。システム管理者は、ネットワークの特定のニーズに合わせてリース ポリシーを調整できます。

リースは、スコープと呼ばれるアドレス プールにグループ化され、要求ホストで使用できるIPアドレスのセットを定義します。リースは予約可能(ホストは常に同じIPアドレスを受け取る)または動的(ホストは、スコープ内で次に使用可能な未割り当てのリースを受け取る)できます。サイトのDHCPサーバーは、アドレス192.168.1.100から192.168.1.199をリースするように構成されています(下の図を参照)。

スコープに構成されたアドレスよりも多くのネットワーク デバイスを使用しない場合は、ネットワーク トラフィックとDHCPサーバーの負荷を軽減するために、1～2週間など、長いリース時間を定義できます。

図 *2 : DHCP* サーバーからのリースを要求する *DHCP* ホスト



スコープとポリシー

スコープには、サブネットのアドレスのセットと、必要な構成パラメーターが含まれます。動的アドレス指定を行う各サブネットに対して、少なくとも1つのスコープを定義する必要があります。

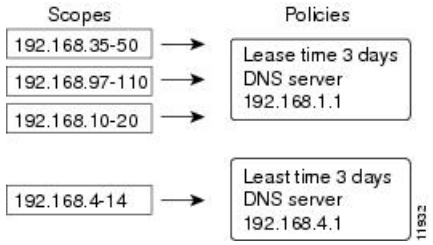
ポリシーには、DHCPサーバーがクライアントに通信するリース時間およびその他の構成パラメータが含まれます。ポリシーを使用して、要求に応じてDHCPサーバーがクライアントに提供するDHCPオプションを構成します。ポリシーを使用すると、DHCPサーバーがスコープごとに個別に指定しなくても、スコープに対して正しいオプションをすべて提供できます(下の図を参照)。

スコープとポリシーの違いは、スコープには、アドレスに関するサーバー情報(どのアドレスが使えなくなっているか、リースを提供する前にクライアントにpingを実行するかどうかな

ど)が含まれることです。ポリシーには、ローカルDNSサーバーのリース期間やアドレスなどのクライアント構成データが含まれます。

ポリシーは、サーバー上に複数のスコープがある場合に特に便利です。すべてのスコープまたは選択したスコープに適用されるポリシーを作成できます。Cisco Prime Network レジストラーポリシー階層は、最も限定的なポリシーから最も具体的なポリシーを定義する方法です。たとえば、通常は各ポリシーにルーターオプションを指定します。このようなスコープ固有のポリシーは、スコープ埋め込みポリシーで定義できます。リース時間を参照するような、より一般的なポリシーは、システム全体のポリシーに適用できます（「DHCPポリシーの設定 （197ページ）」を参照）。ポリシーの割り当てを処理する拡張機能を作成することもできます（「DHCPサーバーの動作に影響を与える拡張機能の使用 （28 ページ）」を参照）。

図 *3:* スコープとポリシー



リンクとプレフィックス

明示的な DHCPv6 構成オブジェクトは、リンクおよびプレフィックスです。

- Link:1 つ以上のプレフィックスを持ち、DHCPv6 クライアントにポリシーを適用できる追加レイヤを追加できるネットワーク セグメント。
- Prefix— IPv4 のスコープに相当します。プレフィックスに関連付けられたリンクは、別のプレフィックスではなくリンクの名前を付ける点を除いて、プライマリ スコープに似ています。

スコープの場合と同様に、同じ IPv6 プレフィックスに対して複数のプレフィックス オブジェクトを作成できます。ただし、明示的な開始アドレスと終了アドレスを持つ複数の範囲をサポートするのではなく、プレフィックスは、プレフィックスオブジェクトと同じ長さ、または長い IPv6 プレフィックスである必要がある 1 つの範囲のみをサポートします。たとえば、2001:/64 のプレフィックスを 2001::/96 の範囲で定義すると、サーバーは 2001:0:0:0:0:0:0:0 から 2001:0:0:0:0:0:ffff のみアドレスを割り当てることができます。範囲:

- 2の累分に制限される。
- 一意である必要があります (別の VPN を除き、他の範囲で複製することはできません)。
- 以下で説明するプレフィックスの委任プレフィックスを除き、別の範囲に含めたり含めたりすることはできません。
- 以下で説明するプレフィックスの委任プレフィックスを除き、指定されていない場合は完全な IPv6 プレフィックスです。

プレフィックスの委任プレフィックス オブジェクトが指定されていない範囲で定義されている場合、プレフィックス委任プレフィックス以外のプレフィックスが含まれている可能性があり、有効範囲は次のいずれかになります。

- 同じ IPv6 プレフィックスを持つ他のプレフィックスが存在しない場合は、完全な IPv6 プレフィックス
- 同じ IPv6 プレフィックスを持つプレフィックス オブジェクトの他のすべての範囲が IPv6 プレフィックスから削除された場合に残るプレフィックス。

リンクを作成するのは、異なる IPv6 プレフィックスを持つ複数のプレフィックス オブジェクトがリンク上に存在する場合だけです。サーバーが設定をロードするときに、プレフィックスに明示的なリンクがない場合、サーバーはLink-vpn.name/]という名前の暗黙的なリンクを検索または作成します。同じIPv6プレフィックスを持つすべてのプレフィックス オブジェクトは、リンクを指定しないか、同じリンクを明示的に指定する必要があります。

DHCPv6 対応サーバーは、DHCPv6 の VPN アドレス空間をサポートします。リンクオブジェクトとプレフィックスオブジェクトの両方を VPN に割り当てることができます。ただし、リンク上のすべてのプレフィックスは同じ VPN ID を使用する必要があります。現在、DHCPv6 VPN オプションがないため、クライアントまたはクライアント クラスのoverride-vpn属性を使用して、クライアントに VPN からのアドレスを割り当てることのみが可能です。

関連項目

リンクとプレフィックスの決定 （156 ページ）
アドレスの生成 （157 ページ）
委任プレフィックスの生成 （158 ページ）
プレフィックス安定性 （158 ページ）

シスコ プライムネットワーク レジストラー DHCP 実装

Cisco プライムネットワーク レジストラー DHCP サーバーは、ネットワーク上のホストに IP アドレスを自動的に割り当てる信頼性の高い方法を提供します。DHCP クライアント設定を定義し、Cisco Prime Network レジストラーデータベースを使用して、クライアント IP アドレスの割り当ておよびその他のオプションの TCP/IP およびシステム設定パラメータを管理できます。TCP/IP 割り当て可能なパラメーターには、次のものがあります。

- ホスト内の各ネットワーク アダプタ カードの IP アドレス。
- 物理 (サブネット) ネットワーク識別子である IP アドレスの一部のサブネット マスク。
- サブネットを他のネットワーク セグメントに接続するデフォルト ゲートウェイ (ルーター)。
- ドメイン名など、DHCP クライアントに割り当てることができる追加の構成パラメータ。

Cisco プライムネットワーク レジストラーは、DHCP サーバー ソフトウェアをインストールすると、データベースを自動的に作成します。WEB UI または CLI を使用して、DHCP スコープとポリシーを定義するときにデータを追加します。

Cisco Prime Network レジストラー DHCP サーバーは、仮想プライベート ネットワーク(VPN)およびサブネットのアドレスをオンデマンドアドレス プール用のプール マネージャ デバイスに割り当てることもサポートしています。これらの機能の詳細については、以下の項で説明します。

関連項目

バーチャル プライベート ネットワーク （7 ページ）

サブネットの割り当てと DHCP アドレス ブロック （116 ページ）

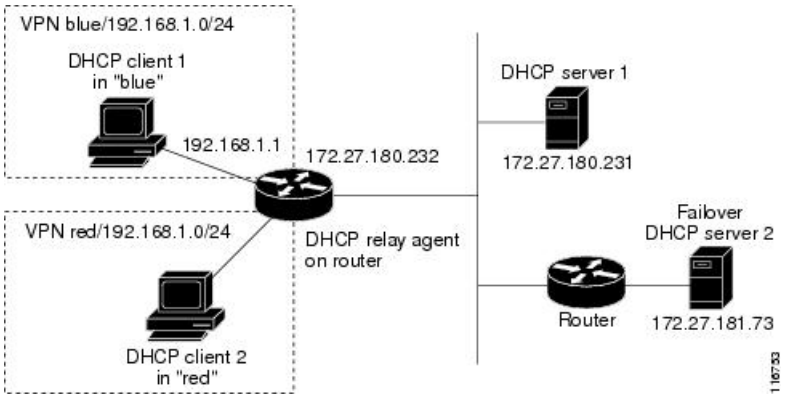
バーチャル プライベート ネットワーク

仮想プライベート ネットワーク（VPN）では、別々のネットワークの 2 つのプールが同じ DHCP サーバーが使用する同じアドレス空間をプライベート ネットワーク アドレスを使用して保持できます。これにより、貴重なパブリックアドレスを使用しなくても、アドレスリソースを節約できます。ただし、これらの VPN アドレスには、他の重複する IP アドレスと区別するために特別な指定子が必要です。クライアントと同じ VPN 上にない Cisco Prime Network レジストラー DHCP サーバーは、リースとアドレスをクライアントに割り当てることができ、1 つの VPN から別の VPN にアドレスを区別できます。

Cisco Primeネットワーク レジストラー DHCP サーバーおよび Cisco IOS DHCP リレー エージェントに加えられた変更を通じて、DHCP サーバーは複数の VPN 上のクライアントにサービスを提供できます。VPN は、DHCP サーバー オブジェクトのセットを区別し、他のアドレス空間にある同じオブジェクトから独立しています。同じアドレスを含む複数の VPN を定義できます。Cisco IOS リレーエージェントで設定された VPN 識別子に基づいて VPN を作成します。

次の図は、一般的な VPN 対応 DHCP 環境を示しています。DHCP リレー エージェントは、アドレス空間が重複する 2 つの異なる VPN(青と赤)にサービスを提供します。リレーエージェントは、VPN ブルーのインターフェイス アドレス 192.168.1.1 を持ち、DHCP サーバー 1 には 172.27.180.232 として知られています。DHCP クライアント 1 からの要求を VPN ブルーで処理するサーバーは、クライアントとは異なるネットワークまたはネットワークセグメント上に配置でき、DHCP Server 2 でフェールオーバー構成に入ることができます（「DHCP フェールオーバーの管理 （65 ページ）」を参照）。リレー エージェントは、リレー エージェントと Cisco Prime Network レジストラー管理者の間で調整された、DHCP サーバーへのクライアント アドレス要求の特別な識別ルートを識別できます(RFC 6607 を参照)。DHCP サーバーは、両方の VPN 上のクライアントに重複する IP アドレスに基づいてリースを発行できるようになりました。

図 *4:* バーチャル プライベート ネットワーク *DHCP* 構成

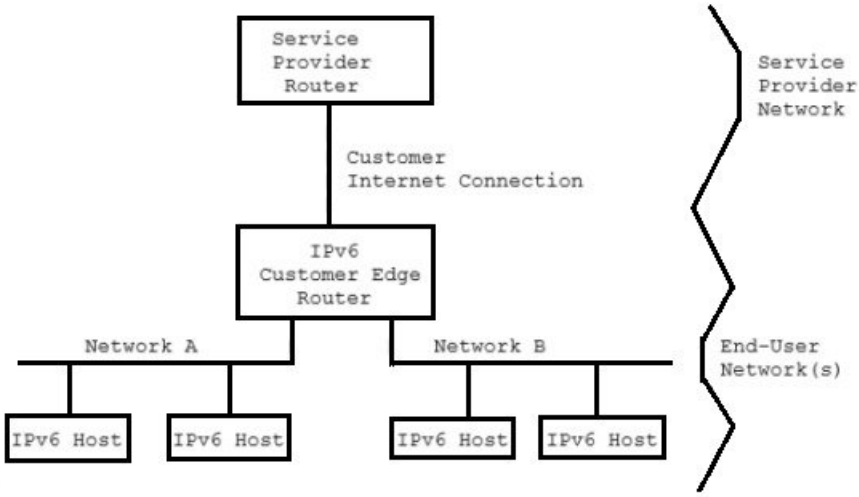


プレフィックス委任

プレフィックスの委任を使用すると、DHCPv6 サーバーから要求元デバイスへのプレフィックスの委任が可能になります。プレフィックス委任は、顧客宅内機器(CPE)デバイスにプレフィックスを割り当てるサービス プロバイダによって使用されます。ISP は、プレフィックスをサブスクライバーに委任するためにも使用されます。

操作中に、要求側デバイスに委任する IPv6 プレフィックスが DHCPv6 サーバーに提供されます。要求側のデバイスは、DHCPv6 サーバーにプレフィックスを要求します。DHCPv6 サーバーは、委任のプレフィックスを選択し、要求側のデバイスにプレフィックスを付けて応答します。要求側のデバイスは、委任されたプレフィックスを担当します。たとえば、要求元のデバイスは、デリゲートされたプレフィックスからそのインターフェイスのいずれかにサブネットを割り当て、そのリンクのプレフィックスの通知の送信を開始できます。各プレフィックスには有効な有効期間と優先存続期間が関連付けられており、要求側のデバイスがプレフィックスを使用できる時間の長さに関する合意が構成されます。要求元のデバイスは、デリゲートされたプレフィックスの有効期間の延長を要求でき、プレフィックスの有効期間が期限切れになった場合に委任されたプレフィックスの使用を終了する必要があります。

図 *5:* エンドユーザーネットワークのモデルトポロジ



DNS 更新

DHCP は IP アドレスの配布の負担から解放されますが、DHCP クライアントの名前とアドレスを使用して DNS サーバーを更新する必要があります。DNS 更新は、名前とアドレスを最新の状態に保つタスクを自動化します。Cisco Prime Network レジストラー DNS アップデート機能を使用すると、名前とアドレスの関連付けが発生または変更されたときに、DHCP サーバーは対応する DNS サーバーに伝えることができます。クライアントがリースを取得すると、Cisco Prime Network レジストラーは DNS サーバーにホスト データを追加するように指示します。リースの期限が切れた場合、またはホストがリースを終了すると、Cisco Prime Network レジストラーは DNS サーバーにアソシエーションを削除するように指示します。

通常の動作では、DHCP を介してクライアントのアドレスが変更される頻度に関係なく、DNS を手動で再構成する必要はありません。Cisco プライムネットワーク レジストラーは、クライアント デバイスが提供するホスト名を使用します。また、Cisco Prime Network レジストラーで、クライアントを提供しないクライアントの名前を合成したり、クライアントルックアップ機能を使用してクライアントに事前設定されたホスト名を使用したりすることもできます。

DHCPv4 および DHCPv6 DNS 更新のユース ケースが異なるために、ホスト名の更新を処理するためにサーバーの設計が異なりました。したがって、ホスト名の DHCPv4 および DHCPv6 DNS 更新の動作の違いが予想されます。

関連項目

DNS へのリース取得の影響 （10 ページ）

リースのリリースによる DNS への影響 （10 ページ）

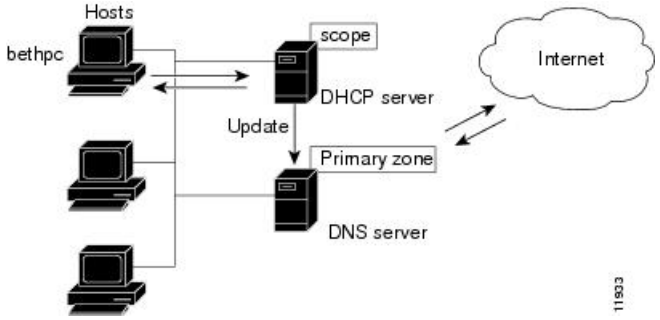
リース再獲得の DNS への影響 （10 ページ）

DNS へのリース取得の影響

ExampleCo の場合、管理者は DHCP サーバーにスコープを作成し、100 のリースを割り当てます (192.168.1.100 から 192.168.1.199)。各デバイスは、その所有者名を取得します。また、管理者は、DNS 更新を使用するように DHCP サーバーを構成し、それに対応する構成済み DNS サーバーに関連付けます。管理者は、DNS サーバー データベースに名前を入力する必要はありません。

月曜日の朝、ベス（bethpc のユーザー）は、アドレスなしでウェブサイトにログインしようとします。ホストが起動すると、アドレス要求をブロードキャストします(下の画像を参照)。

図 *6:* 企業の *DNS* 更新



DHCP サーバーは次のようになります。

1. 次に使用可能な(未割り当て)IPアドレス(192.168.1.125)を bethpc に与えます。
2. ホスト名とアドレス (bethpc 192.168.1.125) で DNS サーバーを更新します。

ベスはウェブサイトにアクセスできるようになりました。さらに、Beth のコンピュータ名を自分の IP アドレスに変換する必要があるプログラム、または逆の方法で DNS サーバーにクエリを実行することもできます。

リース再獲得の DNS への影響

ベスは再び彼女のホストを起動するために彼女の旅行から戻ったとき:

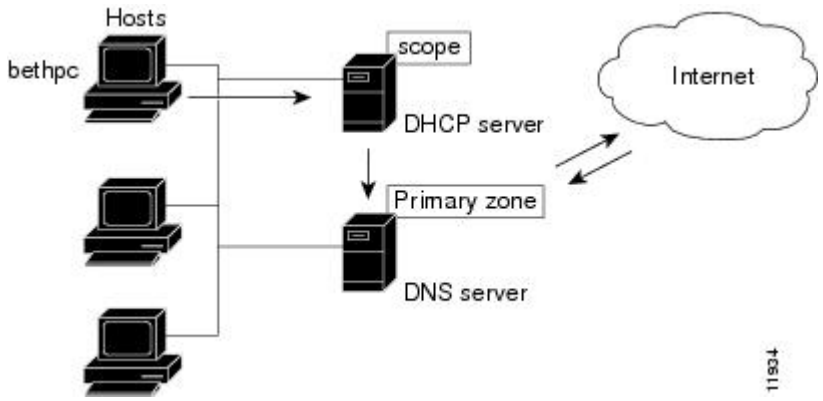
1. 彼女の PC は IP アドレスをブロードキャストします。
2. DHCP サーバーは、ホストが正しいネットワーク上にあるかどうかを確認します。その場合、サーバーはアドレスを発行します。正しくない場合は、正しいネットワーク上のサーバーがアドレスを発行します。
3. DHCP サーバーは、ホストとアドレスのデータを使用して DNS サーバーを再度更新します。

リースのリリースによる DNS への影響

その日の後半,ベスは町を出て行く必要があることを知りました。彼女は、3日後に期限切れになるリースアドレスをまだ持っているホストをオフにします。リースが解放されると、DHCP サーバーは次の処理を行います。

1. IP アドレスが他のユーザーに対して利用可能になったことを確認します(下の図を参照)。
2. ホスト名とアドレスを削除して DNS サーバーを更新します。DNS サーバーは、bethpc またはそのアドレスに関するデータを保存しなくなりました。

図 7: リースの放棄



DHCP フェールオーバー

Cisco Prime Network レジストラーフェールオーバー プロトコルは、何らかの理由でメイン サーバーがオフラインになった場合に、バックアップ DHCP サーバーが引き継がれるように設計されています。8.2 より前のバージョンでは、このプロトコルは UDP ベースで、IPv4 経由でのみ動作し、DHCPv4 のみをサポートしていました。8.2 以降、このプロトコルは TCP ベースで、IPv4 または IPv6 のいずれかを使用するように構成でき、単一の接続で DHCPv4 と DHCPv6 の両方をサポートします。DHCP サーバーは、両方を使用するように構成されている場合、IPv4 と IPv6 の両方のトランスポートを試行し、最初に起動した接続を使用します。既存の DHCP クライアントは、どのサーバーが要求に応答するかを知らなくても、リースを維持および更新できます。

フェールオーバー ペアは、Cisco Prime Network レジストラーのローカル クラスタとリージョン クラスタで作成および同期できます。詳細については、DHCP フェールオーバーの管理（65 ページ） を参照してください。

フェールオーバーによるアドレスの割り当て

ネットワーク パーティションが相互に通信できるが、相互通信できないネットワーク パーティションにもかかわらず、フェールオーバー ペアを動作させるには、単一サーバーの実行に必要なアドレスよりも多くのアドレスを使用できるようにする必要があります。メイン サーバーを構成して、各スコープまたはプレフィックス委任アドレス プールで現在使用可能な (割り当てられていない) アドレスの割合をパートナーに割り当てます。これらのアドレスは、メイン サーバーで使用できなくなります。パートナーは、メイン サーバーとの間で話ができないときに、メイン サーバーがダウンしているかどうかを知らない場合に、これらのファイルを使用します。ただし、フェールオーバー パートナーが通信中に、定期的にこれらのプールのバランスを調整します。

バックアップ サーバーは、メイン サーバーがダウンしているかどうかをバックアップが認識しない期間に到着したすべての新しいDHCPクライアントの要求を満たすのに、各スコープまたはプレフィックスから十分なアドレスを必要とします。フェールオーバーペアの既定のバックアップの割合は 50% です。これにより、フェールオーバー中に他のパートナーのアドレス数が同じになります。

PARTNER-DOWN 状態の間でも、バックアップ サーバーはリースの有効期限とクライアントの最大リードタイム(MCLT)、小さな追加のタイム バッファを待ってから、リースを再割り当てします。これらの時間が経過すると、バックアップ サーバーは次の機能を提供します。

- アドレスのプライベート プールからのリース。
- アドレスのメイン サーバー プールからのリース。
- 新しいクライアントへのリースの期限が切れています。

稼働時間内に、管理スタッフが COMMUNICATIONS INTERRUPTED 状態に 2 時間以内に応答してメイン・サーバーが稼働しているかどうかを判別できる場合、バックアップ・サーバーは、新規DHCPの数に対して妥当な上限をサポートするのに十分なアドレスを必要とします。この 2 時間の間に到着する可能性のあるクライアント。

営業時間外に、管理スタッフが同じ状況に 12 時間以内に応答でき、DHCP クライアントから以前に知られていなかった到着率も少ないと考えると、バックアップ サーバーは妥当な上位をサポートするのに十分なアドレスを必要とします。この 12 時間の間に到着する可能性のあるDHCP クライアントの数に制限されます。

したがって、バックアップサーバーが単独で制御するアドレスの数は、ピーク時およびピーク時以外に指定されたアドレスの数のうち、それぞれで現在利用可能な(未割り当て)アドレスの割合で表されるアドレスの数です。スコープまたはプレフィックス。

（注） DHCPフェールオーバーペアの既定の使用セーフ期間が有効になり、既定のセーフ期間は4時間です。これにより、フェイルオーバー・パートナーが 4 時間の COMMUNICATIONS-INTERRUPTED 状態の場合、安全期間が経過した後に自動的に PARTNER-DOWN 状態になります。

クライアントクラス

Cisco Prime Networkレジストラークライアントおよびクライアント クラスの機能を使用して、共通のネットワークに接続されているユーザーに差別化されたサービスを提供できます。管理基準に基づいてユーザー・コミュニティをグループ化し、各ユーザーが適切なサービス・クラスを受け取れるようにすることができます。

Cisco Prime Network レジストラークライアント クラス機能を使用して、設定パラメータを制御できますが、最も一般的な用途は次のとおりです。

- Lease periods：一連のクライアントがアドレスを保持する期間。
- IP address ranges：クライアントアドレスを割り当てるリースプールの元。

- DNS server addresses：クライアントが DNS クエリを送信する場所。
- DNS hostnames：クライアントを割り当てる名前。
- Denial of service：許可されていないクライアントにリースを提供するかどうか。

クライアントクラスの機能を使用する 1 つの方法は、訪問者がネットワークの一部 (すべてではない) にアクセスできるようにすることです。たとえば、ExampleCo の訪問者である Joe がラップトップをexample.com ネットワークに接続しようとすると、Cisco Prime Network レジストラーはラップトップを外部として認識します。ExampleCo は、ネットワーク全体へのアクセス権を持つクライアントの 1 つのクラスを作成し、サブネットへのアクセス権を持つ別の訪問者クラスを作成します。Joe が標準訪問者アクセス以上のものを必要とする場合は、ラップトップを Cisco Prime Network レジストラーシステム管理者に登録し、適切なサービスを使用して別のクラスに追加できます。

次のセクションでは、DHCP が通常アドレス割り当てを処理する方法、およびクライアントクラス機能を有効にして DHCP がアドレス割り当てを処理する方法について説明します。

関連項目

クライアントクラスなしの DHCP 処理 （13 ページ）
クライアントクラスがある DHCP 処理 （14 ページ）
クライアントクラスへのスコープの定義 （15 ページ）
ネットワークとスコープの選択 （16 ページ）

クライアントクラスなしの DHCP 処理

クライアントクラスの処理を適用する方法を理解するには、DHCP サーバーがクライアント要求を処理する方法を理解しておくと役立ちます。サーバーは、次の 3 つのタスクを実行できます。

- IP アドレスを割り当てます。
- 適切な DHCP オプション(構成パラメータ)を割り当てます。
- 必要に応じて完全修飾ドメイン名 (FQDN) を割り当て、その名前で DNS サーバーを更新します。

以下は、その DHCP サーバーによる追加処理です。

1. 定義されたスコープからクライアントにアドレスを割り当てる: クライアントのアドレスを選択するには、DHCP サーバーが要求パケットの内容に基づいてクライアント サブネットを決定し、そのサブネットに適したスコープを見つけます。

   1 つのサブネットまたは複数のネットワーク セグメント (マルチネット化) に複数のスコープがある場合、DHCP サーバーはラウンドロビン方式でこれらのスコープの中から選択するか、DHCP サーバー のアドレス割り当て優先順位機能を使用してスコープの選択の優先順位割り当て優先順位を使用した複数スコープの設定 （140 ページ）を変更できます (を参照してください)。サーバーは、スコープを選択した後、そのスコープから使用可能な (割り当てられていない) アドレスを選択します。

1. 定義されたポリシーから DHCP オプション値を割り当てます。Cisco プライムネットワーク レジストラーでは、オプションをグループ化するポリシーを使用します。ポリシーには、スコープ固有とシステムの既定の 2 種類があります。クライアントが要求する DHCP オプションごとに、DHCP サーバーは定義された順序で値を検索します。
2. スコープ固有のポリシーにオプションが含まれている場合、サーバーはその値をクライアントに返し、検索を停止します。
3. 見つからない場合、サーバーはシステムのデフォルト・ポリシーを調べ、その値を戻し、検索を停止します。
4. どちらのポリシーにもこのオプションが含まれている場合、サーバーはクライアントに値を返さないので、エラーをログに記録します。
5. サーバーは、要求されたオプションごとにこのプロセスを繰り返します。

2. DNS 更新が有効な場合、サーバーはクライアントに FQDN を割り当てます。DNS アップデートを有効にした場合、Cisco Prime Network レジストラーは DNS ホスト テーブルにクライアント名とアドレスを入力します。DNS 更新 （9 ページ）を参照してください。クライアント名は次のことができます。
   - クライアント リース要求で指定された名前 (既定値)。
   - その MAC アドレス (ハードウェア アドレス、たとえば、00:d0:ba:d3:bd:3b)。
   - デフォルトのプレフィックスdhcpまたは指定したプレフィックスを使用する一意の名前。

クライアントクラスがある DHCP 処理

DHCP サーバーのクライアント クラス機能を有効にすると、要求処理は IPクライアントクラスなしの DHCP 処理 （13 ページ）アドレス、オプション、およびドメイン名を割り当てるのと同じ3つのタスクを実行しますが、機能が追加されます。以下は、その DHCP サーバーによる追加処理です。

1. Considers the client properties and client-class inclusion before assigning an address：通常の DHCP 処理と同様に、DHCP サーバーはクライアント サブネットを決定します。次に、サーバーは、クライアント クラスが定義されているか、またはこのクライアントの MAC アドレスがデータベースに存在するか確認します。次の場合：
   1. クライアント クラスの検索 ID 式によって定義されたクライアント クラスは、このクライアント クラスのメンバーになります。
   2. MACアドレスなし、デフォルトのクライアントを使用します。たとえば、既定のクライアントではクライアント クラス名を Guest に設定し、クライアント クラスは、クライアントが許可されるネットワーク操作を制限できます（オプションとアドレスの選択を使用）。
   3. MAC アドレスがなく、デフォルトのクライアントも、サーバーは通常の DHCP 処理を通じてクライアントを処理します。
   4. クライアント指定子はありませんが、MAC アドレスは、MAC アドレスはクライアント指定子に変換されます。既定のクライアントが定義されている場合、不明なクライアントが既定のクライアントにマップされます。

スコープには、クライアントからアクセス可能なサブネット上のアドレスが必要です。つまり、クライアントクラスに関連付ける選択タグが必要です。同じクライアントを異なるアドレス プールに割り当てるには、別々のスコープを使用する必要があります。

たとえば、スコープにはEmployeeまたはGuestの選択タグが付いていますが、両方は使用できません。この場合、各サブネットには2つのスコープがあります。1つは選択タグEmployee、もう1つはゲストです。各スコープには、ユーザー グループに適切なアクセス権を提供する、関連付けられたポリシーとアドレス範囲が異なります。

2. Checks for :通常client-classのDHCP処理では、サーバーはスコープ固有のDHCPオプションとシステムデフォルトのDHCPオプションをチェックします。 DHCP options クライアント クラスでは、まずクライアント固有のオプションとクライアント クラス固有のオプションもチェックします。
3. Provides additional -FQDNクライアントが要求するホスト名を使用する通常の名前割り当てプロセスを超えて、サーバーは次のことができます。 assignment options
   - それをオーバーライドする明示的なホスト名を指定します。
   - クライアントが要求したホスト名を削除し、置き換えないようにします。
   - クライアントのMACアドレスからホスト名を合成します。

クライアントクラスへのスコープの定義

クライアント クラスを使用する動機付けの要因は、1つまたは別のアドレス プールからクライアントにアドレスを提供することです。もう1つの動機として、クライアントに異なるオプション値またはリース時間を提供することが考えられます。クライアントに別のプールからアドレスを提供するには、複数のスコープを定義する必要があります。

サブネット上で複数のスコープを取得するには、同じネットワーク セグメントから取得する必要があります。ネットワークは、Cisco Primeネットワーク レジストラーでは直接設定されませんが、スコープ設定から推測されます。スコープが関連付けられるようになる (最終的には同じネットワークに入る):

- Implicitly- 2つのスコープのネットワーク番号とサブネット マスクが同じ。これらのスコープは、明示的な構成なしで同じネットワーク上で自然に終了します。
- Explicitly- 1つのスコープは、別のスコープに対するセカンダリとしてマークされます。これは、セカンダリとしてマークされたスコープに、プライマリとは無関係のネットワークとサブネットマスクがある場合に必要です。たとえば、通常のルーティング可能なネットワーク セグメントに10.0.0.0ネットワーク アドレスのセットを配置する場合が考えられます。

Cisco Prime Network レジストラー DHCP サーバーがデータベースからスコープ設定を読み取ると、すべてのスコープがネットワークに配置され、この情報がログに記録されます。同じネットワーク番号とサブネット マスクを持つスコープは同じネットワークに終わり、セカンダリ スコープはプライマリ スコープ ネットワークに終わります。

ネットワークとスコープの選択

DHCP パケットが到着すると、サーバーは、受信元のアドレスを決定します。

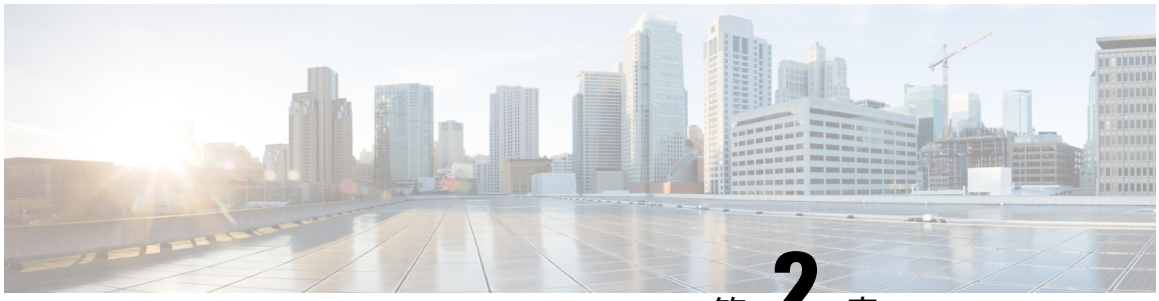
- DHCPv4 パケットが到着すると、サーバーはゲートウェイ アドレス (giaddr) を決定します (もしあれば、BOOTP リレーを介して送信されたパケットの場合)。
- DHCPv6 の詳細については、リンクとプレフィックスの決定を参照してください。
- DHCP クライアントが DHCP サーバーも直接接続されているネットワーク セグメント上にある場合、ブロードキャストパケットが到着したインターフェイスのインターフェイス アドレス。

いずれの場合も、DHCP サーバーはゲートウェイまたはインターフェイスアドレスからネットワークを決定します。次に、ネットワークに複数のスコープがある場合、サーバーはDHCP クライアントにアドレスを割り当てるスコープを決定します。常に、このタイプのクライアントにアドレスを割り当てることができるスコープを探します。たとえば、DHCP クライアントには DHCP をサポートするスコープが必要で、BOOTP クライアントは BOOTP をサポートするスコープを必要とします。クライアントがDHCP クライアントであり、DHCP をサポートするスコープが複数あり、それぞれが使用可能な (割り当てられていない) アドレスを持つ場合、DHCP サーバーは、それらのスコープのいずれかから IP アドレスをラウンドロビン方式で、または割り当て優先順位によって割り当てます。

選択タグとクライアントクラスを使用すると、次のIPアドレスを割り当てるようにDHCPサーバーを構成できます。

- ネットワーク上の 1 つ以上のスコープを 1 つのクラスのクライアントに対して行います。
- 異なるクラスのクライアントに対するスコープの異なるセット。

後者の場合、ゲートウェイまたはインターフェイスアドレスによってネットワークが決まります。クライアントクラス機能は、選択タグのメカニズムを通じて、使用するネットワーク上のスコープを決定します。



第 2 章

DHCP サーバーの管理

この章では、DHCPサーバーパラメータの一部を設定する方法について説明します。クライアントがアドレス割り当てにDHCPを使用できるようにするには、少なくとも1つのスコープをサーバーに追加する必要があります。この機能については、スコープ、プレフィックス、リンク、ネットワークの管理 （137 ページ）で説明しています。

Cisco Prime Network Registrar のフェールオーバー プロトコルは、メイン サーバーが何らかの理由でオフラインになった場合に、バックアップのDHCPサーバーがメイン サーバーを引き継いで動作できるように設計されています。DHCP フェイルオーバーを設定するには、DHCP フェールオーバーの管理 （65 ページ）を読みます。

DHCP サーバーの設定

DHCPサーバーの設定では、サーバーのプロパティ、ポリシーおよび関連するDHCPオプションを設定する必要があります。Cisco Prime Network Registrar では以下が必要です。

- DHCP サーバーの IP アドレス
- 1つ以上のスコープ(スコープの管理 （137ページ）を参照)およびおよび/またはプレフィックス

一般的な設定時の注意事項

DHCP サーバーを構成する前に考慮すべきガイドラインを次に示します。

- Separate the DHCP server from secondary DNS servers used for DNS updating：大規模ゾーン転送時に DHCP サーバーが悪影響を受けないようにするには、セカンダリ DNS サーバーとは異なるクラスタで DHCP サーバーを実行する必要があります。
- Lease times：リース期間のガイドライン （229 ページ）を参照してください。

DHCP サーバー インターフェイスの設定

DHCP サーバーを設定するには、Cisco Prime Network レジストラーの のデフォルトを受け入れるか、データを明示的に指定します。

- Network interface：イーサネット カードの IP アドレス(静的で、DHCP によって割り当てられていないもの)
- Subnet mask：インターフェイス ネットワーク メンバーシップを識別します。サブネット マスクは通常、インターフェイス アドレスのネットワーク クラスに基づいていますが、ほとんどの場合 255.255.255.0 です。

既定では、DHCP サーバーはオペレーティング システムのサポートを使用して、コンピュータ上のアクティブなインターフェイスを自動的に列挙し、それらのすべてをリッスンします。サーバー・インターフェースを手動で構成することもできます。DHCP サーバーが存在するマシン上の NIC カードに割り当てられたすべての IP アドレスを静的に構成する必要があります。マシンは BOOTP または DHCP クライアントであってはなりません。

（注） DHCP に使用するインターフェイスを制限する必要がある場合を除き、特定の DHCP サーバー インターフェイスを構成しないことをお勧めします。サーバーが使用可能なインタフェースを自動的に検出できるようにします。

ローカル アドバンスド Web UI

ステップ 1 [操作（Operate）] メニューのサブメニュー Servers の下で Manage Servers を選択し、[サーバーの管理] ページを開きます。

ステップ 2 [マネージャーサーバー（Manager Servers）] ウィンドウで [ローカルDHCPサーバー（Local DHCP Server）] を選択します。

ステップ 3 [ネットワークインターフェイス（Network Interfaces）] タブをクリックして、サーバーに構成できるネットワーク インターフェイスを表示します。デフォルトでは、サーバーはこれらすべてのものを使用します。

ステップ 4 インターフェイスを設定するには、インターフェイスの [構成] 列の [編集（Edit）] アイコンをクリックします。これにより、インターフェイスが [構成済みインターフェイス] テーブルに追加され、編集または削除できます。

ステップ 5 設定されたインターフェイスの名前をクリックすると、[DHCPサーバーネットワークインターフェイスの編集（Edit DHCP Server Network Interface）] ページが開き、インターフェイスのアドレスとポートを（エキスパート モードで）変更できます。

ステップ 6 Save をクリックして変更を保存します。

ステップ 7 Revert をクリックして、[サーバーの管理（Manage Servers）] ページに戻ります。

CLI コマンド

DHCPdhcp-interfaceサーバーが DHCP クライアントをリッスンするネットワーク インターフェイス カードの IP アドレスを手動で制御するために使用します。デフォルトでは、DHCP サーバーは自動的にすべてのサーバー ネットワーク インターフェイスを使用するため、このコマンドを使用して、使用するネットワーク インターフェイスをより詳細に指定します。

構成変更の妥当性をトラブルシューティングして確認する。

- DHCP サーバーをリロードします。
- dhcp_startup_logファイルまたはname_dhcp_1_logファイルを確認してください。

ログの設定の詳細については、DHCP サーバーの調整 （35 ページ）を参照してください。

詳細なサーバー属性の定義

カスタム DHCP オプションを含む、高度な DHCP サーバー属性を設定できます。

DHCP サーバーをセットアップするには、次の手順を実行します。

1. スコープまたはプレフィックスを構成します。
2. サーバーをリロードします。

関連項目

詳細な DHCP サーバー属性の設定 （19 ページ）

スコープの BOOTP の有効化 （62 ページ）

BOOTP クライアントの移動または廃止 （62 ページ）

動的 BOOTP の使用 （62 ページ）

BOOTP リレー （63 ページ）

詳細な DHCP サーバー属性の設定

次の表は、ローカル クラスタ Web UI および CLI で設定できる高度な DHCP サーバー属性を示しています。

表 *1 : DHCP* の詳細属性

詳細パラメータ	アクション	説明
max-dhcp-requests	設定/設定解除	DHCP クライアントおよびフェールオーバー パートナーからのパケット受信に DHCP サーバーが割り当てるバッファの数を制御します。この設定が大きすぎると、DHCP アクティビティのバーストによって、処理前に古くなった要求でサーバーが詰まる可能性があります。その結果、クライアントが新しいリースを取得しようとしてパフォーマンスが著しく低下し、バーストの処理能力に影響を与える処理負荷が増大します。バッファの設定が低いと要求が調整され、サーバーのスループットに影響する可能性があります。サーバーがバッファを使い果たした場合、パケットは破棄されます。 負荷が高いと予想される場合 (定常状態または頻繁なストレス時間が発生した場合)、または高速マルチプロセッサ システムがある場合は、バッファーを増やすという優れた規則または経験があります。

詳細パラメータ	アクション	説明

		非フェールオーバー展開では、既定の設定 (500) で十分です。フェールオーバー展開では、DHCP ログが要求バッファの数が常に多いことを示している場合は、この数を 1000 に増やすことができます。また、DHCP 応答の数(max-dhcp-responsesパラメーターを参照) を要求バッファーの 4 倍に変更する必要があります。
		
		LDAP クライアントルックアップを使用する場合、バッファは、LDAP 接続の合計数と各接続に許可される要求の最大数で定義された LDAP ルックアップ・キュー・サイズを超えないようにする必要があります。LDAP サーバーの容量をサービス クライアントルックアップに合わせて LDAP キューのサイズを設定します。
		
		次のログ メッセージが頻繁に発生し、短期的なトラフィックの急増 (電源復旧後など) と関連しない場合は、属性の値を増やすことを検討してください。
		
		`4493 DHCP ERROR "DHCP has used xx of its yy request buffers: the server is dropping a request."`
		`4494 DHCP WARNING "DHCP has used xx of yy request packets. Requests will be ignored if no packet buffers are available."`
		`5270 DHCP WARNING "DHCP has used xx of its yy request buffers: the server is congested -- will not keep the client last-transaction-time to within value but will keep it to within value seconds."`
		
		必須。デフォルトは 500 です。

詳細パラメータ	アクション	説明

max-dhcp-responses	設定/設定解除	DHCP クライアントに応答し、DHCP パートナー間でフェールオーバー通信を実行するために DHCP サーバーが割り当てる応答バッファーの数を制御します。
		
		非フェールオーバー展開では、既定の設定の 2 倍の要求バッファー数で十分です。フェールオーバー展開では、これを増やして要求バッファの 4 倍になるようにすることができます。一般に、応答バッファの数を増やしても問題はありませんが、以前に推奨された比率を下回ると、サーバーの応答性に悪影響を及ぼす可能性があります。
		
		次のログ メッセージが頻繁に発生し、短期的なトラフィックの急増 (電源復旧後など) と関連しない場合は、属性の値を増やすことを検討してください。
		
		`4721 DHCP ERROR "DHCP has used all xx response packets. A request was dropped and they will continue to be dropped if no responses are available."`
		`5289 DHCP WARNING "DHCP has used xx of yy response packets. Requests will be dropped if no responses are available."`
		
		必須。デフォルトは 1000 です。

max-ping-packets	設定/設定解除	サーバーがクライアントに対する Ping 要求を開始するために使用できるバッファの数を制御します。スコープ レベルで Ping アドレスオプションを有効にした場合、パケットバッファは ICMP メッセージの送受信に使用されます。ping を有効にした場合、ping 要求のピーク負荷を処理するのに十分な ping パケットが割り当てられている必要があります。デフォルトは 500 ping パケットです。

詳細パラメータ	アクション	説明
defer-lease-extensions	有効にする/ 無効にする	DHCP サーバーが、期限の半分未満のリースを拡張するかどうかを制御します。これは、リース状態データベースへのディスク書き込みの回数を最小限に抑えるのに役立つパフォーマンスチューニング属性です。デフォルト値はオン、または true です。つまり、途中でリースを更新するクライアントは、残りの部分だけを取得でき、延長されません。リース拡張の保留 （25 ページ）を参照してください。
last-transaction-time-granularity	設定/設定解除	last-transaction-time-granularity 属性のデフォルト値が 60 秒から 1 週間に変更されました。この新しいデフォルトは、クライアントと最後のトランザクション時間が、クライアントが最後にサーバーと通信した時刻を正確に反映していない可能性があることを意味します。 クライアントがサーバーに通信するたびに更新されるこの属性に展開が依存する場合、展開に対して適切な値に last-transaction-time-granularity 属性を明示的に設定する必要があります。 defer-lease-extensions を無効にすると、last-transaction-time-granularity 属性を効率よく使用できません。したがって、遅延リース延長を無効にした場合、デフォルト値の変更は影響を受けません。 サーバーの負荷が高く、要求バッファまたは応答バッファが不足している場合、負荷を軽減するためにサーバーは一時的に last-transaction-time-granularity の値を 1 年に設定します。

詳細パラメータ	アクション	説明
discover-queue-limit	設定/設定解除	DHCPDISCOVER および SOLICIT クライアント要求にいつでも使用できる要求バッファの割合の制限を指定します。要求バッファの構成された割合を超えると、追加の DHCPDISCOVER および SOLICIT クライアント要求は破棄されます。DHCPDISCOVER/SOLICIT 要求で使用できる要求バッファを制限することにより、サーバーは DHCPREQUEST/REQUEST 要求を処理するために使用可能な要求バッファを持っていることを保証し、これにより、電源復旧や CMTS の再起動後など、アクティビティのスパイク中にクライアントをオンラインにするのに必要な時間を大幅に短縮できます。 DRL（Discriminating Rate Limiter）属性は、判別レートリミッタの機能を制御します。判別レート リミッターはデフォルトで有効になっており、DHCP サーバーが新しいトランザクションの起動よりも DHCP トランザクションの完了を優先します。多くの場合、これによりすべてのクライアントをオンラインにする時間が短縮されます。アクティビティの要約ロギングが有効になっている場合、レート制限のためにドロップされた DHCPDISCOVER (DHCPv4) および送信請求 (DHCPv6) パケットの数は DRL:number として報告されます。 DHCPv4 統計には、新しいキュー限定検出ドロップ・カウンターが含まれており、DHCPv6 統計には新しいキュー制限付き送信請求ドロップ・カウンターが含まれています。これらのカウンタは、ドロップされたパケットを監視するために使用されます。

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

ステップ 1 Deploy メニューで、[DHCP] サブメニューから DHCP Server を選択し、[DHCPサーバーの管理（Manage DHCP Server）] ページを開きます。

ステップ 2 [DHCPサーバー（DHCP Server）] ペインからサーバーを選択します。

ステップ 3 [ローカルDHCPサーバーの編集（Edit Local DHCP Server）] ページで属性を追加または変更します。
ステップ 4 変更を加えた後に Save をクリックします。

CLI コマンド

現在のサーバーパラメータを表示するには、 dhcp show および dhcp get attribute を使用し、 dhcp set attribute=value [attribute=value ...]、dhcp unset attribute、dhcp enable attribute、および dhcp disable attribute を使用してこれらを変更します（上の表を参照してください）。

リース拡張の保留

遅延リース拡張属性 (事前設定値) を有効にすると、DHCP サーバーは DHCP トラフィックの突然のフラッディングに対する応答を最適化できます。このようなトラフィックスパイクが発生する可能性のあるネットワーク イベントの例としては、ケーブル インターネット サービス プロバイダ (ISP) データ センターで電源障害が発生し、その結果、すべてのケーブル モデム 終端システム (CMTS) が一度にリブートする場合があります。この場合、CMTS に接続されたデバイスは、すぐにオンラインに戻ってくると、DHCP トラフィックが大量に生成されます。

遅延リース拡張属性が有効になっていると、DHCP サーバーは、通常は T1 の前 (通常はリースの途中で) 前に発生するクライアントの更新要求のリースの有効期限の延長を延期する可能性があります。クライアントに構成されたリース時間を完全に設定する代わりに、サーバーは既存のリースの残りの時間を許可します。リースの有効期限の絶対値は変更されないため、サーバーは、サーバーのスループットが大幅に向上するデータベース更新を回避できます。もう 1 つの利点は、リースの有効期限を延長してフェールオーバー パートナーを更新する必要がないようにすることです。

クライアントが T1 以降 (通常は有効期限の途中) にある場合、この属性を有効または無効にしても効果はなく、サーバーは常にリース有効期限の延長を試みます。ただし、フェールオーバーやその他のプロトコル制限により、サーバーが構成された時間の間、リースを延長できなくなる可能性があります。

（注） リース拡張を延期すると、DHCP RFC に準拠したままサーバーのパフォーマンスが大幅に向上します。

リース拡張を延期する場合は、ポリシー属性のリース時間の変更をデフォルトの無効のままにするか、有効な場合は無効に変更することをお勧めします。

サーバーの観点から、次の 3 つの状況について説明します。

- Clientサーバーが遅れると、クライアントが要求を再送信することがretries可能です。DHCP サーバーは、これらの情報を再送信として認識するのに十分な情報を保持しておらず、それぞれが完了するまで処理を行い、再び完全なリース期間を与え、データベースを更新します。サーバーが既に遅れている場合、余分な作業を行うと状況が悪化します。これを防

ぐために、DHCP サーバーは、遅延リース拡張属性の状態に関係なく、30 秒未満のリースを延長しません。

- Client：クライアントリースの有効な更新時間は、設定された更新時間とクライアントの再起動間隔の最小値rebootsです。多くのインストールでは、更新時間が何日も設定されている場合でも、クライアントは 1 日あたり 1 回または 2 回 (一般的なケーブル ネットワークで) 新しいリースを取得する場合があります。遅延リース拡張属性を設定すると、これらの早期更新がデータベース トラフィックを引き起こさないようにすることができます。
- —DHCP サーバーがリースに関して DHCP クライアントにプロアクティブに接続する方法がないため、DHCP サーバーで短いリース時間を設定して、ネットワークの番号変更、アドレスの再割り当て、またはネットワークの再構成(DNS サーバーアドレスの変更など)をタイムリーに行う手段を提供できます。 Artificially short renewal times 目標は、許容できないデータベース更新のオーバーヘッドを発生させることなく、これを行えるようにすることです。

複雑な問題として、サーバーはクライアントから最後に聞こえた時刻も追跡します。最後のトランザクション時間と呼ばれるサイトでは、デバッグの補助としてこの情報を使用することがあります。この時間を維持するには、クライアントとのやり取りごとにデータベースへの書き込みが必要です。最後のトランザクション時間粒度属性は、設定する属性です。(表 1 : DHCP の詳細属性の属性の説明を参照してください)。これは主にデバッグ支援であるため、値は完全に正確である必要はありません。さらに、インメモリ コピーは常に正確であるため、データベース内のexport leases –serverデータが最新でない場合でも、現在の情報を表示するために使用できます。

DHCP 転送の設定

Cisco Prime Network レジストラー DHCP サーバーは、クライアントごとに別の DHCP サーバーへの DHCP パケットの転送をサポートします。たとえば、特定の MAC アドレス プレフィックスを持つ特定のクライアントからのアドレス要求を別の DHCP サーバーにリダイレクトする場合があります。これは、転送先のサーバーが管理するサーバーではない場合に役立ちます。これは、複数のサービス プロバイダが同じ仮想 LAN 上のクライアントに DHCP サービスを提供する環境で発生します。

DHCP 転送を有効にするには、拡張スクリプトを実装する必要があります。DHCP サーバーは、指定されたクライアントをインターセプトし、転送コードを呼び出し、転送されたサーバー アドレスの指定されたリストをチェックします。その後、要求自体を処理するのではなく、要求を転送します。dhcpattachExtensionをdhcp使用して、DHCP サーバーとの間で拡張機能を接続およびデタッチdetachExtensionします。

DHCP 転送機能は次のように機能します。

1. DHCP が初期化されると、サーバーは UDP ソケットをオープンし、それを使用して転送されたパケットを送信します。複数の IP アドレスを持つサーバーをサポートするために、ソケット アドレスのペアは、INADDR_ANY と任意のポート番号で構成されます。これにより、クライアントはサーバーの IP アドレスのいずれかを使用できます。

2. DHCP サーバーは、クライアントから要求を受信すると、次の拡張ポイント スクリプトを処理します。

   • post-packet-decode
   • pre-client-lookup
   • post-client-lookup

   DHCP サーバーはこれらのスクリプトを処理するに従って、次の文字列の環境ディクショナリをチェックします。

   ```
   cnr-forward-dhcp-request
   ```

3. その文字列が見つかったとき、値true (有効) を持つサーバーは、その転送コードを呼び出します。
4. 転送コードは、次のキーを持つ文字列の環境ディクショナリをチェックします。

   ```
   cnr-request-forward-address-list
   ```

   この例のように、コロンで区切られたポート番号を使用して、コンマで区切られたIPアドレスのリストが必要です。

   ```
   192.168.168.15:1025,192.168.169.20:1027
   ```

   デフォルトでは、サーバーは DHCPv4 のサーバーポートと DHCPv6 の v6 サーバーポートに転送します。クライアント要求全体のコピーを各IPアドレスとポートに順番に送信します。リスト内の要素のいずれかが無効な場合、サーバーはリストの解析を停止します。
5. 転送コードが戻った後、サーバーは要求の処理を停止します。ただし、クライアント参照後の拡張ポイント スクリプトでは、この操作によって、クライアント エントリの詳細を含むオプションのログ メッセージが作成される場合があります。

   次のTCL拡張スクリプトの一部の例では、要求の情報に基づいて別のサーバーに要求を転送するようにDHCPサーバーに指示します。同じ環境に複数のデバイス プロビジョニング システムがある場合は、このようなスクリプトを使用できます。この場合、ルーターがブロードキャスト要求を転送するDHCPサーバーで拡張スクリプトを実行します。スクリプトは、他のサーバーが要求を処理する必要がある場合は、その要求を処理する必要がある場合、その要求を転送するように元のサーバーに指示します。

   サンプル スクリプトでは、MAC アドレス プレフィックスの静的マッピングを使用して、特定のベンダーから特定のシステムにモデムを送信します。

```
proc postPktDecode {req resp env} {
  set mac [$req get chaddr]
  set addrs ""
  # Very simple, static classifier that forwards all requests from devices
  # with a mac-address vendor-id of 01:0c:10 to the DHCP servers at
  # 10.1.2.3 and 10.2.2.3:
  switch -glob -- $mac {
    01:0c:10* {
      set addrs "10.1.2.3,10.2.2.3"
    }
  }
  # If we decide to forward the packet, the $addrs var will have the IP
  # addresses where to forward the packet:
  if {$addrs != ""} {
    # Tell the DHCP server to forward the packet...
    $env put cnr-forward-dhcp-request true
    # ...and where to forward it:
```

```
        $env put cnr-request-forward-address-list $addrs
        # No more processing is required.
        return
      }
    }
```

より柔軟なスクリプトでは、Cisco Prime Network レジストラークライアントエントリなどのクライアントごとの設定オブジェクトを使用して、どのDHCPサーバーが要求を取得するかを指定できます。

（注） DHCP 転送は DHCPv4 でのみ使用できます。DHCPv6 用ではありません。

DHCPv6 サーバー属性の編集

DHCPv6 に関連する DHCP サーバー属性を編集できます。これらの属性は次のとおりです。

- v6-client-class-lookup-id：DHCPv6 クライアント要求に基づいて client-class を決定し、設定済みの client-class の名前または <none>（式で client-class を指定しない場合）の文字列を返す式。属性にはプリセット値がありません。
- max-client-leases：DHCPv6 クライアントがリンクで保持できるリースの最大数。この属性を使用して、クライアントを 1 つのリースのみに制限しないでください。プリセットは 50 です。

[ローカル基本（Basic）] または [アドバンスド（Advanced）] Web UI

[展開（Deploy）] メニューの [DHCP] サブメニューの下で DHCP Server を選択し、[DHCPサーバーの管理（Manage DHCP Server）] ページを開きます。[ローカルDHCP サーバー（Local DHCP Server）] リンクをクリックして [DHCPサーバーの編集（Edit DHCP Server）] ページを開き、前述の DHCPv6 属性値を変更して、 Save をクリックします。

CLI コマンド

dhcp を使って前述の DHCPv6 サーバー属性を表示し、dhcp set attribute=value [attribute=value ...] を使用して変更します。

DHCP サーバーの動作に影響を与える拡張機能の使用

Cisco プライムネットワーク レジストラーは、拡張、プログラムを通じて、TCL または C/C++ で記述できる DHCP サーバーの動作を変更およびカスタマイズする機能を提供します。拡張機能は、要求パケットまたは応答パケットを変更し、環境ディクショナリに保存されている環境

変数を使用して、サーバーと対話します（詳細は、拡張ポイントの使用 （433 ページ） を参照してください）。

たとえば、BOOTP 構成を使用する異常なルーティング ハブがある場合があります。このデバイスは、イーサネット・ハードウェア・タイプ (1) および MAC アドレスを指定した BOOTP 要求をchaddrフィールドに出します。その後、同じ MAC アドレスを持つ別の BOOTP 要求を送信しますが、ハードウェアタイプはトークンリング (6) です。通常、DHCP サーバーは、ハードウェア タイプ 1 の MAC アドレスとタイプ 6 の MAC アドレスを区別し、異なるデバイスであると見なします。この場合は、DHCP サーバーが同じデバイスに対して 2 つの異なるアドレスを配布することを防ぐための拡張機能を作成する必要があります。

次のいずれかの拡張を記述することで、2 つの IP アドレスの問題を解決できます。

- DHCP サーバーがトークン リング (6) ハードウェア タイプ のパケットをドロップする原因となるもの。
- トークン リング パケットをインターネット パケットに変更し、終了時に再度切り替えるパケット。この拡張は複雑になりますが、DHCP クライアントは DHCP サーバーからのリターンを使用できます。

関連項目

拡張機能の作成 （29 ページ）

拡張機能を使用した通信量の多いクライアントの防止 （31 ページ）

拡張機能の作成

TCL または C/C++ で拡張機能を記述できます。

- TCL—拡張機能を書き込むのが少し簡単で迅速になります。拡張が短い場合、TCL の解釈された性質はパフォーマンスに重大な影響を与えません。TCL で拡張機能を記述すると、サーバーをクラッシュさせる可能性のあるバグが発生する可能性が低くなります。
- C/C++：外部プロセスとの通信を含む、可能な限り最大のパフォーマンスと柔軟性を提供します。ただし、C/C++ API の複雑さが増し、拡張機能のバグがサーバーをクラッシュさせる可能性が TCL よりも高くなります。

特定の拡張ポイントで拡張機能を作成します。拡張ポイントには、要求、応答、環境という 3 種類のディクショナリが含まれています。これらの辞書の 1 つ以上は、次の拡張ポイントごとに使用できます。

1. init-entry：拡張ポイントは、DHCP サーバーが拡張機能を設定または構成解除するときに呼び出します。これは、サーバーの起動、停止、または再ロードのときに発生します。このエントリ ポイントのシグネチャは、拡張機能の他のエントリ ポイントと同じです。DHCPv6 処理に必要です。辞書:環境のみ。
2. pre-packet-decode：最初の拡張ポイントは、要求が到着したときに DHCP サーバーが検出し、パケットをデコードする前に呼び出します。辞書: 要求と環境。
3. post-packet-decode：入力パケットを書き換えます。使用するディクショナリは要求と環境です。

4. post-class-lookup：クライアントクラスに対する client-class-lookup-id 操作の結果を評価します。使用するディクショナリは要求と環境です。
5. pre-client-lookup：検索を行うクライアントに影響を与えます(検索を禁止したり、既存のデータを上書きするデータを提供したりすることなど)。使用するディクショナリは要求と環境です。
6. post-client-lookup：クライアント クラスの処理から入力された内部サーバー データ構造を調べるなど、クライアントクラスのルックアッププロセスの動作を確認します。DHCP サーバーが追加の処理を行う前に、この機能を使用してデータを変更することもできます。使用するディクショナリは要求と環境です。
7. generate-lease：DHCPv6 アドレスまたはプレフィックスを生成および制御します。辞書: 要求、応答、および環境。
8. check-lease-acceptable：リース受入テストの結果を変更します。細心の注意を払って行ってください。使用するディクショナリは、要求、応答、環境です
9. lease-state-change：リース状態が、これを変更するタイミングを細心の注意を払って決定します。辞書:応答と環境。
10. pre-packet-encode：応答で DHCP クライアントに返送されるデータを変更するか、DHCP 応答を送信するアドレスを変更します。使用するディクショナリは、要求、応答、環境です
11. post-packet-encode：サーバーがパケットをクライアントに送信する前、またはパケットをドロップする前に、サーバーがパケットを検査して変更できるようにします。使用するディクショナリは、要求、応答、環境です
12. post-send-packet：DHCP 要求/応答サイクルの重大な時間制約の外部で実行する処理用のパケットを送信した後に使用されます。使用するディクショナリは、要求、応答、環境です
13. environment-destructor：エクステンションが保持している可能性のあるコンテキストをクリーンアップできます。使用するディクショナリは環境です。

DHCP サーバーを拡張するには、次の手順を実行します。

ステップ 1 Tcl、C、または C++ で拡張機能を記述し、サーバー拡張ディレクトリにインストールします。

- Tcl—/var/nwreg2/local/extensions/dhcp/tcl
- C or C++—/var/nwreg2/local/extensions/dhcp/dex

これらの拡張機能は、TCL または C/C++ 拡張用の適切なディレクトリに配置するのが最適です。次に、ファイル名を構成するときに、ファイル名自体をスラッシュ（/）を使用せずに入力します。

サブディレクトリに拡張子を配置する場合は、パス区切り記号を付けてファイル名を入力します。

（注） 作成した拡張機能は、/var/nwreg2/local/extensions/dhcp/... エリアに追加する必要があります。Cisco Prime Network Registrar に同梱されている拡張機能は、/opt/nwreg2/local/extensions/dhcp エリアにあります。サーバーは、最初に /var/nwreg2/local/extensions/dhcp/... ディレクトリで拡張機能を検索し、次に /opt/nwreg2/local/extensions/dhcp/... ディレクトリで拡張機能を検索します。

ステップ 2 Web UI の [DHCP拡張の一覧/追加（List/Add DHCP Extensions）] ページを使用するか（[詳細設定（Advanced）] モードで [展開（Deploy）] メニューから、[DHCP] サブメニューの下の [拡張（Extensions）] を選択して [DHCP拡張の一覧/追加（List/Add DHCP Extensions）] ページを開きます）、または CLI の**拡張**コマンドを使用して、DHCP サーバーがこの拡張機能を認識するように構成します。

ステップ 3 dhcp attachExtension を使用して、構成済みの拡張機能を 1 つ以上の DHCP 拡張ポイントに接続します。

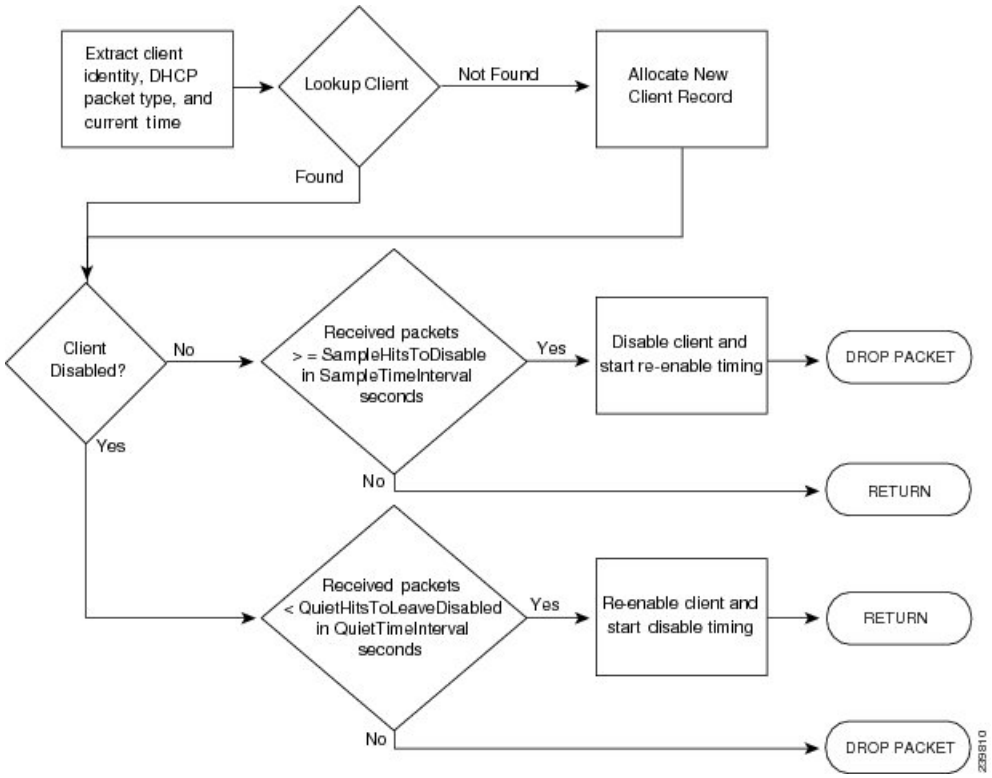
ステップ 4 サーバーをリロードします。

拡張機能を使用した通信量の多いクライアントの防止

拡張の効果的な使用方法の一例として、不要なトラフィックでサーバーがオーバーフローしているクライアントから保護することが挙げられます。ChattyClientFilter 拡張機能を使用すると、これらのチャット クライアント パケットを処理する作業の多くをサーバーが行う必要がないようにすることができます。ネットワーク内に多数のクライアントがある場合は、この拡張機能の実装を検討してください。

ChattyClientFilter 拡張機能は、 Cisco Prime Network Registrar インストールの インストールパス/examples/dhcp/dex ディレクトリで使用可能であり、コンパイル済みのため install-path/extensions/dhcp/dex/dexextension.so または install-path/extensions/dhcp/dex/dexextension.dll ですぐに使用できます。この拡張は、MAC アドレスに基づいてクライアント要求をモニターリングし、ある間隔で特定数を超えるパケットを生成したクライアントを無効にします。あるクライアントを無効にすると、サーバーではそのパケットが破棄されます。ただし、サーバーはそのクライアントのトラフィックを引き続きモニターリングするため、完全に無視するわけではありません。特定の間隔でクライアントが生成するパケット数が特定数を下回り始めたことをサーバーが検出すると、クライアントが再度有効になり、パケットの受信が再開されます。

図 *8:* チャット クライアント フィルタ フロー



クライアントを無効および再度有効にする基準は、ChattyClientFilter拡張の引数で設定します。デフォルトでは、サーバーは30秒以内に15個を超えるパケットを受信すると、クライアントを無効にします。サーバは、10秒以内に送信するパケットが5個未満の場合に、クライアントを再び有効にします。これらのデフォルト値は控え目な値であるため、すべての状況が保護されるわけではありませんのでご注意ください。たとえば、サーバーは3秒ごとにパケットを送信するクライアントを無効にはしません。数回の再送信があったとしても、クライアントが6個を超えるパケットを短い間隔で送信する必要がないためです。

チャット クライアントが疑われる場合は、DHCP サーバー ログを確認して着信レートを確認し、次の表に示す引数を ChattyClientFilter コードに適切に設定します。

表 *2:* 引数を使用します。

チャティクライアントフィルタ引数	説明
–c	環境ディクショナリのdrop属性が「true」に設定されている場合、パケットを無視します。デフォルトでは無視されません。

チャティクライアントフィルタ引数	説明
–d packet-count seconds	指定された時間間隔で指定された数を超える数を受信した場合は、DHCPRELEASE パケットを破棄します。デフォルトでは無効になっています。 サーバーはクライアントが指定された間隔でパケットの送信を中断するまで DHCPRELEASE パケットをドロップし続けます（DHCPv4 クライアントのみ）。 基本的な式は、時間間隔は少なくとも(パケットカウント+ 2)*30秒でなければならないということです。
–h packet-count	SampleHitsToDisable：デフォルトでは 15 パケットです。
–i seconds	SampleTimeInterval：デフォルトでは 30 秒です。
–l packet-count	QuietHitsToLeaveDisabled：デフォルトでは 5 パケットです。
–m seconds	クライアントが無効になる最大時間を秒単位で設定します。デフォルト 0 ～無制限。
–n	更新または再バインドする場合はクライアントをNAKsします。デフォルトではオフになっています。クライアントが、サンプルヒットを超える場合、無効にするクライアントは、DHCP要求を行う、サーバーはパケットを破棄する代わりに DHCPNAK を送信します。 これにより、何らかの理由でリースを更新できないクライアント(ケーブル モデムなど)の問題を解決できます。DHCPNAK を送信すると、クライアントは DHCP ステート マシンを再起動して DHCPDISCOVER を送信します。 この引数を使用する場合は、拡張ポイントに ChattyClientFiltercheck-lease-acceptableをアタッチする必要があります。（DHCPv4 クライアントのみ）。
–q seconds	QuietTimeInterval：デフォルトでは 10 秒です。
–r seconds	StatisticsInterval：デフォルトでは 300 秒（5 分）です。この引数は、無効にして再び有効にしたクライアント数の定期的なログ記録の頻度を制御します。
-s	ドロップされたパケットをサイレントに破棄します。デフォルトではオフになっています。

チャティ クライアントフィルタ引数	説明
-t	指定した場合、チャッティ クライアント フィルタは DHCPv6 クライアントの DUID-LLT の「時刻」を無視します（クライアント ID オプション（1））。フィルターは、DUID-LLT 内の時間をゼロ（00:00:00:00）に変更します。 これは、DUID-LLT を使用するクライアントがこの値を正しく保存しないため、複数のクライアント ID を生成しない場合に使用できます。 **注**：これは、サーバーがこれらのクライアントを処理する方法を変更しません。
–w port	指定したポートで Web アクセスを有効にします。正のポート番号を指定すると、接続はローカルホストに制限されます。負のポート番号を指定した場合は、ポートの絶対値が使用され、接続が制限されません。
-4	DHCPv4 パケットのみをフィルタします。デフォルトは両方です。
–6	DHCPv6 パケットのみをフィルタします。デフォルトは両方です。

（注） h、-i、-l、および-q のデフォルトは、単一のタイプの不適切なクライアントに対処するように設計されていたため、ほとんどの状況に適している可能性は低い。通常の状態に対して間隔とパケット ヒット カウントを長くすると、妥当な結果が得られます。i 120 -h 8 -q 120 -l 8 などの値は、120 秒の期間にわたってクライアントに 8 パケットを許可します。通常の DHCP ディスカバー/オファー/要求/ACK は、クライアントからのパケットが 2 パケットのみです。つまり、ChattyClientFilter を適切に使用するには、特定のネットワーク条件に合わせてこれらの値を調整する必要があります。Cisco Web サイトの Cisco Prime Network レジストラーダウンロード セクションから利用できるログスキャン ツールを使用すると、クライアントのアクティビティを分析するのに役立ちます。

引数の設定と拡張機能の有効化の詳細については、ChattyClientFilter.cpp ファイルのコメントを確認してください。ほとんどの場合、post-packet-decode 拡張ポイントにアタッチします（–n 引数を使用する場合、check-lease-acceptable も含まれます）。

ChattyClientFilter のサンプルの使用例は、DHCPv4 クライアントから送信された DHCPRELEASE パケットをドロップして、リース履歴データベースが境界外に拡張されないようにすることです。

このシナリオでは、–d引数を使用します。設定は次のようになります。

```
nrcmd> extension dexChattyClientFilter create dex libdexextension.so
dexChattyClientFilter
init-entry=dexChattyClientFilterInitEntry
init-args="-d 2 120"

nrcmd> dhcp attachextension post-packet-decode dexChattyClientFilter
```

このセットアップにより、サーバーは、120 秒間隔で同じクライアントからこれらのパケットのうち 2 つ以上を受信した場合に DHCPRELEASE パケットを廃棄し、クライアントが DHCPRELEASE を少なくとも 120 秒間送信しない場合に DHCPRELEASE 処理を再開します。

Cisco Prime Network レジストラーは、チャッティー クライアント フィルタによって監視または無効になっている(ドロップされるトラフィック)クライアントに関する情報を取得するために使用できるミニ Web サーバーをサポートします。一般的な要求は、http://127.0.0.1:<port>/web ブラウザーに入力されたレポートです。

Web サーバーは、次の要求をサポートします。

- status：統計レポートを返します。
- report：統計レポートと完全なクライアントレポートを返します。クライアント レポートには、現在監視されているすべてのクライアントと無効になっているクライアントが含まれます。
- disabled-report：レポートと同じですが、無効なクライアントのみが返されます。
- flush：レポートと同じですが、すべてのクライアントが内部監視および無効リストから削除されます。
- csv-client-list：（監視および無効のクライアントを含む）CSV 形式を使用してクライアント リストを返します。
- csv-disabled-client-list：csv クライアントリストと同じですが、現在無効になっているクライアントのみが含まれます。
- xml-client-list：XML を使用してクライアント リストを返します（監視対象クライアントと無効なクライアントを含む）。
- xml-disabled-client-list：XML を使用して無効なクライアントリストを返します。

（注） この Web サーバーは非常に基本的なサーバーの実装です。上記のリクエストのみをサポートします。

DHCP サーバーの調整

DHCP パフォーマンスを調整するうえで役立つヒントは、次のとおりです。

- 最適なスループットを得るための要求 (最大 dhcp-request) および応答 (max-dhcp-responses) バッファを設定します。詳細については、表 1 : DHCP の詳細属性 を参照してください。
- 遅延リース拡張属性を有効にします。これにより、データベースへの書き込みが減少します。
- 最後のトランザクション時間粒度属性を、リース間隔の半分より大きい値として、最低60秒以上に設定します。
- プロダクション リースを提供するポリシーのリース時間の優先許可属性を無効にします。

- ログ記録とデバッグの設定を最小限にします。ログ記録が必要な場合は、次の表に示すように、制御された数の属性を持つ DHCP サーバーのlog-settings属性を使用します。

表 *3 : DHCP* ログの設定

ログ設定 (数値同等)	説明
default (1)	DHCP サーバーのいくつかの部分で低レベルのログを提供します。デフォルトを設定解除すると、このログは表示されません。
activity-summary (20)	サマリー メッセージを 1 分ごとに表示します。これは、no-xxx ログ設定の多くが有効になっているときに、各 DHCP メッセージに対応するログ メッセージに必要な負荷を課すことなく、サーバーでのアクティビティを把握するのに役立ちます。これらのメッセージの期間は、 DHCP サーバー プロパティのアクティビティの概要 - 間隔を使用して構成できます。
client-criteria-processing (9)	有効なリースを見つけるためにスコープが検査される場合、またはスコープを調べて、既にリースを持っているクライアントに対して引き続きリースが許容できるかどうかを判断するたびに、ログ メッセージが出力されます。クライアント クラススコープの条件の処理を構成またはデバッグする場合に非常に便利です。これは、ログに記録される情報の適度な量を引き起こし、当然として有効にしておく必要があります。
client-detail (8)	すべてのクライアント クラスクライアントルックアップ操作の終了時に、単一行をログに記録します。この行には、クライアントに対して検出されたデータと、クライアントのクライアント クラスで検出されたデータの合成が表示されます。クライアント クラスの構成をセットアップする場合や、クライアント クラス処理で問題をデバッグする場合に便利です。
dns-update-detail (7)	サーバーが各 DNS 更新を送信し、更新メッセージへの応答を受信する場合に、メッセージをログに記録させます。
dropped-waiting-packets (15)	最大待機パケットの値がゼロ以外の場合、IP アドレスのキュー長がmax-waiting-packetsの値を超えると、パケットがドロップされることがあります。ドロップ待機パケットが設定されている場合、サーバーは、IP アドレスのキューから待機パケットをドロップするたびにメッセージをログに記録します。
failover-detail (10)	ほとんどのフェイルオーバー・トランザクションで、サーバーが単一のメッセージをログに記録するようにします。ログに記録される情報は、フェールオーバーがどのように動作しているかを理解するのに非常に役立ちます。

ログ設定 (数値同等)	説明
incoming-packet-detail (4)	DHCP サーバーが受信したすべての DHCP パケットの内容が、人間が判読可能な方法で解釈され、ログ ファイルに出力されます。これにより、入力パケットに対して組み込みの DHCP パケット スニッファが有効になります。この設定が有効になると、ログ ファイルがいっぱいになり、非常に急激に切り替わります。この設定は DHCP サーバーのパフォーマンスに大きな影響を与えるため、当然のことながら有効にしておく必要はありません。
incoming-packets (2)	この設定 (既定ではオン) では、着信パケットごとに 1 つの回線メッセージが記録されます。これは、DHCP サーバーまたは BOOTP リレーを最初に構成する場合に、DHCP サーバーがパケットを受信していることを即座に肯定する機能が存在する場合に特に便利です。
ldap-create-detail (13)	DHCP サーバーが LDAP サーバーに対してリース状態エントリの作成または削除を開始し、応答を受信し、結果メッセージまたはエラー メッセージを取得するたびに、ログ メッセージが表示されます。
ldap-query-detail (11)	DHCP サーバーが LDAP サーバーに対する照会を開始し、応答を受信し、結果またはエラー・メッセージを取得するたびに、ログ・メッセージが表示されます。
ldap-update-detail (12)	DHCP サーバーが LDAP サーバーに対する更新リース状態を開始し、応答を受信し、結果メッセージまたはエラー メッセージを取得するたびに、ログ メッセージが表示されます。
leasequery (14)	leasequery パケットが内部エラーなしで処理され、ACK または NAK が発生したときにログ メッセージが表示されます。
minimal-config-info (24)	サーバーの始動時または再ロード時に印刷される構成メッセージの数を削減します。特に、すべてのスコープについてメッセージをログに記録するわけではありません。
missing-options (3)	この設定 (既定ではオン) は、DHCP クライアントによって要求されたオプションがポリシーで構成されていないため、DHCP サーバーから提供できない場合に、メッセージがログに記録されます。
no-dropped-bootp-packets (18)	ドロップされたすべての BOOTP パケットに対して通常ログに記録される単一行メッセージが表示されないようにします。
no-dropped-dhcp-packets (17)	DHCP 設定が表示されないためにドロップされたすべての DHCP パケットに対して、通常は 1 行メッセージが記録されます。(無効であるためにドロップされたパケットに関連付けられたメッセージについては、無効なパケットがないを参照してください)。

ログ設定 (数値同等)	説明
no-failover-activity (19)	通常のアクティビティと、フェールオーバー用に記録された警告メッセージが表示されないようにします。重大なエラー ログ メッセージは、このログ設定とは無関係に表示され続けます。
no-failover-conflict (25)	フェールオーバー パートナー間の競合がログに記録されない原因になります。
no-invalid-packets (21)	無効であるために廃棄された DHCP パケットごとに、通常は 1 行のメッセージが記録されます。(DHCP サーバーの構成によってドロップされたパケットに関連付けられたメッセージについては、ドロップなしの dhcp パケットを参照してください)。
no-reduce-logging-when-busy (22)	通常、DHCP サーバーは、非常にビジー状態になったとき (つまり、使用可能な受信バッファの 2/3 以上を使用した場合 (それ自体は設定可能な値)、ログを減らします。成功メッセージ、ドロップなし dhcp パケット、ドロップなし bootp パケット、非フェイルオーバ アクティビティ、無効なパケットを設定し、アクティビティサマリー以外のすべてをクリアします。ログ記録アクティビティを設定しない場合、サーバーはこれを行いません。サーバーがビジー状態になくなったとき (つまり、使用可能な受信バッファの 1/3 しか使用していない場合) に、以前の設定が復元されます。
no-success-messages (16)	正常な発信 DHCP 応答パケットごとに通常ログに記録される単一行メッセージが表示されないようにします。正常な発信 DHCP 応答パケットの場合にのみ、ログに影響します。
no-timeouts (23)	リースまたはオファーのタイムアウトに関連付けられたメッセージがログ ファイルに表示されないようにします。
outgoing-packet-detail (5)	DHCP サーバーによって送信されるすべての DHCP パケットの内容が、人間が判読可能な方法で解釈され、ログ ファイルに出力されます。これにより、出力パケットに対して組み込みの DHCP パケット スニッファが有効になります。この設定が有効になると、ログ ファイルがいっぱいになり、非常に急激に切り替わります。この設定は DHCP サーバーのパフォーマンスに大きな影響を与えるため、当然のことながら有効にしておく必要はありません。
unknown-criteria (6)	クライアント エントリが見つかったときに、そのクライアントの現在のネットワークの場所に適したスコープに見つからない選択条件を指定するたびに、単一行のログ メッセージが表示されます。

ログ設定 (数値同等)	説明
v6-lease-detail (27)	サーバーが DHCPv6 リースアクティビティに関する個々のメッセージをログに記録します (非サクセス メッセージ、またはタイムアウトなしに応じてクライアント タイムアウト イベントに応じて、クライアントトランザクションごとに1つのメッセージに加えて、または代わりに)。

- クライアント キャッシュの設定を検討してください（「クライアントのキャッシュ パラメータの設定 （366 ページ）」を参照してください）。
- サーバーのパフォーマンスを監視するためにサーバーの統計情報を確認します（『Cisco プライムネットワーク レジストラー 11.0 管理ガイド』の「統計の表示」セクションを参照）。
- スコープ割り当ての優先順位を設定することを検討してください（「割り当て優先順位を使用した複数スコープの設定 （140 ページ）」を参照してください）。
- アドレスを提供する前にホストに ping を実行する場合は、ping タイムアウト期間の調整を検討してください（アドレス提供前のホストへの ping 実行 （238 ページ）を参照）。
- パフォーマンスを向上させるには、選択タグの数を制限することを検討してください。
- ライトウェイト ディレクトリ アクセス プロトコル（LDAP）サーバーを使用している場合は、「LDAP を使用するように Cisco Prime Network Registrar を設定する （372 ページ）」で説明するパフォーマンスの問題を考慮してください。
- DHCP フェールオーバーを使用する場合は、負荷分散機能の使用を検討してください（「ロード バランシングの設定 （84 ページ）」を参照）。

ヒント DHCP サーバー属性の変更に従って、サーバーの再ロードを行ってください。

DHCP に関連するサーバーの一覧 - フェールオーバー、DNS、LDAP、TCP リスナー サーバー

関連するフェイルオーバー、DNS、LDAP、またはTCPリスナー・サーバーがある場合（フェールオーバー サーバー ペアの設定 （68 ページ）を参照）これらのサーバーの属性にアクセスできます。

ローカル Web UI

[フェールオーバー ペア（Failover Pairs）]ページで、[フェールオーバーサーバの管理（Manage Failover Servers）]タブをクリックし、[関連サーバ（Related Servers）]タブをクリックするか、

[DHCPサーバの管理（Manage DHCP Server）] ページ（[操作（Operate）] > [サーバ（Servers）] > [サーバの管理（Manage Servers）]）の [関連サーバ（Related Servers）] タブをクリックして、[DHCP関連サーバー属性（DHCP Related Server Attributes）] ページを開きます。このページには、サーバーが配置されている通信とフェールオーバーの状態が表示されます。次の表に、このページの属性を示します（このページを表示するには、dhcp-management サブロールを使用して中央 cfg-admin ロールを割り当てる必要があります）。

表 *4*: 関連サーバーの属性

関連サーバー属性	説明
関連サーバーの種類	関連サーバーの種類: DHCP、DNS、または LDAP。
関連サーバーの IP アドレス	関連するサーバーの IP アドレス。DHCP フェールオーバーパートナーの場合は、このリンクをクリックして [フェールオーバー関連サーバーの表示（View Failover Related Server）] ページを開きます。
設定	通信の状態：なし、OK、または中断。
要求	DNS または LDAP 関連のサーバーにのみ適用され、これらのサーバーからの要求の数です。
状態	DHCP フェールオーバー：なし、起動、通常、通信中断、パートナーダウン、潜在的な競合、回復、一時停止、または回復完了 高可用性 (HA) DNS の場合—送信更新、プローブ、または ha 状態不明。正常に更新されたサーバーのみが、Send-Update 状態になることができます。更新を送信していないパートナー サーバーは、常にプローブまたは不明な状態になります。クライアントの動作がない場合に DHCP サーバーが起動すると、両方の DNS サーバーが不明な状態になることが多くなります。これは、DHCP サーバーが DNS 更新を実行しようとすると変更されます。
パートナーの役割	DHCP フェールオーバーの場合のみ、パートナーのフェールオーバーロール (メインまたはバックアップ)。
パートナーの状態	DHCP フェールオーバーの場合のみ、パートナーの状態：なし、起動、通常、通信中断、パートナーダウン、潜在的競合、回復、一時停止、または回復完了
更新応答の完了	DHCP フェールオーバーの場合のみ、完了した更新応答の割合 (未解決の更新応答がある場合にのみ有効)。

表 *5*: *DHCP* 関連のフェールオーバー サーバーの属性

DHCP 関連のフェールオーバー サーバー属性	説明
General attributes	

DHCP 関連のフェールオーバー サーバー属性	説明
failover-pair-name	このサーバーを管理するために使用するフェールオーバー ペア オブジェクトの名前。
current-time	このオブジェクトを返すサーバー上の現在の時刻。
comm-state	なし、OK、または中断。
smoothed-time-delta	ローカル サーバーとパートナー サーバーの時間差。ローカル・サーバー時刻がパートナー・サーバー時刻より前にある場合、属性値は正の値になります。ローカル・サーバー時刻がパートナー・サーバー時刻より遅れている場合、属性値は負になります。サーバーが通信していない場合は、最後に確認された属性値が記録されます。
最大クライアントリードタイム	このシステム上の現在の最大クライアントリードタイム(MCLT)。
sequence-number	フェールオーバー オブジェクト全体で固有のシーケンス番号は、リース内のシーケンスと異なる場合、リースは、sf-to-date のリース フラグとは無関係に「最新ではない」と見なされます。
負荷分散バックアップ-pct	現在のフェールオーバーロードバランシングのバックアップの割合。バックアップの割合が 0 の場合、フェールオーバーの負荷分散は使用されていない (無効)。
Local server information	
our-ipaddr	このサーバーへのインターフェイスの IPv4 アドレス。
our-ip6address	このサーバーへのインターフェイスの IPv6 アドレス。
role	このオブジェクトを返すサーバーのフェールオーバー ロール (なし、メイン、またはバックアップ)。
state	ローカルサーバーの状態：なし、起動、通常、通信- 中断、パートナーダウン、潜在的な競合、回復、一時停止、または回復完了

DHCP 関連のフェールオーバー サーバー属性	説明
start-time-of-state	現在のフェールオーバー状態が開始された時刻。
start-of-comm-interrupted	このパートナーが通信中断状態に入った時刻。これはリロード全体で有効ですが、開始時点の状態が最新のサーバーのリロードよりも早い時間を持つことはありません。
est-end-recover-time	update-request-in-progress がないに設定されている場合に有効です。表示された場合、更新要求が完了した場合にサーバーがリカバリ完了状態に入る時刻。表示されない場合、更新要求が完了するたびにサーバーはリカバリ完了に入ります。
use-other-available	false または未設定の場合、このサーバーは他の使用可能なリースを使用できません。true の場合、サーバーは他の使用可能なリースを使用できます。常に有効ですが、パートナーダウン状態の場合にのみ true にする必要があります。
use-other-available-time	パートナーダウン状態で、使用可能な使用が偽または未設定の場合、使用する他の使用可能な時間は true になります。
safe-period-remaining	セーフ期間の残り時間 (秒単位)。0 に設定されていない場合、このサーバーは現在、パートナーに対して安全な期間を実行しています。
load-balancing-local-hba	ローカルサーバーの現在のハッシュバケット割り当て(通常はハッシュバケット番号の範囲として表示されます)。(RFC 3074 を参照してください。
request-buffers-in-use	統計情報の計算時に DHCP サーバーが使用しているフェールオーバー要求バッファの数。
decaying-max-request-buffers-in-use	最近使用されたフェイルオーバー要求バッファーの最大数。
request-buffers-allocated	フェールオーバー機能をサポートするためにサーバーが割り当てた要求バッファの数。

DHCP 関連のフェールオーバー サーバー属性	説明
connection-start-time	最新の接続が開始された時刻。この値は、接続が開始されるたびに設定され、接続が終了してもクリアされません。
connection-end-time	最新の接続が終了した時刻。この値は、接続が終了するたびに設定され、新しい接続が開始されたときにクリアされません。
Partner server information	
ipaddr	パートナー サーバーの IP アドレス。
ip6address	パートナー サーバーの IPv6 アドレス。
partner-role	このオブジェクトを返すサーバーのパートナーのフェールオーバー ロール (なし、メイン、またはバックアップ)。
partner-state	フェールオーバー 関係のパートナーの最後の状態：なし、スタートアップ、ノーマル、通信中断、パートナーダウン、潜在的な競合、回復、一時停止、または回復完了
start-time-of-partner-state	パートナーの現在のフェールオーバー状態が開始された時刻。
est-partner-end-recover- time	パートナーの状態が[回復]の場合、パートナーが MCLT をタイムアウトして回復状態を終了する時期の予測値。
last-comm-ok-time	このサーバーが最後に通信が OK であると判明した時刻。
load-balancing-partner- hba	パートナーサーバーの現在のハッシュバケット割り当て(通常はハッシュバケット番号の範囲として表示されます)。（「RFC 3074」を参照）。
partner-vendor-major- version	パートナー サーバーからのベンダー ID メジャー バージョン。
partner-vendor-minor- version	パートナー サーバーからのベンダー ID マイナー バージョン。
Update requests sent to partner	

DHCP 関連のフェールオーバー サーバー属性	説明
update-request- outstanding	None または unset の場合、サーバーはパートナーのキューに更新要求を持っていません。Noneに設定されていない場合、フェールオーバー パートナーの更新要求がキューに入れられます。有効な値は、なし、更新、および更新すべてです。
update-request-start-time	update-request-outstanding 要求が開始された時刻。
update-request-done-time	更新要求の最後の完了時刻。
v6-update-response-in-progress	応答の種類と発生元。
v6-update-response-percent-complete	現在の IPv6 更新の応答の完了率。
v6-update-response-start-time	v6 更新-進行中の応答で言及された IPv6 更新応答が開始された時刻。
v6-update-response-done-time	最新の IPv6 更新応答がパートナー サーバーに対して行われた更新を送信した時刻。
Update requests processed for partner	
update-response-in- progress	このサーバーが更新応答を処理している場合は、応答の種類と発生元に関する情報を提供します。
update-response-percent- complete	更新-応答の未解決が表示された場合は、現在の更新の応答の達成率。
update-response-start- time	更新応答の進行中に記載された更新応答が開始された時刻。
update-response-done- time	最新の更新の応答が、パートナー サーバーに対して行われた更新を送信した時刻です。
Load Balancing Counters	
load-balancing-processed- requests	負荷分散の対象となる、IPv4 と IPv6 の両方のサーバー処理要求の数。このカウンタには、サーバーから NORMAL 状態への最新の遷移後に行われた要求のみが含まれます。
load-balancing-dropped- requests	負荷分散の対象となる、IPv4 と IPv6 の両方のサーバー ドロップ要求の数。このカウンタには、サーバーが Normal 状態に遷移した最新の移行後に行われた要求のみが含まれます。

DHCP 関連のフェールオーバー サーバー属性	説明
load-balancing-processed- total	サーバーが処理した IPv4 と IPv6 の両方の要求のうち、ロード バランシングの対象となるものの数。このカウンタには、このサーバーが最後に開始または再ロードされてからの要求が含まれます。
load-balancing-dropped- total	サーバーがドロップした IPv4 と IPv6 の両方の要求のうち、ロード バランシングの対象となるものの数。このカウンタには、このサーバーが最後に起動またはリロードされてからの要求が含まれています。
Binding Update or Ack Counters (this connection)	
binding-updates-sent	フェールオーバー パートナーに送信されたバインド更新 (BNDUPD) メッセージの数。
binding-acks-received	フェールオーバー パートナーから受信したバインド確認 (BNDACK) メッセージの数。
binding-updates-received	フェールオーバー パートナーから受信したバインド更新 (BNDUPD) メッセージの数。
binding-acks-sent	フェールオーバー パートナーに送信されたバインド確認 (BNDACK) メッセージの数。
v6-binding-updates-sent	最後に確立された接続の開始以降にフェールオーバー パートナーから受信した IPv6 バインディング更新 (BNDUPD6) メッセージの数。
v6-binding-acks-received	最後に確立された接続の開始以降にフェールオーバー パートナーから受信した IPv6 バインディング確認 (BNDACK6) メッセージの数。
v6-binding-updates-received	最後に確立された接続が開始されてから、フェールオーバー パートナーから受信した IPv6 バインディング更新（BNDUPD6）メッセージの数。
v6-binding-acks-sent	最後に確立された接続の開始以降にフェールオーバー パートナーに送信された IPv6 バインディング確認 (BNDACK6) メッセージの数。
バインディング更新/ACK カウンタの合計	

DHCP 関連のフェールオーバー サーバー属性	説明
binding-updates-sent-total	最新の統計リセット以降にフェイルオーバー・パートナーに送信された IPv4 バインディング更新 (BNDUPD) メッセージの数。
binding-acks-received-total	最新の統計リセット以降にフェールオーバーパートナーから受信した IPv4 バインド確認 (BNDACK) メッセージの数。
binding-updates-received-total	最新の統計リセット以降にフェイルオーバー・パートナーから受信した IPv4 バインディング更新 (BNDUPD) メッセージの数。
binding-acks-sent-total	最新の統計リセット以降にフェールオーバーパートナーに送信された IPv4 バインド確認 (BNDACK) メッセージの数。
v6-binding-updates-sent-total	最新の統計リセット以降にフェイルオーバー・パートナーに送信された IPv6 バインディング更新 (BNDUPD6) メッセージの数。
v6-binding-acks-received-total	最新の統計リセット以降にフェールオーバーパートナーから受信した IPv6 バインディング確認 (BNDACK6) メッセージの数。
v6-binding-updates-received-total	最新の統計リセット以降にフェイルオーバー・パートナーから受信した IPv6 バインディング更新 (BNDUPD6) メッセージの数。
v6-binding-acks-sent-total	最新の統計リセット以降にフェールオーバーパートナーに送信された IPv6 バインディング確認 (BNDACK6) メッセージの数。
フロー制御カウンター (この接続)	
current-binding-updates-in-flight	現在、現在処理中 (送信) されているバインド更新の数 (IPv4 と IPv6 の両方)。

DHCP 関連のフェールオーバー サーバー属性	説明
current-binding-updates-queued	現在キューに入っているバインド更新の現在の数 (IPv4 と IPv6 の両方)。 通常、この数は（総リースのパーセンテージとして）小さいはずですが、更新を必要とする多数のリースがある場合（パートナーのいずれかが停止した後に統合された場合など）、またはパートナーが更新の処理に時間がかかる場合は、大きくなる可能性があります。この数がリースの合計数を超えることはありません。 この数が 1000 を超えるか、リースの 10%（いずれか大きい方）を超える場合、あるいはさらに増加し続けている場合には、懸念が生じます。通常、これはフェールオーバーパートナーで要求の処理に大きな遅延が発生していることを意味します（通常はディスクの遅延が主な問題です）。
maximum-binding-updates-in-flight	一度に処理中 (送信) されたバインディング更新 (IPv4 と IPv6 の両方) の最大数。
maximum-binding-updates-queued	一度にキューに入れられたバインディング更新の最大数 (IPv4 と IPv6 の両方)。
last-binding-update-sent-time	最後のバインディング更新 (IPv4 または IPv6) が送信された時刻。
last-binding-ack-received-time	最後の IPv4 または IPv6 バインディング確認応答 (NAKed かどうか) を受信した時刻。
last-binding-update-received-time	最後のバインディング更新 (IPv4 または IPv6) を受信した時刻。
last-binding-ack-sent-time	最後の IPv4 または IPv6 バインディング確認 (NAKed かどうか) が送信された時刻。

表 *6 : DNS* 関連のフェールオーバー サーバーの属性

DNS 関連サーバー属性	説明
General attributes	
current-time	このオブジェクトを返すサーバー上の現在の時刻。

DNS 関連サーバー属性	説明
ipaddr	IP アドレス
comm-state	なし、OK、中断の 3 つの値が考えられます。 DHCP とリモート サーバー間の通信の状態。'OK' は、DHCP サーバーがリモート サーバーとの通信に成功したことを示します。'中断' は、DHCP サーバーがリモート サーバーとの通信に失敗したことを示します。
dns-server-state	プローブまたは SEND-UPDATE の 2 つの値があります。 PROBE は、DHCP サーバーがこのサーバーと通信を試みていないか、ダウンしていると判断され、プローブがアクティブであることを示します (これは、DHCP サーバーが 1 つの更新要求を送信するだけであることを意味します)。 SEND-UPDATE は、サーバーが通信しているように見え、DHCP サーバーが多くの要求を送信できることを意味します。
probe-polling-event-id	ゼロ。
要求	リモート サーバーで現在未処理の要求の数。
HA DNS Configuration information	
ha-dns-role	この DNS サーバーが果たす役割です。値は、スタンドアロン DNS、HA-MAIN、または HA バックアップです。 DNS サーバーは、スタンドアロン DNS、または HA-DNS が使用されている場合は HA-Main または HA-Backup にすることができます。
dns-timeout	動的 DNS 更新を再試行する前に、動的 DNS 更新に対する DNS サーバーからの応答を DHCP サーバーが待機するミリ秒数です。
max-dns-retries	DHCP サーバーが DNS サーバーに動的更新を送信しようとする回数。

DNS 関連サーバー属性	説明
ha-dns-failover-timeout	DHCP サーバーがフェールオーバーを実行して次の DNS サーバーを使用して動的更新を実行するまで、DHCP サーバーが DNS サーバーからの応答を待機する最大時間 (秒) です。デフォルト値は 30 秒です。
ha-dns-probe-timeout	cnr-ha-dns が有効になっている場合、HA-DNS サーバーがコミュニケーション中断状態または同期中の場合、DHCP サーバーはこのタイマーを使用して HA-DNS サーバー間のオーバーオーバーの遅延を調整し、遅延を軽減します。デフォルト値は 3 秒です。
ha-dns-probe-retry	cnr-ha-dns が有効になっている場合、DHCP サーバーは、HA-DNS サーバーが COMMUNIATION-INTERRUPTED 状態または同期中の場合に、この再試行回数と ha-dns プローブタイムアウトを使用して、HA-DNS サーバー間でのオーバーオーバーの遅延を調整し、遅延を軽減します。再試行のデフォルト値は 1 です。

表 *7: TCP* リスナおよび接続関連サーバーの属性

TCP リスナーと接続関連サーバー属性	説明
TCP リスナ関連サーバー属性	
ipaddr	リスナーがバインドされているアドレス。これは 0.0.0.0 である可能性があります。
comm-state	通信の状態。これは常に何もありません。
ip6address	リスナーがバインドされている IPv6 アドレス。これは 0::0 である可能性があります。
name	サービスの名前。
port	リスナーがバインドされるポート番号。このポートへの着信接続が処理されます。
total-connections	受信接続の総数。
current-connections	現在アクティブな接続の数。

TCP リスナーと接続関連サーバー属性	説明
rejected-connections	アクティブな接続の最大数を超えたなど、拒否された着信接続の総数。
TCP 接続関連のサーバー属性	
ipaddr	接続のリモート エンドのアドレス。
comm-state	通信の状態。これは常にOKです。
ip6address	接続のリモート エンドの IPv6 アドレス。
name	この接続が受け入れられたサービスの名前。
port	接続のリモート エンドのポート番号。
total-requests	受信した要求メッセージの総数。
current-requests	アクティブな要求の数。
current-state	接続の現在の状態です。
total-replies	送信した応答メッセージの総数。
start-time	接続が確立した時刻。
last-receive-time	受信した最後のバイトの時刻。
last-send-time	最後に送信されたバイトの時刻。
total-bytes-received	この接続で受信した合計バイト数。
total-bytes-sent	接続を介して送信された合計バイト数。
our-ipaddr	接続のローカル エンドのアドレス。
our-ip6address	接続のローカル エンドの IPv6 アドレス。
our-port	接続のローカル エンドのポート番号。

その他のコントロールは、次のページで使用できます。

- [関連サーバー（Related Server）] タブのデータを更新するには Refresh Data をクリックします。
- パートナーが通信が中断したフェールオーバー状態の場合、[関連サーバー（Related Server）] タブで、パートナーダウン日の設定の入力フィールドに関連付けて Set Partner Down をクリックできます。この設定は、start-of-communications- interrupted の値に初期化されます（通常の Web UI モードでは、この日付を初期化された日付より前の値に設定することはできません。エキスパート Web UI モードでは、この値を任意の日付に設定できます。Set

Partner Down をクリックした後は、[DHCPサーバーの関連サーバーの一覧（List Related Servers for DHCP Server）] ページに戻り、パートナーダウンアクションの結果を表示します。両方のパートナーをパートナー ダウン モードに設定しないでください。

- [DHCPサーバの関連サーバの一覧（List Related Servers for DHCP Server）] ページまたは [フェールオーバー関連サーバーの表示（View Failover Related Server）] ページから戻るには、Return をクリックします。

CLI コマンド

DHCPサーバーの関連サーバーを、値のサブセットと共に簡単なテーブル形式で一覧表示するには、dhcp getRelatedServers を使用します。完全な詳細を報告するには（テーブルではなく通常のオブジェクトフォーム表示で）dhcp getRelatedServers full を使用します。

バーチャル プライベート ネットワークの設定

このセクションでは、仮想プライベート ネットワーク(VPN)をサポートするようにCisco Prime Network レジストラー DHCP サーバーを設定する方法について説明します。

VPN の設定には、通常の DHCP ホスト IP アドレス指定に調整を加えることが関係します。VPN で使用するプライベート アドレス空間は、インターネット全体から見て一意ではない場合があります。このため、Cisco Primeネットワーク レジストラーは、VPN 識別子によって識別されるIPアドレスをサポートします。ルーター上のリレーエージェントもこの機能をサポートする必要があります。VPN 識別子は、クライアントが属する VPN を選択します。DHCP 用 VPN は現在 Cisco IOS ソフトウェアでのみサポートされており、最新バージョンでは、リレーされた DHCP メッセージに VPN ID を含めることができます。

関連項目

DHCP を使用した仮想プライベート ネットワークの設定 （51 ページ）

VPN とサブネット割り当ての調整パラメータ （59 ページ）

DHCP を使用した仮想プライベート ネットワークの設定

作成する VPN は、次の場合にフィルタリング メカニズムを提供します。

- 統合アドレス空間の表示（アドレス空間の表示 （125 ページ）を参照）
- 住所ブロックの一覧表示（アドレス ブロックの追加 （117 ページ）を参照）
- サブネットのリスト（アドレス ブロックとサブネット （115 ページ）を参照）
- DHCP 使用率の照会（使用率履歴データの照会 （132 ページ）を参照）
- リース履歴の照会（IP リース履歴の実行 （274 ページ）を参照）

VPN を設定しない場合、Cisco Prime Network レジストラーは、各スコープでグローバル VPN 0 を使用します。

クライアントがリレー エージェントを使用して DHCP サーバーに IP アドレスを要求できるように VPN を構成するには、VPN を定義し、スコープを関連付ける必要があります。具体的には次のとおりです。

1. DHCP VPN トラフィックを処理するリレー エージェントが、DHCP のリレー エージェント情報オプション(82)のvpn-idサブオプションをサポートするバージョンの Cisco IOS ソフトウェアで設定されていることを確認します。
2. VPN が VPN ID または VPN ルーティングおよび転送インスタンス(VRF)名によって識別されることを、Cisco IOS リレー エージェント管理者と調整します。
3. VPN のスコープを作成します。

関連項目

標準 仮想プライベート ネットワーク （52 ページ）

仮想プライベート ネットワークの作成と編集 （53 ページ）

VPN の使用状況 （55 ページ）

標準 仮想プライベート ネットワーク

図 4 : バーチャル プライベート ネットワーク DHCP 構成は、VPN ブルーの一部として DHCP クライアント 1 を使用し、VPN クライアント 2 を VPN レッドで示す一般的な VPN シナリオを示しています。たとえば、VPN ブルーの DHCP クライアント 1 と VPN 赤のクライアント 2 の両方に同じプライベート ネットワーク アドレスがあります: 192.168.1.0/24。DHCP リレー エージェントには、2 つの VPN に含まれるゲートウェイ アドレスとグローバル アドレス (172.27.180.232)があります。2 つのフェールオーバー DHCP サーバーがあり、どちらも外部ゲートウェイ アドレスを介してリレー エージェントを認識しています。

サーバーがクライアントに VPN サポート アドレスを発行するために行われる処理は次のとおりです。

1. DHCP クライアント 1 は、その MAC アドレス、ホスト名、および要求された DHCP オプションを含む DHCPDISCOVER パケットをブロードキャストします。
2. アドレス 192.168.1.1 の DHCP リレー エージェントはブロードキャスト パケットをピックアップします。パケットにRelay エージェント情報オプション (82) を追加し、サブネットとして 192.168.1.0 を識別するサブネット選択サブオプションが含まれています。このパケットには、VPN を青で識別するvpn-idサブオプションも含まれています。DHCP サーバーは要求元のクライアントと直接通信できないため、server-id-overrideサブオプションには、クライアントによって認識されるリレー エージェントのアドレス (192.168.1.1) が含まれています。リレー エージェントはパケットの外部ゲートウェイ アドレス (giaddr) にも含まれます。
3. リレーエージェントは、DHCPDISCOVER パケットをサブネット上の構成済み DHCP サーバーにユニキャストします。
4. DHCP サーバー 1 はパケットを受信し、vpn-idおよびサブネット選択のサブオプションを使用して、適切な VPN アドレス空間から IP アドレスを割り当てます。サブネットと VPN で使用可能なアドレス 192.168.1.37 を検出し、パケットのyiaddrフィールド (クライアントに提供されるアドレス) に配置します。

5. サーバーは、GIADDR値で識別されるリレーエージェントにDHCPOFFERパケットをユニキャストします。
6. リレーエージェントは、リレーエージェント情報オプションを削除し、DHCPクライアント1にパケットを送信します。
7. DHCPクライアント1は、DHCPREQUESTメッセージをブロードキャストして、それが提供されたIPアドレスと同じIPアドレスを要求します。リレー エージェントは、このブロードキャスト メッセージを受信します。
8. リレー エージェントはDHCPREQUESTパケットをDHCPサーバー1に転送し、ユニキャストDHCPACKパケットをクライアントに返します。
9. リース更新の場合、クライアントはDHCPACKメッセージのDHCPサーバー識別子オプションで見つかったIPアドレスにDHCPRENEWパケットをユニキャストします。これは、リレーエージェントのアドレスである192.168.1.1です。DHCPリレー エージェントはパケットをDHCPサーバーにユニキャストします。サーバーは、最初に元のアドレスを提供したサーバーが必ずしも知らなくても、通常の更新処理を行います。サーバーはユニキャストDHCPACKパケットで応答します。リレー エージェントは、次に、ciaddrフィールド値で識別されるクライアントIPアドレスにDHCPACKパケットを転送します。

   リレーエージェント情報オプション(82)のサーバーID オーバーライドサブオプションが存在する場合、DHCPサーバーはその値を使用して応答パケットのdhcp-server-identifierオプションの値と比較します。DHCPクライアントユニキャストが行うパケットは、サーバーではなくリレー エージェントに直接送信されます(実際にはクライアントからはアクセスできない可能性があります)。パケットにserver-id-overrideサブオプションが含まれている場合、フェイルオーバー環境の両方のパートナーはリースを更新できます。

仮想プライベート ネットワークの作成と編集

VPNとそのインデックスを設定するには、次の手順を実行します。

ステップ1 リレー エージェントのVPN IDまたはVRF名によってVPNが設定されていることをCisco IOSリレー エージェント管理者と調整します。これはCiscoプライムネットワーク レジストラーでVPNを識別する方法を決定します。

ステップ2 IOSスイッチまたはルータで設定されているVPNにDHCPクライアントをプロビジョニングできるように、VPNを作成します。

ステップ3 VPNインデックスを入力**します。**関連付けられたIDも一意である必要があります。

インデックスを追加するには、次の手順に従います。

- Local cluster（Advanced）：[設計（Design）]メニューの[DHCP設定（DHCP Setting）]サブメニューの[VPN]を選択して、[VPNの一覧/追加（List/Add VPNs）]ページを開きます。VPNに、クラスター内の数値キー識別子と一意の名前を指定します。

- Regional cluster：VPNを含むローカルクラスタを追加します（[操作（Operate）]メニューの[サーバー（Servers）]サブメニューの下の[クラスタの管理（Manage Clusters）]を選択します）。次に、[設計（Design）]メニューの[VPN]を選択します。[VPNの一覧/追加（List/Add VPNs）]ページが開きます。このページでVPNを作成するか、ローカル クラスタからVPNをプルできます。

- VPN を作成する場合は、数値キー識別子と一意の名前を指定します。
- ローカル クラスタから VPN をプルする場合は、[VPNの一覧/追加（List/Add VPNs）] ページの [VPN] ウィンドウで [データのプル（Pull Data）] アイコンをクリックし、選択したクラスタから特定の VPN またはすべての VPN をプルします。

[VPNの一覧/追加（List/Add VPNs）] ページの [プッシュ（Push）] または [すべてプッシュ（Push All）] アイコンをクリックして、VPN をクラスターにプッシュすることもできます。次に、[VPNデータをローカルクラスターにプッシュ（Push VPN Data to Local Clusters）] ページで、VPN をプッシュする同期モードとクラスターを選択します。

- CLI で、vpn name create key を使用します。次に例を示します。

```
nrcmd> vpn blue create 99
```

ステップ 4 VPN ID または VRF 名で適切な VPN 識別子を指定します。一方のみでかまいません。

- VPN ID を使用する場合は、VPN の vpn-id 属性値を設定します。値は IETF RFC 2685 に従って、通常は 16 進数で、 oui:index の形式です。この 3 オクテット VPN 組織固有識別子 (OUI) で構成され、その後に、VPN の所有者または ISP に対応し、その後にコロンが続きます。その後、VPN 自体の 4 オクテット インデックス番号が続きます。VPN ID の値を [VPNのリスト/追加（List/Add VPNs）] ページに追加します。CLI で、vpn-id 属性を設定します。次に例を示します。

```
nrcmd> vpn blue set vpn-id=a1:3f6c
```

- VPN ルーティングおよび転送(VRF)インスタンス名を使用する場合は、VPN のVRF 名属性値を設定します。シスコのルータは、VRF 名を頻繁に使用します。[VPNのリスト/追加（List/Add VPNs）] ページに VRF 名の値を追加します。CLI で、vrf 名属性を設定します。次に例を示します。

```
nrcmd> vpn blue set vrf-name=framus
```

ステップ 5 VPN の説明を追加します (オプション)。

ステップ 6 Add VPN をクリックします。VPN を編集して、[VPN の編集] ページの値を変更できます。

ステップ 7 VPN のスコープを作成します。

識別のために、VPN 名とスコープ名をできるだけ類似する必要があります。

1. Web UI の [デザイン（Design）] メニューから DHCPv4 サブメニューの下の [スコープ（Scopes）] を選択し、[DHCPスコープの一覧/ 追加（List/Add DHCP Scopes）] ページを開きます。
2. Web UI の上部にある [設定（Settings）] ドロップダウン リストの下にある [VPN] サブメニューから VPN を選択します。スコープの作成時に VPN を設定した後は、VPN を変更することはできません。

   CLI で、次の 3 つの方法のいずれかで、スコープがどの VPN に属しているかを特定します。

   - VPN 名は vpn属性(VPN ID をスコープに適用) で使用します。
   - vpn-id属性を介した VPN ID 自体。
   - コマンド ラインで VPN またはその ID を省略した現在のセッション VPN 名。

現在のセッションのデフォルト VPN を設定するには、セッション セット current-vpnを使用します。その後、スコープの通常のアドレス範囲と必要なオプションのプロパティを設定できます。次に例を示します。

```
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 vpn=blue
```

または

```
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 vpn-id=99
```

または

```
nrcmd> session set current-vpn=blue
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0
```

実行されるアクション（Then）

```
nrcmd> scope blue-1921681 addRange 192.168.1.101 192.168.1.200
nrcmd> scope-policy blue-1921681 setOption routers 192.168.1.1
```

ステージング DHCP 編集モードの場合は、すべての VPN とスコープを作成した後で DHCP サーバーをリロードします。

VPN の使用状況

VPN 名は、IP アドレス(リース)、スコープ、サブネットなど、Cisco Prime Network レジストラーの多くの DHCP オブジェクトを修飾するために使用されます。たとえば、リース名には次の構文を使用できます。

vpn/ipaddress

たとえば、red/192.168.40.0

VPN には、予約語globalとallを除く任意の一意のテキスト文字列を使用できます。データをリースglobalするall場合に使用できます。VPNglobalは [なし] VPN にマップされます。VPNallは、特定の VPN と [なし] VPN の両方にマップされます。

CLI では、オブジェクトの定義時に VPN またはその ID を省略すると、VPN は デフォルトで session set current-vpnによって設定された値になります。Web UI では、現在の VPN が定義されていない場合、デフォルトで [none] VPN が使用され、定義済み VPN の外部のすべてのアドレスが含まれます。

これらのオブジェクトには、関連する VPN プロパティがあります。

- Address blocks：アドレス ブロックの VPN を定義します。Design > DHCPv4 メニューから Address Blocks を選択して、[DHCPアドレスブロックの一覧/追加（List/Add DHCP Address Blocks）] ページを開きます（詳細モードで使用できます）。Web UI の上部にある [設定（Settings）] ドロップダウン リストの下にある [VPN] サブメニューから VPN を選択します。CLI で、作成およびdhcp-address-block属性設定コマンドを使用します。次に例を示します。

  ```
  nrcmd> dhcp-address-block red create 192.168.50.0/24

  nrcmd> dhcp-address-block red set vpn=blue

  nrcmd> dhcp-address-block red set vpn-id=99
  ```

（注） オブジェクトを作成する前に、vpn-id 値を dhcp アドレス ブロックを作成する必要がある VPN に設定します。vpn-id が常に現在の VPN であると仮定しないでください。

- Clientsand :外部ではなく、Cisco Prime Network レジストラー IP Express 内で VPN をプロビジョニングするのが最善の場合client-classesがあります。 この機能をサポートするために、クライアントまたはクライアント クラスの VPN を指定できます。次の 2 つの属性が提供されます。
  - default-vpn —着信パケットにvpn-idまたはvrf-name値がまだない場合にパケットが取得する VPN。属性は、クライアントおよびクライアント クラスで使用できます。
  - override-vpn —着信パケットのvpn-idまたはvrf-name値に何が提供されても、パケットは何を取得します。この属性は、クライアントとクライアントクラスで使用できます。クライアント クラスで優先 VPN を指定し、クライアントの既定の VPN を指定した場合、クライアント クラスのオーバーライド VPN がクライアントの既定の VPN よりも優先されることに注意してください。

  ローカル クラスタでClients、Client ClassesDesign>DHCPSettingsメニューから選択するか(詳細モードで利用可能)。クライアント クラスまたはクライアントを作成または編集し、default-vpn属性値とオーバーライド VPN属性値を入力します。

  地域クラスターで -Client ClassesDesign>DHCPSettingsメニューから (詳細モードで使用可能) を選択します。クライアントクラスを作成またはプルしてから編集し、default-vpn属性値とオーバーライド VPN属性値を入力します。

  CLI で、作成およびclient-class属性設定コマンドを使用します。次に例を示します。

  ```
  nrcmd> client 1,6,00:d0:ba:d3:bd:3b set default-vpn=blue

  nrcmd> client-class CableModem set override-vpn=blue
  ```

  たとえば、ケーブル モデムの導入では、override-vpn属性を使用してケーブル モデムをプロビジョニングできます。クライアント クラスはケーブル モデムのスコープを決定し、スコープは uBR の VPN を決定します。ケーブル モデムを介したユーザー トラフィックは、vpn-idサブオプションを設定して、特定の VPN を使用します。オーバーライド VPN 値は、クライアントに設定されたデフォルト VPNもオーバーライドします。

- Leases：リースのリスト、リースの表示、またはリース属性の取得。

  CLI で、リースをインポートするにはimport leases 、filenameを使用します。ファイル内の各リース エントリには、行の末尾に VPN を含めることができます。この機能が見つからない場合、Ciscoプライムネットワーク レジストラーは[none]VPNを割り当てます。（リース データのインポートとエクスポート （235 ページ）も参照してください）。

  ```
  nrcmd> import leases leaseimport.txt
  ```

  VPN を含むようにアドレスまたはリースデータをエクスポートするには –vpn オプションの export leases を使用します。VPN 値の予約語 global または all

  - Global：定義された VPN（[none] VPN）の外部にあるアドレス。

- All：[なし] VPN を含むすべての VPN。

VPN を省略すると、エクスポートは によって設定された現在のsession set current-vpnVPN を使用します。現在の VPN が設定されていない場合、サーバーは [none] VPN を使用します。

```
nrcmd> export addresses file=addrexport.txt vpn=red

nrcmd> export leases -server -vpn red leaseexport.txt
```

- Scopes：DHCP を使用した仮想プライベート ネットワークの設定 （51 ページ） で説明しているとおり、範囲には VPN 名とその ID を含めることができます。

> 注 スコープの作成時に VPN を設定した後は、VPN を変更することはできません。

- Subnets：サブネットの一覧表示、サブネットの表示、またはサブネットの vpn 属性または vpn-id 属性の取得は VPN を示します。DHCP サブネットの割り当ての設定 （58 ページ） を参照してください。
- DHCP server：vpn-communication 属性が有効な場合（デフォルト）、DHCP サーバーは、DHCP リレーエージェント機能を強化して、DHCP サーバーと異なる VPN 上にある DHCP クライアントと通信できます。この機能は、リレー・エージェント情報オプション（82）のserver-id-override サブオプションによって示されます。

> 注 DHCP サーバーは、VPN に存在するクライアントに対して ping を実行しません。

サブネットの割り当ての設定

このセクションでは、オンデマンド アドレス プールのサブネット割り当てをサポートするように Cisco Prime Network レジストラー DHCP サーバーを設定する方法について説明します。

サブネット割り当てとは、クライアント (通常はルーターまたはエッジ デバイス) にサブネットをリースし、DHCP サービスを提供できるようにする方法です。この方法は、個々のクライアントアドレスの管理とともに使用したり、その代わりに使用できます。サブネット割り当てを使うと、DHCP インフラストラクチャによるサブネットのダイナミックな管理によって、IP アドレスのプロビジョニング、集約、特性評価、配布を大幅に改善できます。DHCP を介したサブネット割り当ては現在、Cisco IOS ソフトウェアでのみサポートされており、最新バージョンにはオンデマンド アドレス プール機能が組み込まれています。

（注） DHCP フェールオーバーには、DHCPv4 サブネット割り当ては含まれません。

関連項目

DHCP サブネットの割り当ての設定 （58 ページ）
VPN とサブネット割り当ての調整パラメータ （59 ページ）

DHCP サブネットの割り当ての設定

次のセクションでは、DHCP サーバーを使用してサブネット割り当てを設定する例を示します。図 10 : DHCP サブネット割り当ての構成例は、プロビジョニング デバイスに割り当てられたサブネットを使用したサブネット割り当ての構成例と、従来の DHCP クライアント/サーバー構成を示しています。

サブネットを割り当てる前に、DHCP サーバーはまずクライアントが接続している VPN を次の順序で決定します。

1. サーバーは、着信 VPN オプションを検索し、VPN の値を使用します。
2. VPN オプションが見つからない場合、サーバーは Relay Agent サブオプション値を使用し、VPN をサブネット アドレスと結合して一意の識別子を形成します。
3. リレー エージェント サブオプションが見つからない場合、サーバーはクライアント・クラス情報 (選択タグ) を探します。

DHCP サブネット割り当てを構成するには、次の手順を実行します。

ステップ 1 サブネットの DHCP アドレス ブロックを作成し、初期サブネット マスクとその増分を設定し、他のサブネット割り当て要求属性を設定します。また、ポリシーを関連付けるか、組み込みポリシーを定義します。

- VPN を使用する場合は、vpn 属性または vpn-id 属性を指定できます（DHCP を使用した仮想プライベート ネットワークの設定 （51 ページ）を参照）。
- サーバーは、要求パケットにサブネット アロック DHCP オプション (220) の存在を使用して、パケットがサブネット割り当て要求であることを判断します。サーバーまたは VPN に addr ブロック使用選択タグ属性を設定する場合は、サブネット名サブオプション (3) を選択タグとして使用するようにサーバーを構成できます。
- オプションで、DHCP サーバーまたは VPN オブジェクトの addr ブロック -default-selection-tags 属性を設定して、デフォルトの選択タグを設定できます。これは、アドレスを割り当てる 1 つ以上のサブネットを識別します。リレー エージェントがサブネットに関連付けられた VPN 文字列 (VPN オプションまたはリレー エージェント サブオプションを使用して) を送信する場合、その文字列を addr ブロック -default-選択タグ値の 1 つとして持つアドレス ブロックは、そのサブネットを使用します。
- サーバーと VPN の場合のデフォルトの動作では、DHCP サーバーは、クライアントが既に使用しているアドレス ブロックを使用して、クライアントにサブネットを割り当てようとします。addr ブロック

使用クライアント アフィニティ属性を無効にすると、サーバーはクライアントメッセージ内の他の選択データに基づいて、適切なアドレス ブロックからサブネットを提供します。

- 1 つの LAN セグメント上で複数のアドレス ブロックの構成をサポートする場合 (プライマリ スコープとセカンダリ スコープの使用に似ています)、セグメント名属性文字列値を DHCP アドレス ブロックに追加します。リレー エージェントは、単一のサブネット選択アドレスを送信するときに、そのセグメント名文字列値でタグ付けされたアドレス ブロックを選択します。ただし、LAN セグメント機能 (addr ブロック - lan-segments)をサーバーレベルまたは VPN レベルで明示的に有効にする必要もあります。
- ポリシーを関連付ける代わりに、アドレス ブロック埋め込みポリシーのプロパティを設定できます。クライアント、クライアント クラス、スコープの組み込みポリシーと同様に、アドレス ブロック ポリシーの属性を有効にしたり、無効にしたり、設定したり、設定を解除したり、取得したり、表示したりできます。また、DHCP オプションの設定、設定解除、取得、および一覧表示、およびベンダー オプションの設定、未設定、および一覧表示を行うこともできます。アドレス ブロックの埋め込みポリシーを削除すると、埋め込みポリシーのすべてのプロパティが解除されることに注意してください。

ステップ 2 サーバーは、リレー エージェント要求に基づいてサブネットを割り当てることに注意してください。要求されていない場合、既定のサブネット サイズは 28 ビットのアドレス マスクです。DHCP アドレス ブロックの既定サブネット サイズ属性を設定することで、必要に応じてこの既定値を変更できます。

次に例を示します。

```
nrcmd> dhcp-address-block red set default-subnet-size=25
```

ステップ 3 DHCP サーバーが作成するサブネットは、アドレス ブロックから制御できます。vpn-name/netipaddress/mask の形式でサブネットを識別し、vpn-nameはオプションです。サブネット制御には、リースと同様にサブネットのアクティブ化と非アクティブ化が含まれます。同様に、サブネットを強制的に使用できるようにし、その前に、サブネットが割り当てられたクライアントがサブネットを使用しなくなったことを確認する条件を満たすことができます。まず、作成されたサブネットを表示します。

ステップ 4 DHCP サーバーをリロードします。

VPN とサブネット割り当ての調整パラメータ

VPN およびオンデマンド のアドレス プールに対して、これらの調整パラメータを検討してください。

- Keep orphaned leases that have nonexistent VPNs：Cisco Prime Network レジストラー は、通常、関連付けられた VPN を持たないリースを Cisco Prime Network レジストラー 状態データベースに保持します。この変更は、 DHCP 属性の delete-orphaned-leases を有効にすることで変更できます。サーバーは、クライアントをリースに関連付けるリース状態データベースを保持します。スコープの変更によって既存のリースが無効になった場合、リース データベースには孤立したリース エントリが含まれます。サーバーは、このデータを使用してクライアントをリースに再関連付けしようとするので、通常はリースの期限が切れても削除されません。この欠点の 1 つは、リース データベースがディスク領域を過剰に消費する可能性があります。delete-orphaned-leases 属性を有効にすると、このようなリース デー

タベースエントリは、次回のサーバーの再ロード時に削除されます。ただし、この属性を有効にする場合は、リースを無効にレンダリングすると、サーバーが空きであると考えられるリースを使用するクライアントが発生する可能性があるため、注意が必要です。これにより、ネットワークの安定性が損なわれます。

- Keep orphaned subnets that have nonexistent VPNs or address blocks：これはデフォルトの動作ですが、DHCP 属性 DHCP を有効にして delete-orphaned-subnets を有効にすることで変更できます。DHCP サーバーは起動すると、サブネットのデータベースを読み取り、各サブネットの親 VPN とアドレスブロックの検索を試みます。この属性が有効な場合、サブネットがサーバーで構成されなくなった VPN を参照している場合、またはサーバーがサブネットを含む親アドレスブロックを見つけることができない場合、サーバーは状態データベースからサブネットを完全に削除します。
- Keep the VPN communication open：これはデフォルトの動作ですが、DHCP 属性 vpn-communication を無効にすることで変更できます。サーバーは、拡張 DHCP リレーエージェント機能を使用して、サーバーとは異なる VPN 上に存在するクライアントと通信できます。これは、リレーエージェント情報オプション(82)のvpn-idサブオプションの出現によって通知されます。サーバーがサーバーとは異なる VPN 上のクライアントと通信する必要がない場合は、vpn 通信属性を無効にできます。通常、その動機は、不正な DHCP クライアント アクセスを防止することで、ネットワーク セキュリティを強化することです。

BOOTP の設定

BOOTP (BOOTstrap プロトコル) は、ディスクレスコンピュータをロードするために作成されました。その後、ホストがインターネットを使用するために必要なすべての TCP/IP 情報を取得できるようにするために使用されました。BOOTP を使用することにより、ホストは、ネットワーク上で要求をブロードキャストし、BOOTP サーバーから必要な情報を取得できます。BOOTP サーバーは、着信 BOOTP 要求をリッスンし、そのネットワーク上の BOOTP クライアントの構成データベースから応答を生成するコンピューターです。BOOTP は、DHCP とは異なり、リースまたはリースの有効期限の概念が存在しません。BOOTP サーバーが割り当てるすべての IP アドレスは永続的です。

Cisco プライムネットワーク レジストラーを BOOTP サーバーのように動作するように設定できます。また、BOOTP では通常静的アドレス割り当てが必要ですが、IP アドレスを予約するか(したがって静的割り当てを使用する)、または BOOTP クライアントに IP アドレスを動的に割り当てるかを選択できます。

関連項目

BOOTP について （61 ページ）
スコープの BOOTP の有効化 （62 ページ）
BOOTP クライアントの移動または廃止 （62 ページ）

BOOTP について

BOOTP パケットを返すように DHCP サーバーを構成する場合、オプション領域以外のフィールドで、BOOTP はDHCP パケットの情報を必要とすることに注意してください。BOOTP デバイスは、DHCP パケットのブート ファイル(file)、 サーバー IP アドレス (siaddr)、およびDHCP パケットのサーバー ホスト名 (sname) フィールドに情報を必要とします (RFC 2131 を参照)。

すべての Cisco Prime Network レジストラー DHCP ポリシーには、ファイル、siaddr、または snameフィールドに直接返す情報を設定できる属性があります。Cisco Prime Network レジストラー DHCP サーバーは、ポリシー オプションを設定し、BOOTP デバイスに返すファイル、sname、またはsiaddrの値を決定する設定パラメータもサポートしています。

Cisco Prime Network レジストラーは、オプションと、DHCP クライアントに返すオプション、sname、またはsiaddrの値を設定できる、類似の設定パラメータをサポートしています。これは、DHCP 要求のdhcp パラメータ要求オプションで DHCP クライアントによって要求されるオプションに追加されます。したがって、BOOTP と DHCP の両方の応答パケットをデバイスに適切に設定できます。

ステップ 1 BOOTP 属性に使用する値を決定します。

- file：起動ファイルの名前
- siaddr：サーバー IP アドレス
- sname：任意のサーバーホスト名

ステップ 2 BOOTP クライアントに返すオプションとその値のリストを決定します。

ステップ 3 BOOTP 要求に関連付けるポリシーに、次の値を設定します。

- BOOTP クライアントに送信する属性（packet-siaddr、packet-file-name、packet-server-name）。
- BOOTP クライアントに戻すサーバー アドレスやドメイン名などのオプション値。
- BOOTP クライアントに返すフィールドとオプションのリスト。

ステップ 4 関連するスコープを BOOTP 処理用に使用可能にします。

ステップ 5 このスコープで、要求するBOOTP クライアントのアドレスを指定する場合は、動的BOOTP 処理を有効にします。動的BOOTP を有効にしていない場合は、このスコープでアドレスを指定する各BOOTP クライアントに予約を行う必要があります。

スコープの BOOTP の有効化

スコープに対して BOOTP 処理を有効にすることができます。ローカル クラスタ Web UI で作成されたポリシーに対して特定の属性と BOOTP 応答オプションを設定するか、または名前 policycreate[属性=値] とpolicy名前set属性=値[属性=値..] を使用して BOOTP を設定します。ポリシー属性とオプションをカンマ区切りリストとして設定します。属性は、クライアントブートプロセスで使用するエンティティです。

- packet-siaddr：次のサーバーの IP アドレス
- packet-file-name：ブートファイルの名前
- packet-server-name：サーバーのホスト名

サーバーは、これらの属性値の最初のインスタンスをポリシー階層に調べています。

CLI policy 名 setOption<opt-name|id>value[-blob] [-roundrobin] には、値の前に空白 (等号ではない) が必要です。-roundrobinが有効な場合、DHCP サーバーは、異なる回転順序で複数の値を含むオプションデータを返すように指示します。特定のクライアントは常に同じ順序を取得しますが、異なるクライアントは、クライアント識別子に基づいてオプションに対して構成された複数の値の順序の異なる「ローテーション」を取得します。

また、必要に応じて BOOTP および動的ブート・ブート・ブート・プログラムを使用可能にし、DHCP サーバーが BOOTP 要求を使用して DNS サーバーを更新することを確認します。次のオプションがあります。

- オプション dhcp-lease-time を設定します。
- dynamic-bootp 属性を有効にします。
- update-dns-for-bootp 属性を有効にします。
- update-dns-for-bootp 属性を有効にします。

BOOTP クライアントの移動または廃止

BOOTP クライアントを移動または使用停止にした場合、そのリースを再利用できます。BOOTP クライアントを使用停止にするには、そのリース予約をスコープから削除し、リースを強制的に使用可能にする必要があります。

ローカル クラスター Web UI でリースを使用するように強制するかscope、nameremoveReservation を使用します (ipaddr |マカドル|検索キー)[-mac |-**ブロブ**|-**文字列**]およびlease[VPN 名/]ipaddrforce-availableを CLI で実行します。

動的 BOOTP の使用

動的 BOOTP を使用する場合、その他の制限が適用されます、スコープ内のアドレスの使用には、BOOTP クライアントが永続的に割り当てられ、無期限のリースを受信します。

DHCPフェールオーバーを使用している場合、スコープの動的ブートオプションが有効になっていないサーバーが PARTNER-DOWN 状態になると、メイン サーバーとバックアップ サーバーのどちらで最初に使用できるかに関係なく、そのスコープから使用可能な IP アドレスを割り当てることができます。ただし、dynamic-bootpオプションが有効な場合、メインサーバーとバックアップサーバーは、独自のアドレスのみを割り当てることができます。したがって、dynamic-bootpオプションを有効にするスコープでは、フェールオーバーをサポートするためにより多くのアドレスが必要になります。

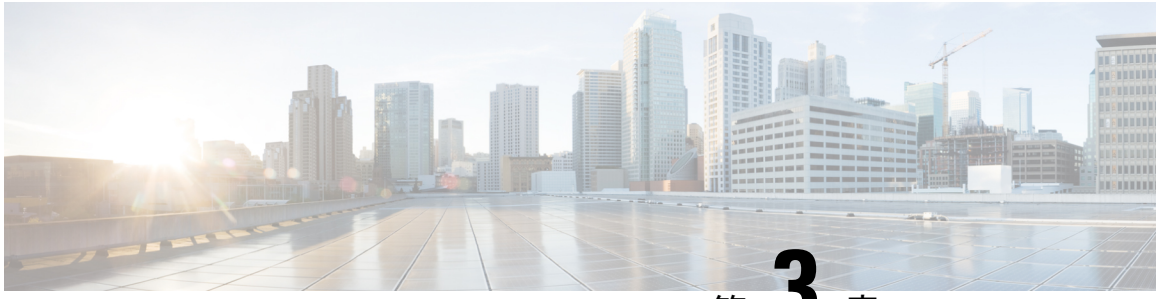
動的ブートを使用する場合:

1. 動的 BOOTP クライアントを単一のスコープに分離します。DHCP クライアントがそのスコープを使用できないようにします。ローカル クラスター Web UI で、スコープの BOOTP 属性の下で、dhcp属性を無効にします。CLI で、scope name disable dhcp を使用します。
2. DHCP フェールオーバーを使用している場合は、DHCP サーバーのフェールオーバー動的bootp-backup-percentage属性を設定して、このスコープのバックアップ サーバに対して、より大きな割合のアドレスを割り当てます。この割合は、通常のバックアップの割合よりも 50% も高くなる可能性があります。

BOOTP リレー

BOOTP リレーをサポートするルーターは、通常、DHCP サーバーを指すアドレスを持ちます。たとえば、Cisco ルータを使用している場合、特定のマシンのアドレスを含む IPヘルパー アドレスという用語が使用されます。この場合、このアドレスを使用して、すべての BOOTP (および DHCP) ブロードキャスト パケットを転送します。このアドレスは、ホストに最も近いルーターで構成してください。

ヒント DHCP クライアントが DHCP サーバーからアドレスを受信していない場合は、ネットワーク設定、特にルータまたはリレー エージェントの設定をチェックして、ネットワーク デバイスが Cisco Prime Network レジストラー DHCP サーバー アドレスを指す設定になっていることを確認します。



第 3 章

DHCP フェールオーバーの管理

Cisco Prime Network レジストラーフェールオーバープロトコルは、何らかの理由でメインサーバーがオフラインになった場合に、バックアップDHCPサーバーが引き継がれるように設計されています。8.2より前のバージョンでは、このプロトコルはUDPベースで、IPv4経由でのみ動作し、DHCPv4のみをサポートしていました。8.2以降、このプロトコルはTCPベースで、IPv4またはIPv6のいずれかを使用するように構成でき、単一の接続でDHCPv4とDHCPv6の両方をサポートします。DHCPサーバーは、両方を使用するように構成されている場合、IPv4とIPv6の両方のトランスポートを試行し、最初に起動した接続を使用します。DHCPフェールオーバーでは、次の機能がサポートされています。

- DHCPv4 アドレス
- DHCPv6 アドレス (非一時および一時)
- DHCPv6 プレフィックス委任

DHCP フェールオーバーは、DHCPv4 サブネット割り当て (オンデマンド アドレス プール) には適用されません。

DHCP フェールオーバーの仕組み

DHCPフェールオーバーは、サーバーとパートナーの関係に基づいています。パートナーは、サーバーと同じ DHCPv4 スコープ、DHCPv6 プレフィックス、DHCPv6 リンク、予約、ポリシー、およびクライアント クラスを持つ必要があります。サーバーが起動すると、サーバーは互いに連絡を取ります。メインサーバーは、パートナーに DHCPv4 アドレスと DHCPv6 委任接頭部を提供し、そのパートナーをクライアント操作ごとに更新します。メインサーバーに障害が発生した場合、パートナーは、DHCPv4アドレスとDHCPv6委任プレフィックスを使用して、リースの提供と更新を引き継ぎます。メイン・サーバーが再び稼働可能になると、管理者の介入なしにパートナーと再統合されます。これらのサーバーは、フェールオーバー ペアと呼ばれる関係にあります。

次の場合、フェールオーバー プロトコルは DHCP を動作可能にします。

- The main —server**メイン**サーバーがダウンしている間に、パートナーがfailsサービスを引き継ぎます。パートナーを更新する前にメイン サーバーで障害が発生した場合でも、サーバーは重複するアドレスを生成できません。
- Communication —パートナーは、相手サーバーか、またはパートナーとの通信で障害が発生したのかを判断できない場合でも、正しく動作failsできます。サーバーは、両方とも実行されていて、それぞれがクライアントのサブセットとしか通信できない場合でも、重複するアドレスを発行することはできません。

フェールオーバー ペアを構成した後:

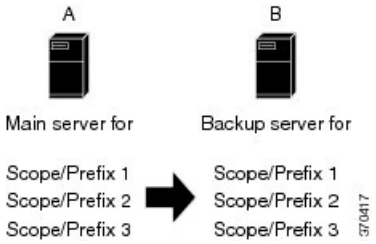
1. パートナーは接続します。
2. メイン サーバーは、既存のすべてのリースに関するデータをパートナーに提供します。
3. バックアップ サーバーは、メイン サーバーからバックアップ アドレスのプールを要求します。
4. メインサーバーは、各スコープまたはプレフィックスからパートナーに使用可能なアドレスの割合で応答します。
5. バックアップサーバーは、メインサーバーがダウンしているか、またはフェールオーバーペアの負荷分散が有効になっていると感じなければ、すべてのDHCPDISCOVER要求と送信要求要求を無視します。通常の操作では、バックアップサーバーは一部の更新要求と再バインド要求のみを処理します。
6. メイン サーバーは、すべてのクライアント操作の結果でパートナーを更新します。

フェールオーバー ペアのサーバーの構成は自動的に同期化できます。2つのサーバーは、使用可能なリースの再調整を動的に行います。メインサーバーが利用可能なリースの大部分を引き渡す場合、パートナーからリースを回収できます。

DHCP シンプル フェールオーバー

Cisco プライムネットワーク レジストラーは、単純なフェールオーバー設定のみをサポートします。単純なフェールオーバーには、1 つのメイン サーバーと 1 つのバックアップ サーバーのペアが含まれます (下の図を参照)。この例では、メイン サーバー A には 3 つのスコープまたはプレフィックスがあり、バックアップ サーバー B で同じように構成する必要があります。

図 *9:* 単純なフェールオーバーの例



DHCPv6 フェールオーバー

DHCPv6 フェイルオーバーは、DHCPv4 の単純なフェイルオーバー構成と非常によく似ています。DHCPv6 フェールオーバーパートナーは、ステートフルアドレスと委任されたプレフィックスリースで相互に更新を行い、通信が復元されたときに同期を実行し、一般的に DHCPv4 フェールオーバープロトコルの要件に従い、遵守します（ただし、DHCPv6 フェールオーバーパートナーは、ステートフルアドレスと委任されたプレフィックスリースを保持します）。

最大クライアント リード タイム (MCLT) とリース時間の制限は DHCPv6 リースに適用され、有効な有効期間と優先リースの有効期間は、フェールオーバー ペアに定義された MCLT に制限されます。フェールオーバーペアで許可される最長のリース時間が、構成された優先有効期間を超え、構成された優先有効期間が構成された有効な有効期間よりも短い場合にのみ、優先ライフタイムと有効なリースの有効期間が異なる場合があります。委任されたプレフィックスは、DHCPv4 アドレスと同様に管理され、バランスが取れています。

最も大きな違いは、DHCPv6 フェールオーバー サーバーが各プレフィックスで使用可能なアドレスのバランスを取らず、アルゴリズムを使用して各サーバーがリースできる新しいアドレスを決定することです。アルゴリズムはアドレスの最下位ビットを使用し、メインサーバーは奇数アドレスを割り当てますが、バックアップサーバーは偶数アドレスを割り当てます。これは、クライアントが要求し、ランダムに生成されたアドレスに適用され、次の場合は適用されません。

- リースは既にクライアントに割り当てられています。
- クライアントに予約が存在します。
- 割り当てアルゴリズムのインターフェイス識別子が設定され、使用されます。この場合、インターフェイス識別子(EUI-64)ビットは一意であると想定され、グローバル ビットが設定されると、これらのアドレスはグローバル ビットが設定されていないのでランダムに生成されたアドレスと競合しません。

- クライアント予約はプレフィックスで構成されます。
- 拡張機能はアドレスを提供します。

フェールオーバー サーバー ペアの設定

ローカル クラスターとリージョン クラスターでフェールオーバー ペアを作成および同期できます。

フェールオーバーペアには、構成とサーバーが保持する状態情報という2つの主要な要素があります。主要な構成属性は、フェールオーバーペアの名前、ローカルサーバーの役割(メインまたはバックアップ)、およびパートナーのアドレスです。フェールオーバー状態は、サーバーを再ロードし、サーバーが起動時にこの状態データを処理するときに定義されます。

（注） Cisco プライム ネットワーク レジストラー 8.2 以降の DHCP フェールオーバーは、Cisco Prime ネットワーク レジストラー 8.1 以前のリリースの DHCP フェールオーバーと相互運用されません。メインサーバーとバックアップサーバの両方を同じメンテナンス期間内にアップグレードする必要があります。

関連項目

フェールオーバー ペアの追加 （68 ページ）

フェールオーバー ペアの同期 （74 ページ）

フェールオーバー チェックリスト （77 ページ）

フェールオーバー ペアの追加

メイン サーバーとバックアップ サーバーのクラスタに基づいて、DHCP フェールオーバー ペアを作成します。次に、フェールオーバー ペアの構成を同期して、スコープ、プレフィックス、ポリシー、およびその他の DHCP プロパティがサーバー間で一致するようにします。

フェールオーバー ペアを追加するには、次の手順を実行します。

ローカルおよびリージョン Web UI

ステップ 1 Deployメニューから、FailoverPairsDHCPサブメニューの下で選択し、「DHCP フェイルオーバー・ペアのリスト/追加」ページを開きます。

ステップ 2 [フェールオーバー ペア] ペインのPairアイコンをクリックします。 Add Failover

ステップ 3 [DHCP フェールオーバー ペアの追加] ダイアログ ボックスで、フェールオーバー ペア名を追加します。

これは必須であり、任意の区別名を指定できます。（フェールオーバー ペア名の変更 （94 ページ） を参照）。

ステップ 4 メインの DHCP サーバーのクラスタを選択します。これは、ローカルホストまたはその他の定義されたクラスターです。

ステップ 5 バックアップ DHCP サーバーのクラスタを選択します。これは、メイン サーバー クラスタと同じにすることはできませんが、メイン クラスタが localhost でない場合は localhost にする必要があります。

ステップ 6 [DHCP **フェールオーバー ペアの追加**]をクリックします。フェールオーバー ペアが作成されます。

ステップ 7 クライアントの最大リードタイム(mclt)やバックアップ率(backup-pct)など、追加の属性を設定できます。ほとんどのデフォルト値は最適化されています。ペアのフェールオーバーを一時的に無効にする場合を除き、既定でフェールオーバー属性を有効のままにします。

メイン クラスタ オブジェクトとバックアップ クラスタ オブジェクトに対して構成されている値を上書きする場合、または特定のトランスポートを無効にする場合は、メイン サーバー属性、バックアップ サーバー属性、メインIP6 アドレス属性、またはバックアップIP6 アドレス属性を指定できます(0.0.0.0 または 0:0 を指定して)アドレスの場合)。IPv4 アドレスと IPv6 アドレスの両方が使用可能な場合、フェールオーバーは両方のトランスポートで接続を試み、最初に起動した接続を使用します。

また、リレーヘルスチェックの項で属性を設定して、DHCP リレーの状態チェックを構成することもできます（「DHCP リレー ヘルス チェック （111 ページ）」を参照）。

[保存（Save）] をクリックして、これらの変更を保存します。

[DHCPフェールオーバーペアの編集（Edit DHCP Failover Pair）] ページ（詳細モード）では、次の属性を構成できます。

表 *8 :* フェールオーバー ペアの属性

属性	説明
メインサーバー (メイン)	フェールオーバー ペアのメイン サーバーを持つクラスターを識別します。
バックアップ サーバー(バックアップ)	フェールオーバー ペアのバックアップ サーバーを含むクラスターを識別します。
スコープ テンプレート (スコープ テンプレート)	スコープ テンプレートを指定したフェールオーバー ペアに関連付けます。
Failover Settings	
フェールオーバー	フェイルオーバー構成を有効にします。この属性を無効にすると、構成の基本を変更することなく、接続されているサブネットでのフェールオーバーをオフにします。

属性	説明
mclt	クライアントの最大リードタイムを秒単位で設定します。この属性は、クライアント・リースの有効期限を作成できるバックアップ・サーバーの前にどれだけの距離を設定するかを制御します。この値は、メインサーバーとバックアップ サーバーの両方で定義し、両方のサーバーで同じ値を指定する必要があります。
backup-PCt	メイン サーバーがバックアップ サーバーに送信する使用可能なアドレスの割合を制御します。この値をメインサーバーに設定します。バックアップ・サーバーに設定されている場合、(構成のコピーを有効にするために)無視されます。スコープでこの値を明示的に設定し、負荷分散を無効にしない限り、ここで設定した値が既定値になります。
dynamic-bootp-backup-pct	動的BOOTPが有効になっているスコープについて、メイン サーバーがバックアップ サーバーに送信する使用可能なアドレスの割合を決定します。定義されている場合は、メイン サーバーで定義する必要があります。バックアップ・サーバーで定義されている場合、(構成のコピーを有効にするために)無視されます。この値がまったく定義されていない場合、または値が 0 の場合は、代わりにbackup-pctが使用されます。動的 BOOTP がスコープで有効になっている場合、サーバーは PARTNER-DOWN 状態であっても、他のサーバーで使用可能なアドレスにリースを付与することは決してないので、このパラメーターは「backup-pct」とは別です。. MCLT は動的 BOOTP リースには意味を持ちません。
load-balancing	フェールオーバー ペアでロード バランシング (RFC 3074) が有効かどうかを決定します。デフォルトではディセーブルになっています。有効にすると、バックアップ pct は無視され、メイン サーバーとバックアップ サーバーは、フェールオーバー 関係のすべてのスコープ (つまり、backup-pctが 50% で構成されているかのように) に対してクライアントの負荷と使用可能なリースを均等に分割します。

属性	説明
再バインド制限	T1 (リバインド時間) を超えて、通常は要求、書き換え、またはREBIND要求に応答しないフェイルオーバー・パートナーがパートナーの代わりに応答するときに使用するT2 (再バインド時間) の制限を設定します。 ゼロ以外の値に設定すると、クライアントが更新を開始し、フェールオーバーが通常の状態になると、クライアントがフェールオーバー パートナーによってサービスを受けるのに迅速にクライアントを返す速度が速まります。 クライアントは通常、数秒後に更新要求を再送信し、指数バックグラウンド アルゴリズムを使用して再試行するので、再バインド制限の妥当な値は 60 ~ 600 秒の範囲である可能性があります。 **注**:DHCPv6 の場合、サーバーがserver-idオプションに基づいてサービスを提供する RENEW 要求をドロップすることもあります(フェールオーバー パートナーが操作可能でクライアントにサービスを提供する必要がある場合)。
safe-period	安全期間を秒単位で制御します。メイン サーバーとバックアップ サーバーの両方で同じである必要はありません。これは、使用セーフ期間が有効になっている場合にのみ意味を持ちます。
使用セーフ期間	オペレーター・コマンドを使用せずに、サーバーが PARTNER-DOWN 状態に入ることができるかどうかを制御します。無効にした場合、サーバーはオペレーター・コマンドを指定せずに PARTNER-DOWN に入ることはありません。
リレー ヘルス チェック	
リレーヘルスチェック	正常な状態の状態で、正常性チェックが有効かどうか、およびフェールオーバー通信がNORMAL状態の場合にどのプロトコルに対して有効にするかを設定します。無効(デフォルト)、v4のみ、v6のみ、またはその両方に設定できます。
rhc 応答しない時間	ヘルスチェックの応答なし時間を秒単位で設定します。これは、このサーバーが通信の問題があると想定して、パートナーに代わって要求に応答する前に、別のサーバー宛の (DHCPv4) DHCPREQUEST または (DHCPv6) 要求パケットを受信できない最小時間です。リレーとそのパートナーの間で。

属性	説明
rhc 要求数	ヘルスチェックパートナーリクエスト数を設定します。これは、(DHCPv4) DHCPDISCOVER または (DHCPv6) このサーバーが通常は応答しないクライアント要求に応答する前に、パートナーが応答できた要求の数です。
rhc 再起動時刻	ヘルスチェックの再起動時間を秒単位で設定します。この時間の間、パートナーが応答する (DHCPv4) DHCPDISCOVER または (DHCPv6) 要請要求を受信しない場合、監視対象リレーの時間とカウントが再開されます。これは、ヘルスチェックが非常に最近のデータに基づいていることを保証します。
rhc 警告間隔	パートナーとリレー エージェント間の通信に問題がある可能性があるために、サーバーがパートナーに応答する場合に警告メッセージをログに記録する間隔を最小間隔で設定します。
rhc 経過時間	パートナーとリレーの間の通信がダウンしていると判断されたときに、サーバーがパートナーに応答するまでに、クライアントが DHCPv4 秒フィールドまたは DHCPv6 経過時間 (8) オプションで報告する必要がある最小時間を設定します。 0 (推奨しない) に設定すると、サーバーはすべての要求に応答します。
rhc-サーバー設定	フェールオーバー パートナーの代理として要請に応答するときに使用する DHCPv6 サーバーの設定を設定します。サーバー設定オプションが設定されており、rhc-server-preference 値が低い場合にのみ使用されます。
フェールオーバー サーバーアドレス	
メインサーバー	メイン サーバーのフェールオーバー プロトコルに使用する IPv4 アドレスを制御します。この値が設定されていない場合は、メイン・クラスターに指定されたアドレスが使用されます。Cisco では、サーバーが設定管理およびクライアント要求に対して異なるインターフェイスで設定されている場合にのみ、この値を設定することを推奨します。 この値を 0.0.0.0 に設定すると、フェイルオーバー通信に IPv4 を使用できなくなります。 IPv4 と IPv6 の両方のアドレスが使用可能な場合、サーバーは TCP 接続用の両方のトランスポートを試み、最初に起動した場合はどちらを使用しても使用します。

属性	説明
backup-server	バックアップ サーバーのフェールオーバー プロトコルに使用する IPv4 アドレスを制御します。この値が設定されていない場合は、バックアップ クラスタに指定されたアドレスが使用されます。Cisco では、サーバーが設定管理およびクライアント要求に対して異なるインターフェイスで設定されている場合にのみ、この値を設定することを推奨します。 この値を 0.0.0.0 に設定すると、フェイルオーバー通信に IPv4 を使用できなくなります。 IPv4 と IPv6 の両方のアドレスが使用可能な場合、サーバーは TCP 接続用の両方のトランスポートを試み、最初に起動した場合はどちらを使用しても使用します。
main-ip6address	メイン サーバーのフェールオーバー プロトコルに使用する IPv6 アドレスを制御します。この値が設定されていない場合は、メイン クラスタで指定されているアドレスが使用されます。Cisco では、サーバーが設定管理およびクライアント要求に対して異なるアドレスで設定されている場合にのみ、この値を設定することを推奨します。 この値は 0:0 に設定できますを使用して、フェイルオーバー通信に対する IPv6 の使用を無効にします。 IPv4 アドレスと IPv6 アドレスの両方が利用可能な場合、サーバーは TCP 接続に両方のトランスポートを試行し、先に利用可能なものを使用します。
backup-ip6address	バックアップ サーバーのフェールオーバー プロトコルに使用する IPv6 アドレスを制御します。この値が設定されていない場合は、バックアップ クラスタで指定されているアドレスが使用されます。Cisco では、サーバーが設定管理およびクライアント要求に対して異なるアドレスで設定されている場合にのみ、この値を設定することを推奨します。 この値は 0:0 に設定できますを使用して、フェイルオーバー通信に対する IPv6 の使用を無効にします。 IPv4 アドレスと IPv6 アドレスの両方が利用可能な場合、サーバーは TCP 接続に両方のトランスポートを試行し、先に利用可能なものを使用します。

CLI コマンド

failover-pair name create main-cluster/address backup-cluster/address [attribute=value ...] を使用します。次に例を示します。

```
nrcmd> failover-pair example-fo-pair create Example-cluster Boston-cluster
```

関連項目

フェールオーバー チェックリスト （77 ページ）
フェールオーバー ペア名の変更 （94 ページ）
フェールオーバー ペアの同期 （74 ページ）
フェールオーバー サーバーの再起動 （94 ページ）

フェールオーバー ペアの同期

フェールオーバー ペアを作成したら、フェールオーバー ペアの構成を同期する必要があります。

ローカルおよびリージョン Web UI

ステップ 1 Deployメニューから、FailoverPairsDHCPサブメニューの下で選択し、「DHCP フェイルオーバー・ペアのリスト/追加」ページを開きます。

ステップ 2 [フェールオーバー] ウィンドウでフェールオーバー ペアを選択します。

ステップ 3 [DHCP フェールオーバー ペアの一覧表示/追加] ページで、[**フェールオーバー ペアの同期**]タブをクリックします。

地域 Web UI での同期については、『 』の「DHCP フェールオーバーCisco プライムネットワーク レジストラー 11.0 管理ガイドペアの管理」セクションを参照してください。

ステップ 4 同期の方向を選択します。同期の方向は、メインからバックアップ サーバーへ、またはバックアップからメイン サーバーに行うことができます。

ステップ 5 バックアップ サーバーのオブジェクトを置き換える主なサーバー オブジェクトの程度に応じて、同期操作を選択します。サーバーで実行できる基本的な同期操作を次に示します。

- Update operation：これはデフォルトで最も過激な操作です。更新の同期には、バックアップ サーバーの一意のプロパティに対する影響が最も少ないという点で適切です。
- Complete operation：この操作は、すべての初期同期に適しています。バックアップ サーバーの一意のプロパティの多くはそのまま維持しながら、更新操作よりも完全です。
- Exact operation：この操作は、単純なフェイルオーバー構成に適しています。

この操作では、一意の DHCP サーバーとバックアップ サーバー上の拡張ポイントを保持しますが、2 つのサーバーは、可能な限り相互のイメージをミラー化します。

（注） 初期フェールオーバー構成の場合は、[完全（Exact）] または [完全（Complete）] 操作を使用します。

オブジェクトのクラスで実行される関数の理解を深めるには、次の例を考えてみます。ここでは、メインサーバーとそのバックアップサーバーと次のオブジェクトがあります。

メインサーバー上	バックアップサーバー上
Name1=A Name2=C	Name2=B Name3=D

（注） この例では、メインサーバーからバックアップサーバーへのフェールオーバー同期を検討します。

各操作は、オブジェクトのクラスに対して異なる関数の組み合わせを実行します。選択した操作に基づいてオブジェクトに対して実行される 4 つの関数を次に示します。

- 変更なし:バックアップ サーバーのプロパティまたは値のリストは変更されません。

  たとえば、結果は Name2=B、名前 3=D になります。

- ensure：メインサーバーオブジェクトのコピーがバックアップに存在することを確認します。メインサーバー オブジェクトと同じ名前のターゲット サーバー オブジェクトは変更されず、ターゲット サーバー上にないオブジェクトは追加され、ターゲット サーバー上のオブジェクトだけが変更されません。

  たとえば、結果は、名前 1=A、名前 2=B、名前 3=D になります。

- replace：ターゲットサーバーの既存のオブジェクトが、同じ名前のメインサーバーオブジェクトに置き換えられることを確認します。また、ターゲット サーバー上にないオブジェクトも追加され、ターゲットサーバー上のオブジェクトだけが変更されません。唯一の例外は、オプション リストがリスト エントリを比較するために抽出されるポリシーとオプション定義セットです。

  たとえば、結果は、名前 1=A、名前 2=C、名前 3=D になります。

  （注） メインサーバー上のクライアントを削除し、フェールオーバー同期 Update または Complete 操作を実行してバックアップのエントリを削除した後、クライアントはバックアップから削除されません。バックアップのクライアントエントリをメインサーバーから削除した後に、クライアント エントリを削除する唯一のフェールオーバー同期操作は、フェールオーバー同期の正確な操作です。

- exact：メインサーバーオブジェクトの正確なコピーをバックアップサーバーに配置し、一意のものを削除します。つまり、ターゲットサーバーのオブジェクトは、メインサーバーのオブジェクトと同一になります。

  たとえば、結果は Name1=A、名前 2=C になります。

詳細については、次の表を参照してください。この表は、選択した操作（更新、完了、正確）に基づいてオブジェクトに対して実行される機能（変更なし、確認、置換、または正確な操作）に関する情報を提供します。

表 *9:* フェールオーバー ペアの同期関数

データの説明	更新（Update）	完了（Complete）	完全一致（Exact）
DHCP サーバー： クライアントクラス プロパティ クライアント ホスト名処理プロパティ 動的 DNS プロパティ フェールオーバー チューニングのプロパティ	置換	置換	置換
その他のすべてのプロパティ	変更なし	置換	置換
LDAP リモート サーバー	確認	置換	正確な操作
ポリシー： オプション リストのプロパティ パケット ブート ファイルのプロパティ その他のプロパティ	確認 確認 置換	置換 置換 置換	正確な操作 正確な操作 正確な操作
クライアント	正確な操作	正確な操作	正確な操作
クライアントクラス	置換	置換	正確な操作
スコープとスコープの予約	正確な操作	正確な操作	正確な操作
Links	正確な操作	正確な操作	正確な操作
プレフィックスとプレフィックス予約	正確な操作	正確な操作	正確な操作
DNS 更新 コンフィギュレーション	置換	置換	正確な操作
トラップの設定	確認	置換	正確な操作
VPNs	置換	置換	正確な操作
オプション キー（Keys）	置換	置換	正確な操作
拡張子(拡張ファイルをコピーする必要があります。	確認	置換	正確な操作
拡張ポイント	置換	置換	置換
オプションの定義： カスタム オプション リスト ベンダー オプション リスト	確認	置換	正確な操作

データの説明	更新（Update）	完了（Complete）	完全一致（Exact）
DHCP リスナー設定	確認	置換	正確な操作

ステップ 6 [Reportフェールオーバー ペアの同期] ページをクリックします。

- 同期の方向を選択するオプションと、同期操作の目的のモードをチェックするオプションもあります(**更新**,**完了**,**正確**)。目的の値を確認し、[**レポート**] をクリックします。結果のページには、同期を実行した場合に同期が適用される変更セット エントリが表示されます。[**更新の実行**] をクリックするか、[**戻る**] をクリックして [フェールオーバー ペアの同期] ページに戻ります。

ステップ 7 Save をクリックして変更を保存します。

ステップ 8 [DHCP フェールオーバー ペアの一覧/追加] ページで、[**フェールオーバー サーバーの管理**] タブをクリックします。

ステップ 9 [**サーバーの再起動**]アイコンをクリックして、バックアップ サーバーをリロードします。

ステップ 10 リースを取得してみてください。

ステップ 11 [フェールオーバー サーバーの管理] タブで、サーバーの正常性を確認します。また、[**ログ**] タブをクリックして [サーバーのログ] ページのログ エントリを表示し、サーバーが NORMAL フェールオーバー モードになっていることを確認します。ログ ファイルには、次のような項目が含まれている必要があります。

```
06/19/2003  9:41:19 name/dhcp/1 Info Configuration 0 04092 Failover is enabled server-wide. Main
 server name: '192.168.0.1',
backup server name: '192.168.0.110', mclt = 3600, backup-pct = 10, dynamic-bootp-backup-pct = 0,
 use-safe-period: disabled,
safe-period = 0.
```

CLI コマンド

failover-pair 名前を sync使用update{ |complete |exact} [{**メインからバックアップ**|**メインへのバックアップ**}][-**レポートのみ**|-**レポート**]:

```
nrcmd> failover-pair example-fo-pair sync exact main-to-backup -report
```

フェールオーバー チェックリスト

フェールオーバー ペアを作成したら、フェールオーバー サーバーの構成を同期する必要があります。このチェックリストを使用して、有効なフェイルオーバー構成に備えます。

- DHCPv4 スコープ、DHCPv6 プレフィックス、DHCPv6 リンク、予約 (IPv4 および IPv6)、選択タグ、ポリシー、DHCP オプション、IP アドレス、クライアント クラス、動的 DNS 更新、動的 BOOTP、VPN、DHCP 拡張機能、DHCP 拡張、LDAP サーバー、およびアドレスを複製します。単純なフェールオーバー シナリオでフェールオーバー サーバー ペアを同期させることによって、パートナー サーバー上の構成。
- バックアップ サーバーが、メイン サーバーがダウンしている間に妥当な時間のリースを提供できるように、両方のパートナーが十分な範囲のアドレスで構成されていることを確認します。

- BOOTP (DHCP) リレー エージェント (IP ヘルパー) を使用する場合は、すべての BOOTP リレー エージェントが両方のパートナーを指するように構成します。Cisco プライムネットワーク レジストラーでは、この機能は自動的には検出されません。

  BOOTP 構成エラーを検出するには、ライブ テストを実行し、メイン サーバーを定期的にサービス停止にして、バックアップ サーバーが DHCP クライアントで使用できることを確認します。

シナリオに基づいたフェールオーバーパラメータの設定

設定する必要がある詳細なフェールオーバー プロパティを次に示します。

- バックアップの割合（バックアップの割合の設定 （78 ページ）を参照）
- バックアップ割り当ての境界（バックアップ割り当て境界の設定 （93 ページ）を参照）
- 最大クライアントリードタイム（MCLT）（最大クライアント リードタイムの設定 （80 ページ）を参照）
- 安全期間（フェールオーバー セーフ期間を使用して、サーバーを PARTNER-DOWN 状態に移行する （81 ページ）を参照）
- 要求および応答パケットバッファ（DHCP 要求と応答パケット バッファの設定 （84 ページ）を参照）
- ロードバランシング（ロード バランシングの設定 （84 ページ）を参照）

バックアップの割合の設定

ネットワーク パーティションに関係なくフェールオーバー パートナーを動作させ続けるには (両方のサーバーがクライアントと通信できるが、互いに通信できない場合)、単一サーバーのアドレスよりも多くのアドレスを割り当てます。メイン サーバーを構成して、各スコープおよびプレフィックスの委任プレフィックスで現在使用可能なアドレスの割合をバックアップ サーバーに割り当てます。これにより、これらのアドレスはメイン サーバーで使用できなくなります。バックアップ サーバーは、メイン サーバーとの間で話ができず、ダウンしているのかどうかを確認できない場合に、これらのアドレスを使用します。

メイン サーバーがアドレス プールの残高が大幅に不足しているか、サーバーにリースがないことを検出した場合、フェールオーバー ペアが NORMAL 状態で機能している場合でも、使用可能なリースまたは他の利用可能なリースのプールは再調整されます。フェールオーバー ペアはフェールオーバー中に注意深く監視する必要があり、フェールオーバー パートナーが長時間ダウンしている場合は、フェールオーバー パートナーを PARTNER-DOWN 状態に移行するためにオペレーターの介入が必要になる場合があります。

現在使用可能なアドレスの割合は、フェールオーバー ペアまたは DHCPv4 スコープ (CLI の名前setfailbackup-pctまたはscope名前setbackup-pct)failover-pairにbackup-pct属性を設定することで設定できます。デフォルトのバックアップの割合は 50 % です。DHCPv6 プレフィックスの委任プレフィックスは、バックアップ pct に対応する 50% に固定されます。

フェールオーバー ペア レベルでバックアップの割合を設定すると、その属性で設定されていないすべてのスコープの値が設定されることに注意してください。ただし、スコープ レベルで設定すると、バックアップの割合はフェールオーバー ペア レベルのバックアップ率よりも優先されます。フェールオーバーfailover-pairペア(CLI の名前enableload-balancing)に対してロードバランシング属性が有効になっている場合、バックアップの割合は 50% に固定され、(フェールオーバー ペアまたはスコープ上の) バックアップパーセンテージ属性は無視されます。

バックアップの割合は、メイン サーバーで障害が発生した場合にバックアップ サーバーが新しいクライアントにサービスを提供し続けることができるように、十分な大きさに設定する必要があります。バックアップの割合は、使用可能なアドレスの数に基づいて計算されます。通常のリース活動の過程でメイン サーバーの使用可能なアドレス プールがそれより低い場合、メイン サーバーは定期的にアドレスを (1 時間に 1 回) 回収するので、拡張停止が予想される場合は、バックアップ率を大きな値に設定しても問題ありません。定義済みの割合。たとえば、バックアップ率が 60% に設定されている場合、メイン サーバーはアドレス プールが 60% を下回るとアドレスを再利用します。

（注） フェールオーバーの負荷分散が有効な場合、メイン サーバーとバックアップ サーバーは、使用可能なリースのバックアップ率を維持するために、使用可能なリースをアクティブに移動します。ロード バランシングの設定 （84 ページ）を参照してください。

割合は、新しいクライアントの到着率とネットワーク オペレータの応答時間によって異なります。新しいDHCPクライアントの到着率とネットワーク管理スタッフの応答時間によって異なります。バックアップ サーバーは、メイン サーバーがダウンしているかどうかを認識できない時間に到着するすべての新しいクライアント要求を満たすのに、各スコープから十分なアドレスを必要とします。PARTNER-DOWN 状態の間でも、バックアップ サーバーは、リースを再割り当てする前に、最大クライアント リード タイム (MCLT) とリース時間の期限が切れるまで待機します。最大クライアント リードタイムの設定 （80 ページ）を参照してください。この時間が経過すると、バックアップサーバーは以下を提供します。

- プライベート プールからのリース。
- メイン サーバー プールからのリース。
- 新しいクライアントに期限切れのリース。

稼働時間内に、オペレーターは、2 時間以内に COMMUNICATIONS-INTERRUPTED 状態に応答して、メイン・サーバーが稼働しているかどうかを判別します。バックアップ サーバーは、その2時間の間に到着する可能性のある新しいクライアントの数を適切に上限にサポートするのに十分なアドレスを必要とします。

営業時間外には、未知のクライアントの到着率は低くなる可能性があります。オペレータは通常、同じ状況に対して12時間以内に応答することができます。バックアップ サーバーは、その12時間の間に到着する可能性のあるクライアント数の上限を十分にサポートするのに十分なアドレスを必要とします。

バックアップ サーバーが単独で制御するアドレスの数は、2つの数値のうち大きい値です。この数値は、各スコープで現在利用可能な(未割り当て)アドレスの割合として表すことができま

す。クライアント クラスを使用する場合、一部のクライアントでは一部のスコープ セットしか使用できません。

（注） フェールオーバー中に、クライアントは、有効期限が構成されている量よりも短いリースを取得することがあります。これは、サーバー パートナーの同期を維持する通常の部分です。通常、これは最初のリース期間、または通信中断状態の間にのみ発生します。

関連項目

BOOTP バックアップの割合 （111 ページ）

最大クライアント リードタイムの設定

リース期間の調整を制御するフェールオーバーペアのプロパティ、つまりクライアントの最大リードタイム (MCLT) を設定できます。MCLT は、サーバー間の接続が不確実な時間帯に調整します。これは、1 つのサーバーが、最初にパートナーとの長い時間をネゴシエーションせずに、クライアントにリースを許可(または拡張)できる最大時間です。今回は、次の意味があります。

- クライアントは、MCLT の長さのリースのみを最初に (またはパートナーが通信していない場合) 受信することがあります。つまり、フェールオーバーを行わない場合よりも早くリースを更新する必要があります。この更新時に、クライアントは、(パートナーが通信していない場合を除き) 完全なリース時間を取得する必要があります。
- サーバーが PARTNER-DOWN 状態になると、パートナー・ダウン時間の後またはパートナーと通信した最新のリース有効期限が過ぎるまで、MCLTが終了するまで待たなければなりません。パートナーに通知される最新の最小有効期限は、通常、通信が中断される前の最後のクライアント リース要求のリース時間の 1.5 倍です。
- 1 つのパートナーが何をしたか(リース データベースを失ったときなど) について不確定な状態でフェールオーバー回復が発生した場合、パートナーはMCLT期間のリース活動を同期後に制限してから、フェールオーバーを通常の状態に戻す必要があります。操作。

デフォルト MCLT は 1 時間で、ほとんどの構成に最適です。フェールオーバー プロトコルで定義されているように、クライアントに対して指定されたリース期間は、MCLT にフェールオーバーパートナーから受信した最後の潜在的な有効期限を超える期間、または現在の時刻を超える値を超えることはありません。そのため、最初のリース期間は、更新の場合に予想よりも1時間長い場合があります。実際のリース時間は、メインサーバーが復帰したときに再計算されます。

フェールオーバーによる遅延更新の使用のためにMCLTが必要です。遅延更新を使用すると、サーバーはパートナーを更新する前にクライアントにリースを発行または更新し、更新をバッチ処理できます。サーバーがダウンしてリース情報をパートナーに伝えることができない場合、パートナーは、最後に有効期限が何であるかに基づいて、リースを別のクライアントに再提供しようとします。MCLTは、クライアントが更新する機会の追加ウィンドウがあることを保証します。MCLT とリースの提供と更新が機能する方法は次のとおりです。

1. クライアントはDHCPDISCOVERまたはDHCPv6要請をサーバーに送信し、必要なリース期間 (たとえば、3 日間) を要求します。サーバーは、MCLT (既定では 1 時間) の初期リース期間でDHCPOFFERまたはDHCPv6アドバタイズを使用して応答します。クライアントは MCLT リース期間を要求し、サーバーはそれを確認します。
2. サーバーは、パートナーに、クライアントのリース有効期限を含むバインド更新を、現在の時刻と MCLT として送信します。更新プログラムには、現在の時刻にクライアントが希望する期間に加えて、クライアントの希望期間の半分 (3 +1.5 = 4.5 日) の有効期限が含まれます。パートナーは、潜在的な有効期限を確認し、それによってトランザクションを保証します。
3. クライアントがリースの途中で (100 分) で更新要求を送信すると、サーバーはクライアントの希望するリース期間 (3 日) を確認します。サーバーは、現在の時間に希望のリース期間 (3 日) を加えたリース期限と、潜在的な有効期限 (4.5 日) を持つパートナーを更新します。（**ステップ** 2 の説明を参照）。パートナーは、この潜在的な有効期限が 4.5 日であることを確認します。このようにして、メインサーバーは、クライアントに常に提供できるように、クライアントリース期間を常に理解して、パートナーにクライアントを導いさせようとします。

MCLTに正しい値はありません。選択にはさまざまな要因の間には明確なトレードオフがあります。ほとんどの人は1時間のプリセット値を効果的に使用し、ほぼすべての環境でうまく機能します。短い MCLT と長い MCLT のトレードオフの一部を次に示します。

- Short MCLT—：MCLT 値が短い場合は、PARTNER-DOWN 状態に入った後、サーバーがパートナー IP アドレスを DHCP クライアントに割り当てるまで、少しだけ待つ必要があることを意味します。さらに、リースの期限が切れてから、そのアドレスを別のDHCPクライアントに再割り当てする必要があります。ただし、すべての新しいDHCPクライアントに提供される初期リース間隔が短くなるため、トラフィックが増加します。また、COMMUNICATIONS-INTERRUPTED状態のサーバーが与えることができるリース拡張は、サーバーが望ましいクライアントリース期間の前後にこの状態になった後にのみMCLTです。サーバーがその状態を長期間保持している場合、渡すリースは短くなり、そのサーバーの負荷が増加し、問題が発生する可能性があります。
- Long MCLT：MCLT の値が長い場合、初期リース期間が長くなり、COMMUNICATIONS-INTERRUPTED 状態のサーバーがリースを延長できる時間（必要なクライアントリース期間の前後にリースが延長された後）が長くなります。ただし、PARTNER-DOWN 状態になるサーバーは、パートナー アドレスを新しい DHCP クライアントに割り当てる前にMCLTを長く待つ必要があります。これは、この期間をカバーするために追加のアドレスが必要であることを意味する場合があります。また、PARTNER-DOWN 状態のサーバーは、アドレスを別の DHCP クライアントに再割り当てする前に、リースの有効期限が切れるまでMCLTが長くなってから待機する必要があります。

フェールオーバーセーフ期間を使用して、サーバーをPARTNER-DOWN状態に移行する

一方または両方のフェイルオーバー・パートナーが、通信中断状態に移行する可能性があります。この状態の間は、重複するアドレスを発行できません。ただし、サーバーが実行できる処

理には制限があるため、長期間にわたってこの状態のサーバーを使用することはお勧めできません。メインサーバーは期限切れのリースを再割り当てできず、バックアップサーバーのプールからアドレスが不足する可能性があります。COMMUNICATIONS-INTERRUPTED 状態は、サーバーが数分から数日の一時的な通信障害を簡単に生き残るために設計されました。クライアントの到着と出発の速度によっては、サーバーがこの状態で短時間だけ効果的に機能する場合があります。その後、サーバーを PARTNER-DOWN 状態に移行して、サーバーが再同期するまでリース機能を完全に引き継ぐようにすることをおお方が良いでしょう。

サーバーが PARTNER-DOWN 状態に移行する方法は 2 つあります。

- User action：管理者は、実際の正確な評価に基づいて、サーバーを PARTNER-DOWN 状態に設定します。フェールオーバー プロトコルがこれを正しく処理します。両方のパートナーを PARTNER-DOWN に設定しないでください。
- Failover safe period expires：サーバーが長時間無人で実行される場合、自動的に PARTNER-DOWN 状態を入力する方法が必要です。

ネットワークオペレータは、サーバーがダウンしているか、通信不能であることをすぐには感知しない場合があります。したがって、COMMUNICATIONS-INTERRUPTED 状態に移行するサーバーに応答する時間をネットワークオペレーターに提供するフェイルオーバーセーフ期間があります。セーフ期間中に、オペレータが両方のサーバーがまだ稼働していることを判断し、実行されている場合は、ネットワーク通信障害を修正するか、安全期間が経過する前にいずれかのサーバーを停止することが唯一の要件です。

セーフ期間の長さはインストールに固有であり、プール内の未割り当てアドレスの数と、アドレスを必要とする未知のクライアントの予想到着率によって異なります。

Cisco Prime Network レジストラー では、フェールオーバーペアに対して use-safe-period 属性がデフォルトで有効になり、デフォルトのセーフ期間は4時間です。これにより、フェールオーバー パートナーが 4 時間 COMMUNICATIONS-INTERRUPTED 状態になると、安全期間が過ぎた後に PARTNER-DOWN 状態が自動的に入力されます。この設定がネットワークに適しているかどうかを確認し、ネットワーク要件に基づいてセーフ期間を調整する必要があります。

さらに、この安全な期間中は、どちらのサーバーも既存のクライアントからの更新を許可しますが、重複アドレスを発行する可能性が大きなリスクがあります。これは、一方のサーバーが、もう一方のサーバーが動作中に突然PARTNER-DOWN状態に入る可能性があるためです。この問題を回避するには、使用セーフ期間のデフォルトの設定を変更するか、フェールオーバーペアが互いに接続できなくなると操作担当者に警告する運用手順を策定することが重要です。特に、ネットワーク通信障害が発生した場合、安全期間が経過する前にオペレーターの介入が必要です。いずれかのフェールオーバー サーバーをオフラインにするか、または安全期間の使用属性を両方のサーバーで無効にしてから、安全な期間を過ぎる必要があります。

（注） Cisco プライムネットワーク レジストラーでは、使用セーフ期間がデフォルトで有効になっています。これがネットワークに適しているかどうかを確認し、使用セーフ期間を無効にするか、ネットワーク要件と監視に基づいてセーフ期間を調整する必要があります。

安全期間に必要な追加アドレスの数は、サーバーが検出した新しいクライアントの予想される合計と同じにする必要があります。これは、未処理のリースの合計ではなく、新しいクライア

ントの到着率に依存します。たとえ短い安全期間しか与えられない場合でも、アドレスの不足や新しいクライアントの到着率が高いため、DHCPが1時間で修正可能な小さな問題を乗り越えることで、実質的に利益を得ることができます。重複アドレス割り当ての可能性が最小限であり、解決された障害後の再統合は自動的に行われ、オペレーターの介入は必要ありません。

フェールオーバー セーフ期間の長さが MCLT の長さを超え、フェールオーバー サーバーが安全な期間のために PARTNER-DOWN 状態になった場合、サーバーはパートナーの他のリースをDHCPクライアントにすぐに割り当て始めることができます。この利点は、サーバーに割り当てる追加のリースが必要です。ただし、ネットワーク通信障害が発生した場合に、安全な期間内にオペレーターの介入が必要になることが欠点です。フェールオーバーサーバーをオフラインにするか、または安全期間の両方で使用セーフ期間属性を無効にしてから、安全な期間が経過する必要があります。オペレーターの介入がなければ、両方のフェイルオーバー・サーバーは PARTNER-DOWN 状態に移行し、パートナー・アドレスを新規の DHCP クライアントに割り当て始めます。

手動介入を使用するか、PARTNER-DOWN状態に移行するための安全な期間を使用するかを決定するために、従うガイドラインをいくつか示します。

- 企業ポリシーで手動による介入を最小限に抑える場合は、安全期間を設定します。セーフ期間を有効にするには、フェールオーバーペア属性の使用セーフ期間を有効にします。次に、DHCP 属性のセーフ期間を設定して、期間を設定します (デフォルトでは 4 時間)。この期間を十分に長く設定して、運用担当者が通信障害の原因を調査し、パートナーが本当にダウンしていることを確認できるようにします。
- 企業ポリシーがどのような状況でも競合を避ける場合は、明示的なコマンドを使用しない限り、どちらのサーバーも PARTNER-DOWN 状態にしないでください。管理カバレッジがない期間に新しいクライアント到着を処理できるように、バックアップサーバーに十分なアドレスを割り当てます。パートナーが通信中断フェールオーバー状態の場合は、パートナーが [フェールオーバー サーバーの管理] タブで PARTNER-DOWN を設定できますSet Partner Down。この設定は、通信の開始中断属性の値に初期化されます。（通常の Web UI モードでは、この日付を初期化された日付より前の値に設定することはできません。エキスパート Web UI モードでは、この値を任意の日付に設定できます。

  パートナーサーバーの名前を指定して、CLI で failover-pair name setPartnerDown 日付を使用します。これにより、コマンドで日時を指定しない限り、パートナーとのフェイルオーバーを実行しているすべてのスコープがただちに PARTNER-DOWN 状態に移行します。この日時は、パートナーが最後に操作可能であることが判明した日時です。

  CLI でsetPartnerDownを使用し、パートナーが最後に動作することが確認された日時を指定すると、フェールオーバー サーバーは setPartnerDown コマンドで指定された時刻から MCLT を計算します。setPartnerDown コマンドに日付と時刻が指定されていない場合、フェールオーバーサーバーが、COMMUNICATIONS-INTERRUPTED 状態に移行した時点からMCLTが計算されます。ネットワーク通信障害が発生した場合は、パートナーが最後に動作可能であることが判明した実際の時刻を setPartnerDown コマンドで指定することが重要です。そうしないと、重複する IP アドレスが発生する可能性があります。

  日付を指定する場合、次の 2 つの規則があります。

- –num 単位（過去の時刻）は、num は 10 進数で、単位は秒、分、時、日、週の場合は s、m、h、d、または w です。たとえば、3 日間は -3d と指定します。
- 月（名前またはその最初の 3 文字）、日、時（24 時間表記）、年 （完全に指定された年または最後の 2 桁）。この例では、2002 年 10 月 31 日の午前 12 時にメイン サーバーがダウンしたことをバックアップ サーバーに通知します。

```
nrcmd> failover-pair dhcp2.example.com. setPartnerDown -3d

nrcmd> failover-pair dhcp2.example.com. setPartnerDown Oct 31 00:00:00 2001
```

注 CLI で日付と時刻を指定する場合は、nrcmd プロセスにローカルな時刻を入力します。サーバーがこのプロセスとは異なるタイム ゾーンで実行されている場合は、サーバーが実行されているタイム ゾーンを無視し、代わりにローカル時刻を使用します。

DHCP 要求と応答パケット バッファの設定

DHCP フェールオーバーでは、限られた数のバインド更新を未処理にできます ((エキスパート モードの)max-unacked-bndupdフェールオーバー ペア属性を使用して設定します。max-un-bndupd のデフォルト値は 1/5(20%)の値の最大-dhcp-requests値、および最低 100 および最大 dhcp 要求です。サーバーは、フェールオーバーに対応するために追加の要求バッファを割り当てます (フェールオーバーに使用できるリソースが必要なため)。

ロード バランシングの設定

通常のフェールオーバー モードでは、フェールオーバー パートナーが NORMAL 通信モードの場合、メイン DHCP サーバーはクライアントにサービスを提供する負担の大部分を担います。メイン サーバーは、すべての新しいクライアント要求に対応するだけでなく、バックアップ パートナーからの要求の更新と再バインド、および期限切れのリースを処理する必要があります。単純なフェールオーバー設定シナリオで 2 台のサーバー間で負荷をより均等に分散するために、Cisco Prime Network レジストラーではロードバランシング機能が導入されました(RFC 3074 に基づく)。

フェールオーバー負荷分散により、両方のサーバーがクライアントに対してアクティブにサービスを提供し、両方のサーバーが同じクライアントにサービスを提供するリスクを冒さずに、各サーバーがサービスを提供する一意のクライアントを決定できます。フェールオーバー負荷分散は、サーバーが NORMAL モードの間にのみ適用されます。他の状態では、両方のサーバーがクライアントに応答できます。

RFC 3074 によると、サーバーはクライアント識別子オプションの値またはハードウェア アドレスに基づいて、サーバーが受信する要求ごとにハッシュ値を計算します。ハッシュ値がそのサーバーに割り当てられている場合、要求は処理されます。

フェールオーバー負荷分散が有効な場合、サーバーはクライアントの負荷を均等に分割します。メイン・パートナーはハッシュ値の 50% を処理し、バックアップ・パートナーは残りの 50% を処理します。

フェールオーバー パートナーは、バックアップ サーバーで利用可能なリースのバランスを定期的に調整するか、またはスコープまたはプレフィックスがリースから外れていると検出された直後に行います。

各パートナーは、パートナーが NORMAL モードでない場合は、すべてのクライアントに応答します。各パートナーは、割り当てられたハッシュ値のクライアントからのブロードキャスト DHCPDISCOVER メッセージまたは SOLICIT メッセージにのみ応答します。

ブロードキャスト DHCPREQUEST メッセージまたは REBIND メッセージの場合、サーバーは、(サーバー ID オプションに基づいて) 対象のメッセージである場合にのみ応答します。したがって、対象サーバーがメインサーバーであり、ダウンしている場合、バックアップはクライアントにサービスを提供しません(リースを解放しない限り)。また、ブロードキャストブート、DHCPINFORM、および情報要求要求も負荷分散されます。

フェールオーバー属性である再バインド制限は、クライアントを NORMAL 状態のフェールオーバー パートナーに戻す方法を提供します。再バインド制限値は 60 ~ 600 秒 (1 ~ 10 分) の範囲で設定することをお勧めします。この属性は、T2 (再バインド時間) を制限するために T1 に追加される時間間隔 (更新時間) を指定します。指定すると、フェールオーバー NORMAL 状態でクライアントに応答しなかったフェールオーバー パートナー (要請要求に応答しない) が、要求、書き換え、または REBIND にクライアントに応答すると、次の 2 つのことが発生します。

- T2 (再バインド時間) を T1 (更新時間) にこの属性値を加えた値に設定します。
- フェールオーバーが正常な状態の場合、このサーバーに送信された RENEW 要求には応答しません。

この 2 つのアクションにより、クライアントが更新を開始し、フェールオーバーが正常な状態になると、クライアントは他のパートナーからかなり迅速に処理されます。更新が行われるサーバーは応答せず、クライアントは REBIND 状態にかなり早く入ります (指定された再バインド制限に基づいて)。

関連項目

ロード バランシングの設定 （85 ページ）

ロード バランシングの設定

Web UI で、ペアのフェールオーバー プロパティを設定する場合（フェールオーバー サーバー ペアの設定 （68 ページ）を参照）、フェールオーバーの load-balancing を有効または無効にする必要に応じて、[フェールオーバー設定（Failover Settings）] 属性の load-balancing 属性を有効または無効にします。CLIで、failover-pair name set load-balancing を使用します。

（注） 変更を適用するには、メインとバックアップの両方でDHCPサーバーを再起動する必要があります。

DHCP フェールオーバーからの回復

通常どおり稼働している間、フェールオーバー パートナーは状態遷移を行います。フェールオーバー サーバーの 1 つに障害が発生した場合、パートナーはプライベート プールを使用してリースの提供と更新を引き継ぎます。メイン サーバーが再度動作すると、管理者が操作しなくても、パートナーと再統合されます。

次のセクションでは、DHCP フェールオーバーの確認方法、DHCP フェールオーバー イベントの監視方法、サーバーがさまざまな状態になったときの動作、およびサーバーの統合方法について説明します。

フェールオーバーの確認

フェイルオーバーを確認するには、次の手順に従います。

ステップ 1 1 つのサーバーから別のサーバーに ping を実行して、TCP/IP 接続を確認します。両方のサーバーにクライアントを転送するようにルーターが構成されていることを確認します。

ステップ 2 [DHCP サーバーの管理] ページまたは [DHCP フェールオーバー ペアの一覧/追加] ページの [**関連サーバー**]dhcp getRelatedServersアイコンをクリックするか、CLI で使用して、サーバーが通常モードであることを確認します。

ステップ 3 起動後、クライアントにリースを取得してもらいます。

ステップ 4 少なくともフェールオーバーの詳細を含むように、メイン サーバーのログ設定を設定します。

ステップ 5 メインサーバーの name_dhcp_1_log ログファイル（in /var/nwreg2/{local | regional}/logs）に、各サーバーからの DHCPBNDACK または DHCPBNDUPD メッセージ（IPv4 の場合）と BNDUPD6 または BNDACK6 メッセージ（IPv6 の場合）が含まれていることを確認します。

ステップ 6 フェールオーバーが正常な状態であるため、バックアップ サーバーがドロップするメッセージがバックアップ サーバーのname_dhcp_1_logログ ファイルに含まれていることを確認します。

ステップ 7 **手順** 2 を繰り返します。

関連項目

統合中のステート移行 （89 ページ）

シナリオに基づいたフェールオーバー パラメータの設定 （78 ページ）

DHCP フェールオーバーのモニターリング

メイン・フェイルオーバー・サーバーがダウンすると、バックアップ・サーバーは COMMUNICATIONS-INTERRUPTED 状態に移行します。バックアップ サーバーは、メイン サーバーが停止しているか、バックアップ サーバーと通信できないかを判断できません。停止の性質に応じて、状況をモニターし、以下のステップに従う必要があります。

1. 両方のフェールオーバー サーバーを監視し、メイン サーバーがダウンした場合は直ちに処理を実行します。
2. バックアップ サーバーが最初に引き継いだ時点で、メイン サーバーを操作に戻します。
3. MCLT 内でメイン サーバーを運用できる場合は、これ以上必要はありません。
4. MCLT の期限が切れるまでメイン・サーバーが作動しない場合は、バックアップ・サーバーを PARTNER DOWN 状態に移動します。バックアップ サーバーで、CLI **でフェールオーバー ペア**名 setPartnerDown [日付] を使用します。
5. メイン サーバーが動作している場合は、再起動する前にバックアップ サーバーに接続できることを確認します。

詳細については、統合中のステート移行 （89 ページ）を参照してください。

フェールオーバーの状態と遷移

通常の運用中、フェールオーバー パートナーは状態間の移行を行います。状態遷移のすべてのアクションが完了するまで、現在の状態にとどまります。通信が失敗した場合、次の状態の条件が満たされるまで、現在の状態にとどまります。状態とその遷移については、次の「表 10 : フェールオーバーの状態と遷移 」で説明します。

表 *10 :* フェールオーバーの状態と遷移

状態	サーバーのアクション
STARTUP	パートナーに連絡して状態を確認し、短い時間(通常は数秒)の後に別の状態に移行します。
NORMAL	パートナーと通信できます。メイン サーバーとバックアップ サーバーは、次の状態で動作が異なります。 • メイン サーバーは、プールを使用してすべてのクライアント要求に応答します。パートナーがバックアップ プールを要求すると、メイン サーバーによってバックアップ プールが提供されます。 • バックアップ サーバーは、更新要求と再バインド要求にのみ応答します。メイン サーバーからバックアップ プールを要求します。

状態	サーバーのアクション
COMMUNICATIONS-INTERRUPTED	パートナーと通信できない場合、パートナーと通信している場合でも、そのパートナーとの通信がダウンしている場合でも、パートナーと通信することはできません。接続が失敗して回復したとき、または操作可能と非稼働状態の間でサーバーが循環する場合は、この状態と NORMAL 状態の間を循環します。この間、サーバーは重複するアドレスを提供できません。 この状態の間、通常は、サーバーを介入して PARTNER-DOWN 状態に移行する必要はありません。ただし、これは実用的でない場合もあります。この状態で実行されているサーバーは、使用可能なプールを効率的に使用していません。これにより、サーバーがクライアントに効果的にサービスを提供できる時間を制限できます。 サーバーは、通信中断状態で制限されます。 • 期限切れのアドレスを別のクライアントに再割り当てすることはできません。 • 現在のリース時間を超える最大クライアントリードタイム(MCLT)を超えるリースまたは更新を提供することはできません。MCLT は、バックアップ サーバーが考えているよりもクライアントリースの有効期限がどれくらい前に入っているかを制御する、わずかな追加時間です。 • バックアップ サーバーは、通常は小さなプールしか持っていないので、新しいクライアントにアドレスを使い果たすことができます。 サーバーは、割り当てられたアドレスの数と新しいクライアントの到着率によってのみ制限されます。新しいクライアントの到着率または離職率が高い場合は、サーバーをより迅速に PARTNER-DOWN 状態に移行する必要があります。
PARTNER-DOWN	次のいずれかの事実に基づいて、それが唯一の運用サーバーであるかのように動作します。 • パートナーはシャットダウン中に通知を行いました。 • 管理者は、サーバーを PARTNER-DOWN 状態にします。 • 安全期間が切れ、パートナーは自動的にこの状態に入りました。 この状態では、サーバーは、他のサーバーがまだ動作する可能性があることを無視し、別のクライアントのセットをサービスできます。すべてのアドレスを制御し、リースとエクステンションを提供し、アドレスを再割り当てすることができます。通信中断状態のサーバーに対する同じ制限は適用されません。 どちらのサーバーもこの状態にできますが、サーバーが重複アドレスを発行せず、後で適切に再同期できるように、一度に1つだけが存在する必要があります。それまでは、アドレスは保留中の状態です。

状態	サーバーのアクション
POTENTIAL-CONFLICT	自動再統合を保証せず、パートナーとの再統合を試みている状況である可能性があります。サーバーは、2 つのクライアント (動作していない可能性があります) が提供され、同じアドレスを受け入れたことを判断し、この競合を解決しようとします。
RECOVER	安定したストレージにデータがない、または、その安定ストレージをリフレッシュしようとしている PARTNER-DOWN 状態から回復した後に再統合しようとしています。この状態のメイン サーバーは、リースのサービスをすぐに開始しません。このため、この状態でサーバーを再ロードしないでください。
RECOVER-DONE	RECOVER または PARTNER-DOWN 状態から、または通信中断から通常状態に移行できます。
PAUSED	パートナーに、短時間サービスが切れであることを通知できます。その後、パートナーは COMMUNICATIONS-INTERRUPTED 状態に移行し、クライアントのサービスを開始します。

統合中のステート移行

通常の運用中、フェールオーバー パートナーは状態間で移行します。状態遷移のすべてのアクションが完了し、通信が失敗した場合は、次の状態の条件が満たされるまで、現在の状態にとどまります。次の表は、サーバーがさまざまな状態に入ったときにどうなるか、およびサーバーが最初に統合して、後で特定の条件下で互いに再統合する方法を示しています。

表 *11 :* フェールオーバー状態の移行と統合プロセス

統合	結果
NORMAL 状態で、バックアップ サーバーがメイン サーバーに初めて接続する場合	1. 新しく構成されたバックアップ サーバーは、メイン サーバーに接続します。 2. バックアップ・サーバーは新しいパートナーであるため、RECOVER状態になり、メインサーバーにバインド要求メッセージを送信します。 3. メイン サーバーは、リース状態データベースにリースを含むバインド更新メッセージを返します。 4. バックアップ サーバーがこれらのメッセージを確認すると、メイン サーバーは Binding Complete メッセージで応答します。 5. バックアップ サーバーは RECOVER-DONE 状態になります。 6. 両方のサーバーが NORMAL 状態になります。 7. バックアップ サーバーは、プール要求メッセージを送信します。 8. メイン・サーバーは、設定されたバックアップpctに基づいて、バックアップ・サーバーに割り当てるリースに応答します。
通信後 -中断状態	1. サーバーが再起動し、この状態のパートナーと接続すると、戻りサーバーは同じ状態になり、その後すぐに NORMAL 状態になります。 2. パートナーも NORMAL 状態に移行します。
パートナーダウン状態の後	サーバーが復帰して、この状態のパートナーと接続すると、サーバーは、パートナーがこの状態になった時刻とダウンした時刻を比較します。 • サーバーがダウンしたことを検出し、パートナーが次の状態に移行した場合は、次の手順を実行します。 1. 戻りサーバーは RECOVER 状態に移行し、更新要求メッセージをパートナーに送信します。 2. パートナーは、以前に送信できなかったすべてのバインド データを返し、更新完了メッセージをフォローアップします。 3. 戻りサーバーは RECOVER-DONE 状態に移行します。 4. 両方のサーバーが NORMAL 状態になります。

統合	結果
	• 戻りサーバーが、パートナーが PARTNER-DOWN 状態になったときに、まだ動作していたことが検出された場合は、次の手順を実行します。 1. サーバーは潜在的な競合状態になり、パートナーもこの状態になります。 2. メイン サーバーは、バックアップ サーバーに更新要求を送信します。 3. バックアップ サーバーは、メイン サーバーに対するすべての未確認の更新に応答し、更新完了メッセージで終了します。 4. メインサーバーは NORMAL 状態に移行します。 5. バックアップ サーバーは、すべての確認応答されていない更新を要求する更新要求メッセージをメイン サーバーに送信します。 6. メイン サーバーはこれらの更新を送信し、更新完了メッセージで終了します。 7. バックアップ サーバーが NORMAL 状態になります。
サーバーがリース状態データベースを失った後	通常、戻りサーバーはリース状態データベースを保持します。ただし、致命的な障害や意図的な削除が原因で失われることもあります。 1. リース・データベースが欠落しているサーバーが、PARTNER-DOWN 状態または COMMUNICATIONS- INTERRUPTED 状態のパートナーと共に戻ると、サーバーは、そのパートナーが通信したことがあるかどうかを判別します。それがない場合は、データベースを失い、RECOVER 状態に移行し、更新要求メッセージをパートナーに送信します。 2. パートナーは、データベース内のすべてのリースに関するバインドデータで応答し、更新完了メッセージをフォローアップします。 3. 戻りサーバーは、クライアントの最大リードタイム(MCLT)期間(通常は1時間)を待機し、RECOVER-DONE状態に移行します。MCLT の詳細については、 を最大クライアント リードタイムの設定 （80 ページ）参照してください。 4. その後、両方のサーバーが NORMAL 状態になります。

統合	結果
リース状態データベースのバックアップ復元後	戻りサーバーが、そのリース状態データベースをバックアップから復元し、追加のデータを持たないパートナーと再接続する場合、まだ見ていないリースバインドデータのみを要求します。このデータは、期待するデータとは異なる場合があります。 1. この場合、バックアップが発生した時刻に設定されたフェールオーバー回復属性を使用して、戻りサーバーを構成する必要があります。 2. サーバーは RECOVER 状態に移行し、すべてのパートナー データを要求します。サーバーは、バックアップが実行されてRECOVER-DONE状態になったときからMCLT期間(通常は1時間)を待機します。MCLT の詳細は、「最大クライアント リードタイムの設定 （80 ページ）」を参照してください。 3. サーバーが NORMAL 状態に戻ったら、フェールオーバー リカバリ属性を設定解除するか、ゼロに設定する必要があります。 `nrcmd> dhcp set failover-recover=0`
運用サーバーでフェールオーバーが無効になった後	オペレーティング・サーバーでフェイルオーバーが有効になっていたり、無効にされた後に再び使用可能になった場合は、新しく構成されたバックアップ・サーバーを稼働させる際に特別な考慮事項を使用する必要があります。バックアップ サーバーには、リース状態データがなく、フェールオーバー リカバリ属性を現在の時刻から MCLT 間隔 (通常は 1 時間) を引いた値に設定する必要があります。MCLT の詳細は、「最大クライアント リードタイムの設定 （80 ページ）」を参照してください。 1. バックアップ サーバーは、メイン サーバーからすべてのリース状態データを要求することを認識します。このテーブルの「サーバーがリース状態データベースを失った後」で説明されているのとは異なり、バックアップ サーバーはメイン サーバーと通信した記録がないため、このデータを自動的に要求できません。 2. 再接続後、バックアップ・サーバーは RECOVER 状態になり、すべてのメイン・サーバー・リース・データを要求して、RECOVER-DONE 状態になります。 3. 両方のサーバーがNORMAL状態になります。この時点で、バックアップ・サーバーのフェイルオーバー・リカバリ属性を設定解除するか、ゼロに設定する必要があります。 `nrcmd> dhcp set failover-recover=0`

詳細なフェールオーバー属性の設定

設定する必要がある詳細なフェールオーバー プロパティは次のとおりです。

- バックアップ割り当ての境界の設定（バックアップ割り当て境界の設定 （93 ページ）を参照）
- DHCP リースクエリとフェイルオーバー（DHCPLEASEQUERY とフェールオーバー （93 ページ）を参照）

バックアップ割り当て境界の設定

スコープでフェールオーバー バックアップ- バックアップ割り当て境界属性を使用すると、バックアップ サーバーに割り当てるアドレスをより具体的に指定できます。この値として設定された IP アドレスは、バックアップ サーバーにアドレスを割り当てるアドレスの上限です。この境界の下のアドレスのみがバックアップに割り当てられます。この境界の下に使用可能なアドレスがない場合は、その上のアドレスが存在する場合は、バックアップに割り当てられます。実際の割り当てはこのアドレスから下に向かって行われますが、DHCP クライアントの通常の割り当てはスコープ内の最下位アドレスから上に向かって行われます。

スコープにフェールオーバーバックアップ-割り当て-境界を設定する場合は、割り当て先使用可能属性も有効にする必要があります。フェールオーバー-バックアップ- 割り当て - 境界が設定されていないか、ゼロに設定されている場合、使用される境界は、スコープ範囲の最初と最後のアドレスの中間になります。この境界の下に利用可能なアドレスがない場合は、最初に利用可能なアドレスが使用されます。

DHCPLEASEQUERY とフェールオーバー

プライマリサーバーがダウンしたときに DHCP フェールオーバー バックアップ サーバーに送信される DHCPLEASEQUERY メッセージに対応するために、プライマリサーバーは relay-agent-info(82) オプション値をパートナーサーバーに通知する必要があります。これを実現するために、プライマリサーバーは DHCP フェールオーバー更新メッセージを使用します。

フェールオーバー サーバー ペアの保守

このセクションでは、フェールオーバー サーバーペアを維持し、次の管理タスクを実行する方法について説明します。

- フェールオーバーペア名の変更（フェールオーバー ペア名の変更 （94 ページ）を参照）
- フェールオーバー サーバーの再起動（フェールオーバー サーバーの再起動 （94 ページ）を参照）

フェールオーバー ペア名の変更

フェールオーバー ペアの古い名前**セット名** =new-nameを使用して、フェールオーバー ペアの名前を変更します。Web UI では、削除してから新しいオブジェクトを作成する必要があります (新しいオブジェクトが準備ができるまで DHCP サーバーを再ロードせずに削除します)。

（注） フェールオーバー 関係のクラスターの役割が変更された場合 (メインからバックアップ、またはメインへのバックアップ)、そのリレーションシップの既存の状態情報は破棄されます。

フェールオーバー サーバーの再起動

フェールオーバー同期を有効にするには、メイン およびバックアップ フェールオーバー サーバーの両方に最初に接続して再起動する必要があります。

ステップ 1 [DHCP フェールオーバー ペアの一覧表示/追加] ページで、[フェールオーバー ペア] ペインでフェールオーバー ペアを選択します。

ステップ 2 メイン サーバーの [フェールオーバー サーバーの管理] タブで、再起動するサーバーを選択します。

ステップ 3 [サービスの再起動（Restart Service）] アイコンをクリックします。

関連項目

フェールオーバーの確認 （86 ページ）

フェールオーバー設定の回復

Cisco Prime Network レジストラーを最新バージョンにアップグレードすると、アップグレードが失敗した場合に備えて、以前のバージョンに戻すことができます。1 つのパートナーをアップグレードし、正常に動作している状態で NORMAL 状態に回復した後、もう一方のパートナーをアップグレードできます。

アップグレード中に作成されたアーカイブから回復できる場合がありますが、メンテナンス期間中にアップグレードがスケジュールされている場合は、次の作業を行う必要があります。

- systemctl stop nwreglocal 使用して、Cisco Prime Network Registrar を完全に停止させます。
- Cisco プライムネットワーク レジストラー DATADIR(/var/nwreg2/ローカル/データ)をタール アップし、安全な場所に保存します。
- サーバーをアップグレードします。

失敗した場合は、次の手順を実行する必要があります。

- systemctl stop nwreglocal 使用して、Cisco Prime Network Registrar を完全に停止させます。

- Cisco プライムネットワーク レジストラー DATADIR の破損したバージョンを削除します(場所: /var/nwreg2/ローカル/データ)。
- 保存された Cisco プライムネットワーク レジストラー DATADIR tar ファイルを、そのパスから取得したパスに抽出します。
- 既存の DATADIR を検出して使用する Cisco プライムネットワーク レジストラーの元のバージョンをインストールします。

PARTNER-DOWN状態を使用してフェールオーバーパートナーなしでフェールオーバーサーバーを長時間動作する

一方または両方のフェイルオーバー・パートナーが、通信中断状態に移行する可能性があります。この状態の間は、重複するアドレスを発行できません。ただし、サーバーが実行できる処理には制限があるため、長期間にわたってこの状態のサーバーを使用することはお勧めできません。メインサーバーは期限切れのリースを再割り当てできず、バックアップサーバーのプールからアドレスが不足する可能性があります。COMMUNICATIONS-INTERRUPTED 状態は、サーバーが数分から数日の一時的な通信障害を簡単に生き残るために設計されました。クライアントの到着と出発の速度によっては、サーバーがこの状態で短時間だけ効果的に機能する場合があります。その後、サーバーを PARTNER-DOWN 状態に移行して、サーバーが再同期するまでリース機能を完全に引き継ぐようにすることをおお方が良いでしょう。

サーバーが PARTNER-DOWN 状態に移行する方法は 2 つあります。

- User action：管理者は、実際の正確な評価に基づいて、サーバーを PARTNER-DOWN 状態に設定します。フェールオーバー プロトコルがこれを正しく処理します。両方のパートナーを PARTNER-DOWN に設定しないでください。
- Failover safe period expires：サーバーが長時間無人で実行される場合、自動的に PARTNER-DOWN 状態を入力する方法が必要です。

詳細については、フェールオーバー セーフ期間を使用して、サーバーを PARTNER-DOWN 状態に移行する （81 ページ）を参照してください。

（注） フェールオーバー ペアの 1 つのサーバーが長時間サービスを停止した場合、もう一方のサーバーを PARTNER-DOWN 状態にし、フェールオーバー リレーションシップを構成したままにすることを強くお勧めします。

フェールオーバー関係を構成解除する代替方法は、サーバー上で動作を維持する場合とほぼ同じ効果を持ちますが、そのサーバーと戻ってくるフェールオーバーパートナーを、リースに影響を与えない作業フェールオーバーリレーションシップに再統合します。状態データは困難であり、不可能な場合があります。

フェールオーバー ペアの 1 台のサーバーがしばらくダウンした場合は、残りの動作中のサーバーを PARTNER-DOWN 状態にする必要があります。運用サーバーのフェールオーバー 関係を解除しないでください。

復帰するフェールオーバー パートナーの再統合

戻りサーバーが、無傷のリース状態データベースを保持している場合は、そのデータベースはサービスに戻され、運用サーバーとの接続を行う必要があります。

戻りサーバーが致命的な障害を起こして、そのままのリース状態データベースでサービスに戻ることができなかった場合、状況はもう少し複雑になります。この場合、Cisco Prime ネットワーク レジストラーの新規インストールは、通常、戻ってくるサーバー(同じ物理マシンではない場合もあります)に必要です。戻りサーバーは、障害が発生したサーバーと同じ IP アドレスを持ち、新しいCisco Prime ネットワーク レジストラー Cisco PrimeIPインストールは、障害が発生したサーバーと同じ設定にする必要があります。これは通常、運用サーバーと同じです。その後、新しいサーバーがサービスに移行し、既存の運用サーバーとの間に接続します。

（注） どちらの場合も、既存のオペレーションサーバーが実際に稼働しているサーバーがオンラインになった時点で動作することが重要です。運用サーバが何を行ったかを考慮または知らなくても、IP アドレスを配り始めます。

戻りサーバーが最初に起動すると、運用サーバーに接続し、最後に通信した時刻を交換します。

発生する可能性のある状況は 2 つあります。

- (Cisco Prime Network Primeが再インストールされなかった)、そのままのリース状態データベースを持つサーバーがサービスに復帰すると、しばらくサービスが終了したことをパートナーに連絡した後に確認し、RECOVER状態に移行し、そのパートナーはサービスを離れてから何が起きたかについての情報を送信します。この更新が完了すると、両方のサーバーが NORMAL 状態に移行します。
- Cisco Prime Network レジストラー Ciscoが再インストールされたサーバーがこの交換を完了すると、運用サーバーと通信したことがないことが認識され、オペレーションサーバーはサーバーと通信し(または先行サーバー)、新しく復元されたサーバーはリース状態データベースを失ったことに気付きます。RECOVER状態に移行し、すべてのリース状態情報の完全なダウンロードを運用サーバーから要求します。このダウンロードが完了すると(リース状態データベースのサイズとサーバーの負荷に応じて、数分または長くかかる場合があります)、両方のサーバーが NORMAL 状態に移行します。

スタンドアロン DHCP フェールオーバー サーバーの復元（チュートリアル）

ここでは、バックアップサーバーをスタンドアロンモードにしたメインサーバーとバックアップ サーバー間の DHCP フェールオーバー関係を再作成する方法について説明します。この状況はあまり起こらない。

メイン サーバーが数分間を超えてサービスを停止している状況を処理する適切な方法は、バックアップ サーバーを PARTNER-DOWN 状態に設定することです。詳細については、PARTNER-DOWN 状態を使用してフェールオーバー パートナーなしでフェールオーバー サーバーを長時間動作する （95 ページ）を参照してください。

次の手順は、管理者が、メイン サーバーがサービスを提供しなき場合に、バックアップ サーバーをフェールオーバー関係から削除する方法が適切であると誤って考えた状況から回復するために提供されます。繰り返しますが、これは正しい手順ではありません。この間違いから立ち直るのは難しいですが、次の手順が役立ちます。

1. スタンドアロン サーバーは、メイン サーバーの役割を担います。
2. 元のメイン サーバーがバックアップ サーバーになります。
3. パートナーは同期します。
4. サーバーの役割を逆にする意図的に切断されるフェールオーバー 関係。
5. パートナーは、元のフェールオーバー ロールで再同期します。

関連項目

バックグラウンド （97 ページ）

修復手順 （98 ページ）

バックアップ サーバーのフェールオーバー ロールの反転 （98 ページ）

サーバー A の電源をオフにした状態での起動 （99 ページ）

サーバー A を置き換えての起動 （101 ページ）

サーバー A への現在のリース状態の転送 （101 ページ）

バックグラウンド

このセクションの残りの部分では、メイン DHCP フェールオーバー サーバーはサーバー A (クラスタ A という名前のクラスタ オブジェクトを持つ) として識別され、バックアップ サーバーはサーバー B (cluster-B という名前のクラスタ オブジェクトを持つ) として識別されます。サーバー A が管理上または他の方法でシャットダウンされるか、Cisco Prime Network レジストラー サーバー エージェントが停止します。この時点で、サーバー B は通信中断モードに入ります。

システム管理者は、次のいずれかの方法を実行できます。

- **バックアップ サーバー B を通信中断モードで実行し続ける**：バックアップ サーバーを無期限にこのモードで実行するリスクは、バックアップ サーバーが新しいクライアントにサービスを割り当てる利用可能なアドレスの 10% のプールを使い果たす可能性があるというものです。
- **フェールオーバー関係を壊さずにサーバー B をパートナー ダウン モードにする**：フェールオーバーを中断せずに、バックアップ サーバーにアドレス空間のフル コントロールを与える 1 つの重要な注意点は、構成された最大クライアント リード タイム (MCLT) の後までアドレス空間所有権の完全な転送が行われないということです。MCLT は、メイン サーバーに設定された追加の期間で、バックアップ サーバーが検出した期間よりもクライ

アントリースの有効期限が先行する期間を制御します。MCLT は通常 60 分です。MCLT の有効期限が切れるまで、バックアップ サーバーの使用可能なアドレス プールは、割り当てられた予約に制限されます。

- **サーバー B をパートナー ダウン モードにしてフェールオーバー関係を解除する**：この方法では、バックアップ サーバーをスタンドアロン モードにし、管理者がこのシナリオで選択したアプローチになります。決定要因としては、メインサーバーが長時間オフラインになると予想され、オンラインになる新しいデバイスの数が予想を上回ることが考えられます。バックアップサーバーがサービスを提供できるアドレスの割合が低いと、新しいデバイスが停止する可能性があるため、管理者はサーバー B をスタンドアロン モードにします。このアプローチの欠点は、パートナーを元の関係に復元する際に、ネットワークの元の状態を維持するために必要な注意と労力です。

最初の 2 つのアプローチは、3 番目の方法よりも明確な利点があります。ほとんどの場合、MCLT の有効期限が切れるまで、バックアップ サーバーは新しく到着したクライアントをカバーするのに十分なアドレスを持っていると予想されます。3番目のアプローチを追求すると、不必要な管理上の負担とリスクが発生する可能性があります。

修復手順

修復手順は次のとおりです。

1. **バックアップ サーバー B にメイン フェールオーバー サーバーの役割を一時的に割り当てる**：フェールオーバー パートナーの役割を逆にすることで、サーバー A はサーバー B から現在のフェールオーバー状態を学習できます。
2. **サーバー A とサーバー B を元のフェールオーバーの役割に戻す**：目標は、サーバー A が元の状態をメインの DHCP フェールオーバー サーバーとして再取得することです。

前提は次のとおりです。

- 元のメイン サーバー A は非動作であり、Cisco Prime Network レジストラー は停止されます。
- 元のバックアップ サーバー B が動作しています。
- パートナー間のフェールオーバーは管理上無効です。
- 2 つのパートナーのフェールオーバーの役割を完全に取り消さないという決定が下されました。
- ドメイン ネーム システム (DNS) がどちらのフェールオーバー パートナーでも実行されていません。

（注） 例として使用される IP アドレスは、デモンストレーションのみを目的としたものです。

バックアップ サーバーのフェールオーバー ロールの反転

次の手順では、サーバーBを一時的にメインサーバーモードに移行することで、フェールオーバーを復元します。

サーバー B (クラスター B) で次の手順を実行します。

ステップ 1 フェールオーバーが無効になっていることを確認します。サーバー B がメイン、サーバー A がバックアップになるように、フェールオーバー構成を変更します。

```
nrcmd> failover-pair examplepair set failover=false
nrcmd> failover-pair examplepair set main=cluster-B backup=cluster-A
```

ステップ 2 変更を保存して、サーバーをリロードします。

```
nrcmd> save
nrcmd> dhcp reload
```

ステップ 3 フェールオーバーを再度有効にし、サーバーを再度リロードします。

```
nrcmd> failover-pair examplepair set failover=true
nrcmd> dhcp reload
```

サーバー B がメイン フェールオーバー サーバーとなり、パートナーが再び動作可能になる準備が整いました。その間にサーバー A がアドレスを提供し始めないようにするための、これ以上の操作は、現在の状態によって異なります。

サーバー A が次の場合:

- **電源オフ**：サーバー A の電源をオフにした状態での起動 （99 ページ） を参照してください。
- Cisco Prime Network **レジストラー** DHCP **が起動するように設定されていない状態で電源がオンに設定されている場合**は、サーバー A の電源をオンにし、DHCP サーバーを停止した状態での起動 （100 ページ） を参照してください。
- **別のマシンに置き換えられる場合**はサーバー A を置き換えての起動 （101 ページ） を参照してください。

サーバー A の電源をオフにした状態での起動

サーバー A の電源がオフになっている場合は、電源を再びオンにして続行する必要があります。次の手順では、IP アドレスの漏洩を防ぎながら、サーバー A がオンラインになっていることを確認します。

サーバー A (クラスタ A) で次の手順を実行します。

ステップ 1 サーバーの電源を入れる前に、クライアントとの通信を防ぐための手順を実行する必要があります。これを行う最善の方法は、ネットワークケーブルを手動で取り外してから、マシンを起動することです。次の手順を実行するには、ローカル コンソールが必要です。その他の方法としては、サーバーにパケットを転送しないようにリレー エージェントを再構成したり、コンピュータで受信する DHCP トラフィックを防止する (ファイアウォールに DHCP パケット用の一時的なフィルタをインストールするなど) などです。

（注） クライアント トラフィックがサーバーに到達するのを防ぐことができない場合は、DHCP サーバーが停止するまで、クライアントと通信を試みる誤った情報をクライアントに提供する可能性があります。したがって、次の手順で説明するように、サーバーをオンにした後、できるだけ早く DHCP サーバーを停止し、誤った情報を提供する可能性のあるクライアントの数を減らし、リースが重複する可能性があります。

ステップ 2 サーバーの電源をオンにします。

ステップ 3 DHCP サーバーをできるだけ早く停止します。

```
nrcmd> dhcp stop
```

ステップ 4 サーバー A の電源をオンにし、DHCP サーバーを停止した状態での起動 （100 ページ）に移動します。

サーバー A の電源をオンにし、DHCP サーバーを停止した状態での起動

サーバー A の電源がオンになっているが、Cisco Primeネットワーク レジストラー DHCP サーバーが停止しているポイントから開始します。

サーバー A（cluster-A）で、次の手順を実行します。

ステップ 1 サーバー A がバックアップ サーバーになるように、フェールオーバー構成を変更します。

```
nrcmd> failover-pair examplepair set main=cluster-B backup=cluster-A
```

ステップ 2 Cisco プライムネットワーク レジストラーを停止します。

```
systemctl stop nwreglocal
```

ステップ 3 DHCP ログを調べて、DHCP サーバーが動作していないことを確認します。

ステップ 4 サーバー A をネットワークに戻します。ネットワーク ケーブルを再接続するか、リレー エージェントを再構成するか、前のセクションで追加したファイアウォール フィルタを削除します。

ステップ 5 リース状態データベースとイベント ストアを削除します。

```
rm -rf /var/nwreg2/local/data/dhcpeventstore/
rm -rf /var/nwreg2/local/data/dhcp/ndb/
rm -rf /var/nwreg2/local/data/dhcp/ndb6/
```

警告 DHCP データベースを削除する場合は、両方を削除する必要があります：DHCPv4 (.../data/dhcp/ndb) または DHCPv6 (.../data/dhcp/ndb6) リースデータベース。一方のみを削除する（そしてもう一方を残す）ことはサポートされず、予期しない結果が生じる可能性があります。

ステップ 6 Cisco Prime Network レジストラー を起動します。

```
systemctl start nwreglocal
```

ステップ 7 再起動時に DHCP サービスを有効に設定し、DHCP サーバーを起動します。

```
nrcmd> dhcp enable start-on-reboot
nrcmd> dhcp start
```

ステップ 8 サーバー A への現在のリース状態の転送 （101 ページ）に進みます。

サーバー A を置き換えての起動

サーバー A が使用停止され、交換された場合は、Cisco Prime Network レジストラーをインストールし、サーバーBから新しいマシンにフェールオーバー設定をプッシュする必要があります。また、サーバーAに固有の顧客構成を復元する必要があります。これらの手順の後、Cisco プライムネットワーク レジストラーは開始しますが、アドレスは提供しません。

ステップ 1 Server A（クラスタ A）にて、Cisco Prime Network レジストラー をインストールします。

ステップ 2 Cisco ブロードバンド アクセス センターなどの付属ソフトウェアと必要な DHCP 拡張機能を復元して、Cisco Prime Network レジストラーのオペレーティング環境を再構築します。構成をサーバー B にプッシュするまで、構成に対して管理上の変更を行わないでください。

ステップ 3 Server Cisco B Primeネットワーク レジストラー Web UI を使用して、サーバ A に正確なフェールオーバー設定をプッシュします(クラスタ B)。これにより、サーバー A がバックアップ パートナーになります。

ステップ 4 Server A の場合

a) 必要に応じて、CiscoPrimeネットワーク レジストラー設定を、運用環境に必要な設定(管理上の変更を含む) にカスタマイズします。

b) DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

ステップ 5 サーバー A への現在のリース状態の転送 （101 ページ）に進みます。

サーバー A への現在のリース状態の転送

- この時点で、フェールオーバーパートナーシップが再確立し、両方のサーバーが状態を再同期します。
- サーバー A はバックアップ サーバーとして動作可能になります。
- MCLT 期間 (1 時間) の間、操作が一時停止し、両方のパートナーが通常の通信モードでフェールオーバー操作を再開します。

（注） パートナーが同期して通常パートナーを元の役割へ修復 （102 ページ）の通信を報告するまで、に進まないでください。

パートナーを元の役割へ修復

両方のパートナーが完全に同期され、通常の通信を報告することを想定しています。フェールオーバーパートナーが元のロールを引き受けられるようにするには、次の手順を実行します。

ステップ 1 Server A（クラスタ A）では、DHCP サーバーを停止します。

```
nrcmd> dhcp stop
```

ステップ 2 Server B（クラスタ B）では、DHCP サーバーを停止します。

```
nrcmd> dhcp stop
```

ステップ 3 Server A の場合

a) フェイルオーバーを無効にしてから、サーバー A をメインサーバー、サーバー B をバックアップにします。

```
nrcmd> failover-pair examplepair set failover=false
nrcmd> failover-pair examplepair set main=cluster-A backup=cluster-B
```

b) 変更を保存し、DHCP をリロードします。

```
nrcmd> save
nrcmd> dhcp reload
```

c) 構成が適切で、現在実行中であることを確認します。この時点で、サーバー A は、アドレス プールの 100% を持つ唯一の運用 DHCP サーバーです。

d) フェールオーバーを再度有効にします。

```
nrcmd> failover-pair examplepair set failover=true
```

e) DHCP をリロードし、設定変更を再確認します。

```
nrcmd> dhcp reload
```

サーバー A は、サーバー B が動作可能になるのを待つフェールオーバー メイン サーバーになりました。

ステップ 4 Server B: の場合

a) サーバー A をメイン サーバー、サーバー B をバックアップにし、フェールオーバーを有効にします。

```
nrcmd> failover-pair examplepair set main=cluster-Abackup=cluster-B
nrcmd> failover-pair examplepair set failover=true
```

b) 新しい設定を保存しますが、サーバーをリロードしないでください。

```
nrcmd> save
```

c) サーバー B で DHCP サーバーを再起動します。

```
nrcmd> dhcp reload
```

この時点で、フェールオーバー パートナーシップは元の役割で自分自身を再確立し、両方のサーバーが状態を再同期し、サーバー B がバックアップ サーバーとして動作します。この操作は、1 時間の MCLT 期間の間一時停止し、両方のパートナーが通常の通信モードでフェールオーバー操作を再開します。

ステップ 5 Server A および Server B の場合

a) 両方のパートナーが通常のフェールオーバー状態にあるかどうかを検証します。

```
nrcmd> dhcp getRelatedservers
```

b) レポートを実行し、結果が両方のパートナーで一致することを確認し、パートナー間の実行時間の差を少しずらします。

フェールオーバー サーバー ロールの変更

注意 フェールオーバー サーバーの役割を変更する場合は注意が必要です。DHCPv4 スコープまたは DHCPv6 プレフィックスのすべてのアドレス状態は、そのスコープまたはプレフィックスを持たない状態で再ロードされた場合、サーバーから失われる点に注意してください。

関連項目

スタンドアロン サーバーをメインとして使用したフェールオーバーの確立 （103 ページ）
ストレージに欠陥のあるサーバーの交換 （104 ページ）
バックアップ サーバーの削除とフェールオーバー操作の停止 （105 ページ）
既存のバックアップ サーバーへのメイン サーバーの追加 （105 ページ）
複数インターフェイス ホストでのフェールオーバーの設定 （105 ページ）

スタンドアロンサーバーをメインとして使用したフェールオーバーの確立

既存のインストールを更新し、提供する DHCP サービスの可用性を向上させることができます。この手順は、スタンドアロンサーバーがフェールオーバーに参加したことがない場合にのみ使用できます。

ステップ 1 バックアップ サーバーとなるマシンに Cisco Prime Network レジストラーをインストールします。バックアップ サーバーの IP アドレスを記録します。

ステップ 2 クラスタを設定します。スタンドアロンサーバーでフェールオーバーを有効にし、メインサーバーとして構成し、最近バックアップとしてインストールします。

クラスタをコンフィグレーションするには、cluster name create address | ipv6-address scp-port=value admin=value password=value を使用します。次に例を示します。

```
nrcmd> cluster backup create 10.65.201.23 scp-port=1234 admin=admin password=changeme
```

ステップ 3 メインサーバーをリロードします。PARTNER-DOWN状態にする必要があります。バックアップサーバーがまだ構成されていないため、バックアップ サーバーを見つけることができません。この時点で、メインサーバーの操作に変更はありません。

ステップ 4 構成を同期するには、フェールオーバー同期を使用して、メインからバックアップへの正確な同期を実行します。

ステップ 5 ブロードキャスト パケットをメイン サーバーおよびバックアップ サーバーに転送するように、すべての動作中の BOOTP リレーを再構成します。

ステップ 6 バックアップ サーバーをリロードします。

次のタスク

この手順を完了すると、次の状態に入ります。

1. バックアップ・サーバーはメインサーバーを検出し、RECOVER状態に移行します。
2. バックアップ・サーバーは、メイン・サーバーのリース・データを使用して安定したストレージを更新し、完了するとRECOVER-DONE状態に移行します。
3. メイン サーバーが NORMAL 状態に移行します。
4. バックアップ サーバーが NORMAL 状態に移行します。
5. バックアップサーバーは、アドレスのプールを取得するためのプール要求を送信します。
6. これらのアドレスを割り当てた後、メインサーバーはバックアップの割合に基づいてバックアップに IP アドレスを割り当てます。

ストレージに欠陥のあるサーバーの交換

フェールオーバー サーバーが安定した記憶域 (ハード ディスク) を失った場合、サーバーを交換して、パートナーから状態情報を回復させることができます。

ステップ 1 安定したストレージを失ったサーバーを特定します。

ステップ 2 CLI のfailover-pair名前 setPartnerDown[date]を使用して、パートナーがダウンしていることを他のサーバーに伝えます。時刻を指定しない場合は、現在の時刻が使用されます。

ステップ 3 サーバーが再び動作状態になったら、Cisco Primeネットワーク レジストラーを再インストールします。

ステップ 4 フェールオーバー同期を使用して、パートナー構成からサーバー構成を同期します。ただし、以前のバックアップまたはパートナー システムからリース データベースを回復しないでください。

ステップ 5 交換用のサーバーをリロードします。

次のタスク

この手順を完了すると、次の状態に入ります。

1. 回復されたサーバーは RECOVER 状態に移行します。
2. パートナーは、すべてのデータを送信します。

3. サーバーは、最大クライアントリードタイム(およびフェールオーバー・リカバリに設定された任意の時間)に達すると、RECOVER-DONE状態に移行します。
4. そのパートナーは NORMAL 状態に移動します。
5. 回復されたサーバーは NORMAL 状態に移行します。アドレスを要求できますが、パートナーが以前に割り当てたすべてのアドレスをすでに送信しているため、新しいアドレスを割り当てることは少なくなります。

バックアップ サーバーの削除とフェールオーバー操作の停止

バックアップ サーバーを削除し、すべてのフェールオーバー操作を停止する必要がある場合があります。

ステップ 1 バックアップ サーバーで、メイン サーバーへのバックアップとして指定されたすべてのスコープまたはプレフィックスを削除します。

ステップ 2 メイン サーバーで、バックアップ サーバーのメインだったスコープまたはプレフィックスからフェールオーバー機能を削除するか、構成されている場合はサーバー全体でフェールオーバーを無効にします。

ステップ 3 両方のサーバーを再ロードします。

既存のバックアップ サーバーへのメイン サーバーの追加

メイン サーバーには既存のバックアップ サーバーを使用できます。

ステップ 1 フェールオーバー同期を使用して、バックアップ サーバー上のメイン サーバー スコープ、ポリシー、およびその他の構成を同期します。

ステップ 2 フェールオーバーを有効にしてバックアップ サーバーをポイントするように、メイン サーバーを構成します。

ステップ 3 新しいメイン サーバーを指す新しいスコープのフェールオーバーを有効にするようにバックアップ サーバーを構成します。

ステップ 4 両方のサーバーを再ロードします。Cisco プライムネットワーク レジストラーは、でスタンドアロン サーバーをメインとして使用したフェールオーバーの確立 （103 ページ）説明されている手順と同じ手順を実行します。

複数インターフェイス ホストでのフェールオーバーの設定

複数のインターフェイスを持つサーバー ホストでフェールオーバーを使用する場合は、ローカル サーバー名またはアドレスを明示的に構成する必要があります。これには追加のコマンドが必要です。たとえば、サーバー A とサーバー B の 2 つのインターフェイスを持つホストがあり、サーバー A をメイン フェールオーバー サーバーにする場合、バックアップ サーバー名(外部サーバー B) を設定する前に、サーバー A をフェールオーバー メイン サーバーとして定

義する必要があります。これを行わない場合、フェールオーバーが正しく初期化されず、間違ったインターフェイスを使用しようとする可能性があります。

フェールオーバーサーバー-メイン サーバーおよびフェールオーバーバックアップ サーバーの DHCP サーバー プロパティを設定する :

1 つのホストに複数のインターフェースがある場合は、1 つのアドレスまたはレコードのみを指すホスト名を指定する必要があります。ラウンドロビンをサポートするためにサーバーをセットアップすることはできません。

フェールオーバーパートナーの別ネットワークへの移動

フェールオーバーパートナーが動作している可能性があるネットワークの番号を変更したり、フェールオーバー パートナーを別のネットワーク セグメントに移動したりする必要が生じる場合があります。このような場合、サーバーの再起動が必要な構成変更が必要なため、サービスの停止が短時間で発生します。また、新しいサーバーアドレスにトラフィックを転送するために、リレー エージェントを更新する必要があります。

（注） 次の手順では、フェールオーバー ペア オブジェクトで明示的なアドレスが構成されていないと仮定します。メインおよびバックアップ クラスタ オブジェクトから通常継承されたアドレスを上書きするように明示的なアドレスが構成されている場合は、フェールオーバー ペア オブジェクトのアドレスを手動で更新する必要があります (手順 1 と 2)。

両方のフェールオーバーパートナーのアドレスを変更する場合は、次の手順を使用することをお勧めします。

ステップ 1 メインで、**クラスタ**名set ipaddr=アドレスまたは**クラスタ**名 set ip6address=addressコマンドを使用して、バックアップの新しいアドレスを使用するようにバックアップ クラスタ オブジェクトを再構成します。サーバーを再ロードしないでください。

（注） メインのクラスタ オブジェクトのアドレスを変更することはできません。これは、新しいサーバーが移動して起動すると自動的に変更されます。

ステップ 2 バックアップで、メインの新しいアドレスを使用するようにメインクラスタ オブジェクトを再構成します。サーバーはリロードしません。

ステップ 3 バックアップを停止する前に、DHCP サーバーの起動を無効**にします**(dhcp disable on rebootコマンドを使用します)。これにより、サーバーをブートし、DHCP を自動的に実行することが可能になります。

ステップ 4 バックアップ サーバーでCisco プライム ネットワーク レジストラー Ciscoプライム IPを停止するか、シャットダウンします。DHCP サーバーが起動されないので、移動して再起動できます。

ステップ 5 バックアップ サーバーが長時間ダウンする場合 (物理的に移動する必要がある場合など)、メインをパートナーダウン状態に移行する必要があります(フェールオーバー**ペア**名 setPartnerDownコマンドを使用)。

ステップ 6 メイン サーバーをシャットダウンして移動します。この期間中、クライアントはリースを取得または更新できません。

ステップ 7 新しいアドレスでメインサーバーを起動します。メインのローカル クラスタ オブジェクトのアドレスが新しいアドレスであること、およびバックアップ クラスタ オブジェクトのアドレスが有効であることを検証します。また、DHCP トラフィックがリレーから到着していることを確認し、中継エージェントを構成し直して、新しいメイン サーバーアドレスとバックアップ サーバアドレスにトラフィックを適切に転送するようにします。

ステップ 8 バックアップ システムを新しいアドレスで起動します (手順 4 で開始していなかった場合)。バックアップのローカル クラスタ オブジェクトのアドレスが新しいアドレスであること、およびメイン クラスタ オブジェクトのアドレスが有効であることを検証します。

ステップ 9 バックアップで、起動時の再起動を有効にし、dhcp enable-on-rebootコマンドとdhcp startコマンドを使用してサーバーを起動します。

ステップ 10 フェールオーバー通信が動作していることを検証し、通常の状態に戻ります(dhcp getRelatedServersコマンドを使用して、いずれかまたは両方のクラスターのフェールオーバーステータスを表示します)。通信が速やかに再開されない場合は、バックアップで DHCP サーバーを停止し、クラスタ上のアドレスとフェールオーバー ペア オブジェクトの構成変更が正しく適用されていることを確認します。

ステップ 11 地域で、メインおよびバックアップ クラスタ オブジェクトを更新して、新しいアドレスを使用します。または、メイン クラスタとバックアップ クラスタの両方でlicense registerコマンドを使用して、リージョンを更新することもできます。

フェールオーバーのトラブルシューティング

このセクションでは、フェールオーバー構成の誤りを回避し、フェールオーバー操作を監視し、ネットワークの問題を検出して処理する方法について説明します。

関連項目

フェールオーバー操作のモニターリング （107 ページ）

ネットワーク エラーの検出と処理 （108 ページ）

フェールオーバーに関連する問題のトラブルシューティング時に避けるべき事項（109ページ）

フェールオーバー操作のモニターリング

両方のパートナー サーバーの DHCP サーバー ログ ファイルを調べて、フェールオーバー構成を確認できます。

いくつかの重要なログとデバッグの設定を行って、フェールオーバーのトラブルシューティングを行うことができます。DHCPログ設定をフェールオーバーの詳細に設定し、ログに記録されたフェイルオーバーメッセージの数と詳細を追跡します。以前のメッセージが上書きされないようにするには、リストの最後にフェールオーバーの詳細属性を追加します。非フェール

オーバー競合属性を使用して、ログ記録サーバーのフェールオーバー競合を禁止するか、または通常のサーバー フェールオーバー アクティビティのログ記録を禁止する非フェールオーバー アクティビティ属性を使用します。次に、サーバーを再ロードします。

また、[DHCP サーバーの管理] ページまたは [DHCP フェールオーバー ペアの一覧/追加] ページの [**関連サーバー**]dhcp getRelatedServersアイコンをクリックするか、CLI で使用することで、設定ミスをより簡単に切り分けることができます。

ネットワーク エラーの検出と処理

次の表に、フェールオーバーの問題に対する症状、原因、および解決策を示します。

表 *12 :* 障害の検出と処理

症状	原因	ソリューション
新しいクライアントはアドレスを取得できません	バックアップ・サーバーが、アドレスが少なすぎる、COMMUNICATIONS-INTERRUPTED 状態です。	メイン サーバーのバックアップの割合を増やします。
スコープの不一致に関するエラー メッセージ	パートナー間でスコープ構成が一致しません。	サーバーを再構成します。
パートナーとの通信の失敗に関するメッセージをログに記録する	サーバーはパートナーと通信できません。	サーバーのステータスを確認します。
メイン サーバーに障害が発生しました。一部のクライアントは、リースを更新または再バインドできません。バックアップ サーバーがアップ状態で、クライアント要求を処理している場合でも、リースは期限切れになります。	一部の BOOTP リレー エージェント (ip-helper) は両方のサーバーをポイントするように構成されていません。BOOTP リレーの設定（110 ページ）を参照してください。	• BOOTP リレーを、メイン サーバーとバックアップ サーバーの両方を指す設定に戻します。 • ファイアードリルテストを実行する - メインサーバーを1日ほど停止し、ユーザーコミュニティがリースを取得して更新できるかどうかを確認します。
SNMP トラップ: 他のサーバーが応答しません	サーバーはパートナーと通信できません。	サーバーのステータスを確認します。
SNMP トラップ: DHCP フェールオーバー構成の不一致	パートナー間でのスコープ構成の不一致	サーバーを再設定します。

症状	原因	ソリューション
ユーザーが期待どおりにサービスやシステムを使用できないという苦情	パートナー間のポリシーとクライアント クラスの不一致	同一のポリシーを持つパートナーを再構成します。現在、パートナーに直接クライアントを登録している場合は、クライアント登録にLDAP を使用する可能性があります。

フェールオーバーに関連する問題のトラブルシューティング時に避けるべき事項

フェールオーバーを使用する場合、問題のトラブルシューティング時に行わない点がいくつかあります。

- フェールオーバー構成を削除しています。残りのサーバーを PARTNER-DOWN 状態に設定する方がはるかに良いです。リースの再利用に長い待ち時間が必要になる場合もありますが、フェールオーバーを設定して PARTNER-DOWN で動作する方がはるかに安全です。
- DHCP リース データベース (../data/dhcp/ndb および ./data/dhcp/ndb6) を一方のフェールオーバー パートナーから他方のフェールオーバー パートナーにコピーしないでください。フェールオーバーパートナーからリースデータを回復する方法については、『Cisco プライムネットワーク レジストラー 11.0 管理ガイド』の「フェールオーバーサーバーからの DHCP データの復元」のセクションを参照してください。これが行われた場合は、データベースをコピーした後にサーバーデュードを削除するために leaseadmin ツールを使用しなければなりません（leaseadmin ツールの詳細についてはサーバー間でのリースの移動（261 ページ）を参照してください）。リース データベースがコピーされるたびに、サーバーデュードをコピーから削除する必要があります。Cisco Prime Network Registrar 10.0 以降、新しいデータベース（または、server-duid が削除されたデータベース）は、ローカルのクラスタ UUID を使用するため、すべてのデータベースに server-duid が格納されるわけではありません。

注 server-duid を削除しないと、2 台のサーバーで同じ server-id を使用することができるため、DHCPv6 は意図したとおりに動作しません。これは、リージョンのリース履歴データに重大な影響を与える可能性があります。

フェールオーバーでの BOOTP クライアントのサポート

静的と動的の 2 種類の BOOTP クライアントをサポートするようにスコープを構成できます。

関連項目

静的 BOOTP （110 ページ）
動的 BOOTP （110 ページ）
BOOTP リレーの設定 （110 ページ）

静的 BOOTP

DHCP 予約を使用して、静的 BOOTP クライアントをサポートできます。フェールオーバーを有効にする場合は、メイン サーバーとバックアップ サーバーの両方を同一の予約で構成してください。

動的 BOOTP

スコープで動的 bootp 属性を有効にすることで、動的 BOOTPクライアントを有効にすることができます。ただし、フェールオーバーを使用する場合、BOOTP クライアントは無期限の永続的なアドレスとリースを取得するため、このようなスコープでのアドレスの使用に関する追加の制限があります。

スコープの動的ブート オプションが有効になっていないサーバーが PARTNER-DOWN 状態になると、そのスコープから使用可能な(割り当てられていない)アドレスを割り当てることができます。ただし、動的ブートオプションを設定すると、各パートナーは独自のアドレスのみを割り当てることができます。したがって、dynamic-bootpオプションを有効にするスコープでは、フェールオーバーをサポートするためにより多くのアドレスが必要になります。

動的ブートを使用する場合:

- 動的 BOOTP クライアントを単一のスコープに分離します。スコープのdhcp属性を無効にして、DHCP クライアントがそのスコープを使用できないようにします。
- 動的bootp-backup-pctフェールオーバー ペア属性を設定して、このスコープのバックアップ サーバーに対して、通常のバックアップの割合よりも 50% も高いアドレスを割り当てます。

BOOTP リレーの設定

Cisco Prime Network レジストラーフェールオーバー プロトコルは、サーバーにローカルに接続されていないDHCPクライアントをサポートするルータ機能であるBOOTPリレー(IPヘルパーとも呼ばれます)で動作します。

BOOTP リレーを使用する場合は、実装がメイン サーバーとバックアップ サーバーの両方を指していることを確認します。これらのパケットが失敗し、メイン サーバーに障害が発生した場合、クライアントはサービスを提供しません。2 つの異なるサーバーにブロードキャスト パケットを転送するようにBOOTP リレーを構成できない場合は、メイン サーバーとバックアップ サーバーの両方を含む可能性がある LAN セグメントのサブネット ローカル ブロードキャスト アドレスにパケットを転送するようにルーターを構成します。次に、メイン サーバーとバックアップ サーバーの両方が同じ LAN セグメント上にあることを確認します。

BOOTP バックアップの割合

動的BOOTPを有効にするスコープの場合、フェールオーバー ペアのbackup-pct属性ではなく、動的ブート-バックアップ pct属性を使用します。動的 bootp-backup-pctは、BOOTP クライアントで使用するためにメイン・サーバーがバックアップ・サーバーに送信する必要がある使用可能なアドレスのパーセンテージです。

DYNAMIC-bootp-backup-pctは、スコープでBOOTPを有効にした場合、PARTNER-DOWN状態であっても、サーバーが他のサーバーで使用可能なアドレスにリースを付与しないため、バックアップ PCt属性とは異なります。Cisco Prime Network レジストラーは、パートナーが動的 BOOTP を使用してリースを提供する可能性があるため、リースを許可しません。

（注） メイン・サーバー上で動的 BOOTP バックアップ率を定義する必要があります。バックアップ サーバーで定義した場合、Cisco Prime Network レジストラーは、これを無視します(スクリプトを使用した設定の複製を有効にするため)。これを定義しない場合、Cisco Prime Network レジストラーはフェールオーバー ペアまたはスコープにデフォルトのバックアップPCTを使用します。

フェールオーバー プロトコルの使用中に動的 BOOTP を正しくサポートするには、BOOTP をサポートするすべての LAN セグメントで次の手順を実行します。

- 動的ブート・ブート用に 1 つのスコープを作成する
- ブートと動的ブートを有効にする
- そのスコープの DHCP を無効にする

DHCP リレー ヘルス チェック

フェイルオーバーを使用する場合、次の 3 つの異なる通信パスがあります。

- フェールオーバー パートナー間 (IPv4 または IPv6 経由)
- リレー エージェントとメイン フェールオーバー パートナーの間 (IPv4 および IPv6 の場合)
- リレー エージェントとバックアップ フェールオーバー パートナーの間 (IPv4 および IPv6 の場合)

これらのパスの1つ以上が壊れることがあります。たとえば、ルーティングの誤った設定やリンクの障害により、リレーエージェントとメインフェールオーバーパートナー間のトラフィック フローを防止できます。これにより、バックアップ フェールオーバー パートナーがこれらのパケットを受信した場合でも、一部のクライアントがオンラインにならないようにします(フェールオーバーがアップすると、通常はクライアントの要求に応答するため)。DHCP サーバーは、リレー エージェントを監視し、リレー エージェントがダウンしていると検出されたときに通常はフェールオーバー NORMAL 状態でサービスを提供しないクライアントに対して応答を有効にするように構成できます。

DHCP リレーの状態チェックを構成するには、[DHCP フェールオーバーペアの一覧/追加]ページの[リレーヘルスチェック]セクションで属性を設定します。詳細については、フェールオーバー ペアの追加 （68 ページ）を参照してください。

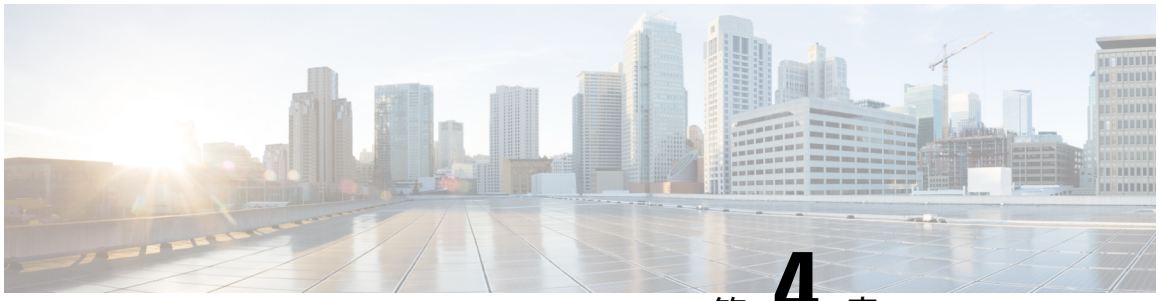
Cisco Prime Network Registrar 11.0 において、IPv4 正常性チェックは、サーバーで使用される dhcp-server-identifier がサーバーのインターフェイスアドレスであり、高速コミットが許可されていない場合にのみ正しく動作します。したがって、giaddr-as-server-id が有効になっているポリシー、明示的な dhcp-server-identifier オプションが指定されているポリシー、または allow-rapid-commit が有効になっているポリシーは、IPv4 に対して自動的に無効になります。Cisco Prime Network Registrar 11.0.1 以降、この機能は拡張され、giaddr-as-server-id がポリシーで有効になっている場合でも IPv4 ヘルスチェックを有効にするようになりました。

IPv6 正常性チェックは、高速コミットが許可されていない場合にのみ正しく動作します。したがって、いずれかのポリシーで allow-rapid-commit が有効になっている場合、IPv6 の正常性チェックは自動的に無効になります。

ただし、サーバーのポリシーチェックでは、クライアントエントリを介して提供されるポリシーはチェックされません。したがって、クライアントポリシーで giaddr-as-server-id、明示的な dhcp-server-identifier オプション、または Rapid-commit が設定されている場合は、リレー正常性チェックを有効にしてください。

CLI コマンド

フェールオーバーが使用されており、ヘルスチェック機能が有効になっている場合は、 dhcp getRelayState [all] [full] コマンドを使用できます。これにより、フェールオーバー パートナーと各リレーエージェント間の通信の状態が報告されます。「すべて」を指定しない場合、フェールオーバー パートナーとの通信に問題があると思われるリレー (つまり、中断状態の中継) のみが報告されます。"full" を指定すると、オブジェクトはテーブルではなく表示されます。



第 4 章

アドレス空間の管理

アドレスブロックは、ネットワークを介して使用されるアドレスの組織構造を提供します。アドレス ブロックは、静的アドレスまたはリース割り当て用に DHCP サーバーに割り当てられた動的アドレスで構成できます。アドレス ブロックは、任意の数の子アドレス ブロックを持つことができます。アドレスブロック管理者は、これらのオブジェクトを担当します。この管理者は、親および子アドレスブロックまたはサブネットを作成できます。静的サブネットは、さらに 1 つ以上の IP アドレス範囲に分割できます。ただし、動的に追加されたサブネットは、管理者が変更または削除できない独自のサブネットを作成します。

（注） IPv6 アドレス管理については、IPv6 アドレス空間の表示 （128 ページ）を参照してください。

アドレス ブロック管理者ロール

アドレスブロック管理者ロールは、特定のサブネットまたは静的アドレス割り当てよりも高いレベルでアドレス空間を管理します。これは、システムにアドレスブロックを配る権限が高くなる可能性が高いため、実際には中間マネージャーの役割です。

関連項目

必要なアクセス許可 （114 ページ）

役割機能 （114 ページ）

必要なアクセス許可

アドレス管理者が使用できる機能を実行するには、次の手順を実行する必要があります。

- Regional cluster：割り当てられた地域追加管理者ロール。この役割は、おそらく、さらなるサブネット使用率、リース履歴、および dhcp 管理サブロールのリースによって妨げられません。
- Local cluster：割り当てられた追加ブロック管理ロール。

役割機能

これらの機能は、アドレス ブロック管理者が以下のサイトで使用できます。

- Regional cluster：
  - アドレスの集約。たとえば、10.0.0.0/16アドレスブロックが地域クラスタに存在し、ローカル クラスタ管理者が 10.1.1.0/24 アドレス ブロックを作成すると、ローカル アドレス ブロック (レプリケーションを通じて) は、その親の下にある地域クラスタにロールアップされます。これにより、ローカルクラスタの構成に影響を与えることなく、地域クラスタのアドレス空間を統一されたビューで表示できます。
  - アドレスの委任。管理者は、アドレス空間をローカルクラスタに委任できるため、委任されたオブジェクトの権限を放棄できます。
  - DHCP使用状況レポート。地域クラスタは、リージョン、プロトコルサーバー、およびネットワーク ハードウェアのセット間での DHCP 使用率レポートをサポートします。中央の設定管理者は、ローカルクラスタに対して、仮想プライベートネットワーク（VPN）によるDHCP使用率をポーリングできます。定義されている場合は、時間範囲と、所有者、地域、アドレスの種類、アドレスブロック、サブネット、またはすべてを含む条件を指定できます。DHCP使用率の照会の詳細については、使用率履歴データの照会 （132 ページ）を参照してください。
  - リース履歴レポート。これにより、複数のDHCPサーバーのリース履歴に関する単一の視点が提供されます。管理者は、ローカルクラスターで履歴データを照会して、履歴レポートの範囲を制限できます。リース履歴は、VPN(定義されている場合)、IPアドレス、MAC アドレス、IP アドレス範囲、またはすべてのいずれかを含む時間範囲と基準によって照会できます。これは、住所トレーサビリティに関する政府および他の機関の義務を満たす重要な機能です。リース履歴の照会の詳細については、をリースの照会 （265 ページ）参照してください。
  - ポーリング構成。管理者は、レプリケーション、IP 履歴、および DHCP 使用率に関するローカル クラスタ ポーリングの間隔と間隔を制御できます。また、リース履歴と DHCP 使用率のトリミング期間と圧縮間隔を CCM サーバー レベルで設定することもできます。(章「中央構成の管理」の章をCiscoプライムネットワーク レジストラー 11.0 管理ガイド参照してください。
  - DHCP とアドレス データの整合性を確認します。
- Local cluster：
  - アドレス ブロック、サブネット、およびアドレスの種類を管理します。
  - DHCP およびアドレス データの一貫性を確認します。

アドレス ブロックとサブネット

アドレスブロックは、権限に委任できる2つのアドレス空間に基づくIPアドレスの集合です。たとえば、192.168.0.0/16 アドレス ブロック(RFC 1918 プライベート アドレス空間の一部)には、216 (または 65536) アドレスが含まれています。アドレス ブロックは、さらに子アドレス ブロックとサブネットに分割できます。たとえば、192.168.0.0/16 アドレス ブロックをさらに 4 つの子アドレス ブロック (192.168.0.0/18、192.168.64.0/18、192.168.128.128/18、および 192.168.192/18) に委任できます。

（注） DHCP サーバーは、アドレス ブロックを使用してオンデマンド アドレス プールのサブネット割り当てを管理します（サブネットの割り当ての設定 （57 ページ）を参照）。動的アドレス プールに使用するアドレス ブロックは、CLI の dhcp-address-block コマンドを使用して作成する必要があります。Web UI の統合アドレス ビューには、これらの動的アドレス ブロックも表示されますが、DHCP サーバーに完全に委任されているため、これらの動的アドレスブロックには編集リンクは提供されません。サブネット割り当てのために、さらに細分化しないでください。DHCP サーバーは、サブネット要求を受信すると、これらのアドレスブロックを自動的に処理します。これらのアドレスプールは、Dによって示されます（「委任済み」の場合）。

サブネットはアドレス空間のリーフ ノードであり、さらに細分化することはできません。192.168.50.0/24サブネットを作成すると、その同じ名前でアドレスブロックを作成でき、サブネットはアドレスブロックの子になります。ただし、192.168.50.0/24サブネットをさらに細分化または委任することはできません。

サブネットには、1つ以上のアドレス範囲を定義できます。アドレスブロックにはアドレス範囲を設定できません。Web UI を使用してサブネットのアドレス範囲を作成すると、そのアドレス範囲は静的範囲になり、DHCPを使用して動的に割り当てることはできません。ただし、Web UI には、サブネットの DHCP スコープによって定義された動的範囲が表示されます。範囲を表示する場合は、アドレス空間に静的アドレスを割り当てる際とスコープの動的アドレスを割り当てる際に、重複が発生する可能性がある場所を示します。

アドレス空間ビューには、アドレスブロックとサブネットの階層、およびそれらの親子関係が表示されます。階層は、各サブネットのアドレス範囲のレベルに下がりません。これらは、サブネットにアクセスするときに表示されます。

関連項目

アドレス ブロック、サブネット、アドレス タイプの表示 （128 ページ）
アドレス ブロックの追加時期の把握 （117 ページ）
アドレス ブロックの追加 （117 ページ）
アドレス ブロックの委任 （120 ページ）
ローカル DHCP サーバーおよびルータへのサブネットのプッシュ （124 ページ）

サブネットの割り当てと DHCP アドレス ブロック

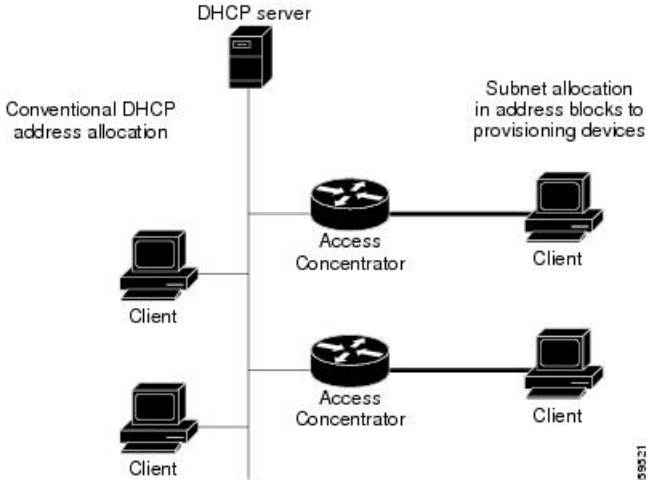
Cisco Prime Network レジストラーは、アドレス プロビジョニングと VPN のネットワーク インフラストラクチャとしてオンデマンドアドレス プールの作成をサポートします。従来、DHCP サーバーは個々のホスト デバイスとの対話に制限されています。サブネット割り当てを通じて、サーバーは VPN ルーターやその他のプロビジョニング デバイスと対話して、IP サブネット全体をプロビジョニングできます。Cisco Prime Network レジストラー機能により、Cisco IOS リレー エージェントで現在サポートされているオンデマンド アドレス プール機能が強化されます。

Cisco プライムネットワーク レジストラーは、明示的にプロビジョニングされたサブネットをサポートします。サーバーがプールまたはリースを割り当てるには、DHCP サーバーのアドレス空間とサブネット割り当てのポリシーを明示的に構成する必要があります。それによって、サブネットを管理し、クライアントデバイスに委任するプールマネージャとしてサーバーを設定できます。

DHCP サブネット割り当てを管理するには、Cisco Prime Network レジストラーの DHCP サーバー アドレス ブロック オブジェクトを使用します。DHCP アドレス ブロックは、割り当てのために DHCP サーバーに委任される連続した IP アドレスの範囲です。サーバーは、これらのアドレスをプールに分割して、アドレスまたは他のサーバーまたはデバイスが割り当てることができるようにします。DHCP アドレス ブロックは、サブネットの親です。これらの DHCP アドレス ブロックは、Cisco Primeネットワーク レジストラー Web UI を使用して作成できるアドレス ブロックとは異なります。DHCP アドレス ブロックには、静的アドレス範囲やリース予約を含めることはできません。

次の図は、DHCP サーバーが個々のクライアントにサービスを提供するだけでなく、コンセントレータまたはその他のプロビジョニング デバイスにアクセスするためにサブネット全体を割り当てるサンプル環境を示しています。従来のクライアント/サーバー関係は図の左側に示され、アクセス コンセントレータへのサブネット割り当ては図の右側に示されています。たとえば、ダイヤルアップの顧客は、DHCP サーバーが存在する管理ネットワーク セグメントに接続する 2 つの ISP ゲートウェイ (ルーター) でサービス プロバイダ ネットワークに接続します。ゲートウェイは、DHCP サーバーから要求されたサブネットに基づいて、接続されているクライアントにアドレスをプロビジョニングします。

図 *10 : DHCP* サブネット割り当ての構成例



アドレス ブロックの追加時期の把握

このユース ケースでは、共有管理ネットワークのネットワークに新しいアドレス ブロックを追加することに関連する一連のユーザー アクションについて説明します。これらの前提条件は、次の前提となります。

1. IPアドレス使用率の概要レポートから、アドレスブロック管理者は、会社の最上位のアドレス ブロックが 90% の使用率マークに近づいていることを指摘します。
2. アドレスブロック管理者は、ARIN（または他の番号指定機関）からより多くのアドレススペースの要求を送信し、要求が許可されます。

アドレス空間が使用可能になったら、地域アドレス管理者は次の手順を実行します。

1. 新しいブロックを中央アドレス ブロック マップに追加し、使用率レポートのレビューに基づいて、ローカル クラスタが使用するアドレス ブロックを作成および委任します。アドレス ブロックを委任するアクションにより、そのアドレス ブロックはローカル クラスタにプッシュされます。
2. フェールオーバー同期を使用して、構成タスクを簡略化するために、必要に応じて新しいアドレス空間をネットワーク要素に割り当てます。
   - サブネットをフェールオーバー ペアに割り当てます (サブネットまたはフェールオーバー ペアからサブネットのスコープ テンプレートを取得します)。
   - 空きサブネットを検索します (正しいタイプのアドレス ブロックを検索します)。
   - 空のサブネットをアドレスの宛先 (DHCP サーバーまたはその他の宛先) に割り当てます。

アドレス ブロックの追加

ネットワークを構成したら、DHCPv4 アドレス ブロックを追加できます。

ローカルの高度な Web UI と地域の高度な Web UI

CCM アドレス ブロックを表示するには、Design メニューで、DHCPv4 サブメニューの下から Address Blocks を選択し、[DHCPアドレスブロックの一覧/追加（List/Add DHCP Address Blocks）] ページを開きます。

アドレスブロックを追加するには、左側の [アドレスブロック（Address Blocks）] ペインの [アドレスブロックの追加（Add Address Block）] アイコンをクリックします。[アドレス（Address）] フィールドにネットワーク アドレスを入力し、ドロップダウンリストからアドレスマスクを選択します。たとえば、[アドレス（Address）] フィールドに「192.168.50.0」と入力し、ドロップダウンリストで 24 を選択して 192.168.50.0/24 アドレスブロックを作成します。

各サブネット マスクで使用可能なアドレスの数の確認については、次の表を参照してください。これらの使用可能なホストは、各範囲内の 2 つのネットワークアドレスとブロードキャスト・アドレスを除外します。

表 *13*：サブネット マスク

ネットワーク マスク	オクテット指定	各アドレス範囲で使用可能なホスト
/8	255.0.0.0	16777214
/9	255.128.0.0	8338606
/10	255.192.0.0	4194302
/11	255.224.0.0	2097150
/12	255.240.0.0	1048574
/13	255.248.0.0	524286
/14	255.252.0.0	262142
/15	255.254.0.0	131070
/16	255.255.0.0	65534
/17	255.255.128.0	32766
/18	255.255.192.0	16382
/19	255.255.224.0	8190
/20	255.255.240.0	4084
/21	255.255.248.0	2046
/22	255.255.252.0	1022
/23	255.255.254.0	510
/24	255.255.255.0	254
/25	255.255.255.128	126

ネットワーク マスク	オクテット指定	各アドレス範囲で使用可能なホスト
/26	255.255.255.192	62
/27	255.255.255.224	30
/28	255.255.255.240	14
/29	255.255.255.248	6
/30	255.255.255.252	2

CLI コマンド

address-block [vpn-name/]address/mask create [attribute=value ...] を使用します。次に例を示します。

```
nrcmd> address-block 192.168.0.0/16 create
```

テナント向け VPN のプライベート ネットワークの構成

テナントの VPN でプライベート ネットワークを構成するには、次の手順を実行します。

地域の高度な Web UI

ステップ 1 Web UI の上部にある [設定（Settings）] ドロップダウン リストの下にある [テナント（Tenant）] サブメニューから、必要なテナントを選択します。

ステップ 2 Web UI の上部にある [設定（Settings）] ドロップダウン リストの下にある [VPN] サブメニューから、必要な VPN を選択します。

ステップ 3 [設計（Design）] メニューに移動し、[DHCPの設定（DHCP Settings）] サブメニューの [VPN] を選択して、[VPNの一覧/追加（List/Add VPNs）] ページを開きます。Tenant-Private-Network 属性を true に設定します。この属性は、この VPN がローカル クラスター上のルーティング不可能な（RFC1918）テナントのアドレスを表すことを示します。

（注） これは地域の CCM クラスターにのみ適用され、ローカル クラスターに設定されている場合は無視されます。

ステップ 4 プライベートアドレスブロックを作成します（例：10.0.0.0/24）。

CLI コマンド

session set attribute=value を使用して、テナントと VPN を設定します。この VPN がローカル クラスター上のテナントのルーティング不可能なアドレスを表すことを示すには、 vpn name set tenant-private-network=value を使用します。次に、 address-block [vpn-name/]address/mask create を使用してプライベート アドレス ブロックを作成します。次に例を示します。

```
nrcmd-R> session set tenant=t1

nrcmd-R [Tenant:t1]> session set vpn=vpn1

nrcmd-R [Tenant:t1 VPN:vpn1]> vpn vpn1 set tenant-private-network=true

nrcmd-R [Tenant:t1 VPN:vpn1]> address-block 10.0.0.0/24 create
```

アドレス ブロックの委任

アドレス ブロックの委任は、ローカル クラスターに委任されているとして、地域クラスターで委任されたアドレスブロックをマークし、ローカルクラスターで委任されたアドレスブロックを作成する協調的なアクションです。アドレス ブロックをローカル クラスタに委任するには、アドレス ブロックに子アドレス ブロックまたはサブネットを含めることはできません。ローカル サーバーで作成される委任されたアドレス ブロックは、地域クラスタのアドレス と同じサイズにする必要があります。

一度に1つのローカルクラスターに委任できるアドレスブロックは1つだけです。複数のローカル クラスターに委任することはできません。アドレス ブロックを所有者に委任することもできます。

アドレス ブロックを委任するには、次の手順を実行する必要があります。

1. 中央の構成管理者に、アドレス ブロックを委任するローカル クラスタを作成してもらいます（『Cisco プライムネットワーク レジストラー 11.0 管理ガイド』の「サーバー クラスタの構成」セクションを参照）。
2. 中央の構成管理者に、地域クラスタをローカル クラスタと同期させます（『Cisco プライムネットワーク レジストラー11.0管理ガイド』の「ローカルクラスタとの同期」セクションを参照）。ローカルクラスタは、同期プロセスを通じて地域クラスタへのアドレス ソース参照を持ちます。
3. アドレス ブロックをクラスターまたは所有者に委任します。

CLI コマンド

地域クラスターに接続する場合は、**アドレス ブロック**名**デリゲート**クラスター名コマンドを使用できます。

サブネットからの逆引きゾーンの作成

手動で行う必要なく、[サブネットの一覧表示/追加（List/Add Subnets）]ページでサブネットから直接リバース ゾーンを作成できます（『Cisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイド』の「サブネットからの逆引きゾーンの追加」セクションを参照）。[逆引きゾーン（Reverse Zone）]タブをクリックし、ドロップダウンリストから構成済みのゾーンテンプレートを選択し、 Report をクリックして作成用の変更セットを表示します。[実行（Run）] をクリックして作成を確定します。

関連項目

サブネットの再利用 （121 ページ）

サブネットへのアドレス範囲の追加 （122 ページ）
アドレス ブロック、サブネット、スコープのアドレス使用率の表示 （125 ページ）
ローカル DHCP サーバーおよびルータへのサブネットのプッシュ （124 ページ）

サブネットの再利用

DHCP サーバーまたはルーターにサブネットを委任すると、必要に応じて再利用できます。

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、DHCPv4 サブメニューから Subnets を選択し、[サブネットの一覧表示/追加（List/Add Subnets）] ページを開きます。

ステップ 2 左側の [サブネット（Subnets）] ペインからサブネットを選択して、対応する [サブネットの編集（Edit Subnet）] ページを開きます。

ステップ 3 ページの上部にある Reclaim をクリックします。[サブネットの再利用（Reclaim Subnet）] ページが開きます。

ステップ 4 サブネットを強制的に削除する場合は、[強制削除（Force Delete）] チェックボックスをオンにします。

ステップ 5 Reclaim Subnet をクリックします。

（注） 管理対象ルーターまたは仮想ルーターのサブネットをプッシュまたは再利用する場合、これにより、すべての関連するサブネットおよびスコープに対してルーターに設定されているプライマリおよびセカンダリの関係も設定されます。ルーターの詳細については、『Cisco プライムネットワーク レジストラー 11.0 管理ガイド』の「ルーター用サブネットのプッシュと再利用」セクションを参照してください。

CLI コマンド

地域クラスターに接続する場合は、subnet name reclaim [-force] コマンドを使用できます。

アドレス ブロックへの子の追加

委任されていないアドレスブロックを子アドレスブロックまたはサブネットに分割できます。

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 Design メニューで、DHCPv4 サブメニューの下から Address Blocks を選択し、[アドレスブロックの一覧/追加（List/Add Address Blocks）] ページを開きます。

ステップ 2 委任（D）としてマークされていないアドレス ブロックの名前をクリックします。[住所ブロックの編集（Edit Address Block）] ページが開きます。

ステップ 3 子アドレス ブロックを追加するには、[子アドレス ブロック] セクションの [アドレス/マスク] フィールドに、アドレス ブロックのネットワーク アドレスの一部であるアドレスを追加します。親アドレス ブロックよりも大きいマスク値を選択し、Addをクリックします。

子アドレスブロックに子サブネットと同じネットワークアドレスを設定しようとすると、エラーメッセージが表示されます。

クリックAddしたときに値を省略すると、親アドレス空間のサブディビジョンが適切なマスク値で自動的に追加されます。たとえば、親スペースが 192.168.50.0/24 の場合、子サブネット値は省略し、Addをクリックすると、Web UI によって次の順序で子が追加されます。

192.168.50.0/26

192.168.50.64/26

192.168.50.128/26

192.168.50.192/26

ステップ 4 子サブネットを追加するには、アドレス ブロック ネットワーク アドレスの一部であるページの [子サブネット] セクションの [アドレス/マスク] フィールドにアドレスを追加しますが、親アドレス ブロックよりも大きいマスク値を選択します。次に、Add をクリックします。

子アドレス ブロックに対して同じネットワーク アドレスを子サブネットとして設定すると、エラー メッセージが表示されます。

をクリックAddしたときに値を省略すると、親アドレス空間のサブディビジョンが適切なマスク値で自動的に追加されます。たとえば、親スペースが 192.168.50.0/24 の場合、子サブネット値は省略し、Addをクリックすると、Web UI によって次の順序で子が追加されます。

192.168.50.0/26

192.168.50.64/26

192.168.50.128/26

192.168.50.192/26

サブネットへのアドレス範囲の追加

サブネットデータを編集し、サブネットに任意の数のアドレス範囲を追加できます。これらの範囲は、サブネットの指定されたネットワーク内に存在する必要があります。

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、 DHCPv4 サブメニューの下から Subnets を選択し、[サブネットのリスト/追加（List/Add Subnets）] ページを開きます。

ステップ 2 左側の[サブネット（Subnets）]ペインで、アドレス範囲を追加するサブネットの名前をクリックします。[サブネットの編集（Edit Subnet）] ページが開きます。

ステップ 3 ページの IP 範囲領域の開始フィールドに範囲の開始アドレスを入力し、終了アドレスを終了フィールドに追加します。これらのフィールドにホスト番号だけを追加すると、アドレスマスクで決定される範囲内の相対アドレスが使用されます。

ステップ 4 Add IP Range をクリックします。

ステップ 5 [保存（Save）] をクリックして、変更内容を保存します。

引っ張りと押し

ローカル クラスタからのレプリカ アドレス空間のプル

明示的に作成するのではなく、ローカル クラスタのレプリカ データからアドレス空間をプルすることもできます。

（注） IPv4 サブネットが削除されたローカル クラスタからレプリカ アドレス空間を取得しても、サブネット上のサーバー名は消去されません。サブネットは使用されなくなりましたが、サーバーに割り当てられていると見なされます。したがって、削除操作はサブネットに対して表示されないため、リージョン クラスターからサブネットを削除することはできません。サブネットを別のクラスターにプッシュまたは再割り当てする場合、またはリージョンのクラスターからサブネットを削除するには、まずサブネットを再利用する必要があります（サブネットの再利用 （121 ページ）を参照）。これにより、ローカル サーバーへの参照がクリアされます。

リージョン詳細Web UI

ステップ 1 [DHCPアドレスツリー（DHCP Address Tree）]（または[DHCPv6アドレスツリー（DHCPv6 Address Tree）]）ページで、[アドレスツリー（Address Tree）] ペインの [データのプル（Pull Data）] アイコンをクリックします。

ステップ 2 [プルレプリカアドレススペースの選択（Pull Replica Address Space）] ページ（または[プルレプリカIPv6アドレススペースの選択（Select Pull Replica IPv6 Address Space）]）ページで、次の手順を実行します。

- レプリカを引き出す間に予約を省略するには、Omit Reservations チェックボックスをオンにします。
- データ同期モード（Update、Complete、または Exact）を選択します。

ステップ 3 ページの上部または下部にある Report をクリックします。

ステップ 4 概要を確認し、OK をクリックします。

CLI コマンド

地域クラスターに接続すると、次のプル コマンドを使用できます。

- ccm pullAddressSpace < update | complete | exact > [-omitreservations] [-report-only] [-report]
- ccm pullIPv6AddressSpace < update | complete | exact > [-report-only] [-report]

ローカル DHCP サーバーおよびルータへのサブネットのプッシュ

サブネットをローカルの DHCP サーバーおよびルーターにプッシュできます。

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 中央の構成管理者にローカルクラスタを作成し、ローカルクラスタと再同期させます。

ステップ 2 リージョンクラスタでサブネットを作成します。

a) Design メニューで、DHCPv4 サブメニューから Subnets を選択します。[サブネットの一覧/追加（List/Add Subnets）] ページが開きます。

b) 左側の [サブネット（Subnets）] ペインの [サブネットの追加（Add Subnet）] アイコンをクリックします。

c) ネットワークアドレスを入力してサブネットのマスクを選択し、Add Subnet をクリックします。

ステップ 3 中央の構成管理者にスコープテンプレートを作成してもらい、サブネットを含むスコープを作成できるようにします。

a) 中央の構成管理者として、リージョナルクラスタにログインします。

b) Design メニューで、DHCPv4 サブメニューの下から Scope Templates を選択して、[DHCPスコープテンプレートの一覧/追加（List/Add DHCP Scope Templates）] ページを開きます。

c) 左側のペインの Add Scope Templates アイコンをクリックして、[DHCPスコープテンプレートの追加（Add DHCP Scope Template）] ページを開きます。

d) スコープテンプレートの名前を入力し、Add Scope Templateをクリックします。

e) [DHCPスコープテンプレートスコープ名の編集（Edit DHCP Scope Template scopename）] ページで、このページの他のエントリの中から、[範囲式（Range Expression）] フィールドに create-range 式を入力して、そのサブネットを持つスコープを作成します。（スコープテンプレートのポリシーを選択する場合は、ポリシーがローカルクラスタに存在することを確認するか、ポリシーをローカルクラスタにプッシュする必要があります。『Cisco プライムネットワーク レジストラー 11.0 管理ガイド』の「ローカル クラスタへのポリシーのプッシュ」セクションを参照してください）。

ステップ 4 リージョナルアドレスの管理者として、サブネットをローカルクラスタ DHCP サーバーに追加します。

a) リージョナルアドレスの管理者としてリージョナルクラスタにログインします。

b) Design メニューで、DHCPv4 サブメニューから Subnets を選択し、[サブネットの一覧表示/追加（List/Add Subnets）] ページを開きます。

c) 左側の [サブネット（Subnets）] ペインからサブネットを選択し、対応する [サブネットの編集（Edit Subnet）] ページを開きます。

d) ページの上部にある Push をクリックします。これにより、[プッシュサブネット（Push Subnet）] ページが開きます。

e) 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。

f) ドロップダウン リストからルータとルータ インターフェイスを選択します。

g) ドロップダウン リストからクラスタを選択します。
h) Push Subnet をクリックします。

CLI コマンド

リージョナルクラスタに接続する場合は、subnet name push cluster/failover-pair [-template=template-name] コマンドを使用できます。

アドレス空間の表示

アドレス空間は、IPv4 のアドレスブロックとサブネットの階層ツリーで、IPv6 ではプレフィックスを IP アドレス順に並べ替えています。ツリーを表示する深さのレベルを選択できます。すべての子ノードを再帰的に展開または縮小するノードを展開および縮小することもできます。新しいレベルを選択すると、以前の拡張または縮小が上書きされます。

ローカルの高度な Web UI と地域の高度な Web UI

アドレス空間を階層ツリーとして表示するには、次のようにします。

- Design メニューで、DHCPv4 サブメニューの下から Address Tree を選択し、[DHCPアドレスツリー（DHCP Address Tree）] ページを開きます。VPN を選択できます (構成されている場合)。
- Design メニューで、DHCPv6 サブメニューの下から Address Tree を選択し、[DHCPアドレスツリー（DHCP Address Tree）] ページを開きます。VPN を選択できることに注意してください（設定されている場合）。

アドレス ブロック、サブネット、スコープのアドレス使用率の表示

アドレス ブロック、サブネット、およびスコープの現在のアドレス使用率を表示できます。

ヒント IPv6 プレフィックスのアドレス使用率については、プレフィックスのアドレス使用率の表示（129 ページ）を参照してください。

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

この機能は、[DHCP アドレス ツリー] ページ、[DHCP アドレス ブロックの一覧/追加] ページ、および [サブネットの一覧/追加] ページで使用できます。Current Usage タブをクリックすると、使用率の詳細が表示されます。

（注） このページでサブネットとサーバーのマッピングを適切に行うには、関連するローカルクラスターとの整合性が取れるように、地域アドレス空間ビューを更新する必要があります。そのためには、レプリカのアドレス空間をプルするか、サブネットを再利用して DHCP サーバーにプッシュします（サブネットの再利用 （121 ページ））を参照）。また、特定の DHCP サーバーが実行されていることを確認します。

[現在の使用状況（Current Usage）] タブのその他の列では、次の項目を識別します。

- Type：アドレス空間がアドレス ブロック、サブネット、スコープのいずれであるか。
- Utilization：アドレスの使用状況と可用性を示す進行状況バーを表示します。
- View Utilization History：地域クラスタにのみ表示されます。**レポート**アイコン（）をクリックすると、使用率の詳細ページが開きます。

[現在の使用状況（Current Usage）] タブでは、[使用率の詳細（Utilization Detail）] 列項目が展開可能であるため、アドレスブロックまたはサブネットのスコープデータを表示できます。この列のアドレスブロック、サブネット、またはスコープ名の横にある [詳細の表示（View Details）] アイコン（）をクリックすると、選択したアイテムの [使用率の詳細（Utilization Details）] ページが開きます。

[使用率の詳細（Utilization Details）] ページは読み取り専用で、アドレス ブロック、サブネット、またはスコープの詳細なアドレス使用率属性をグラフと表形式で表示します。実際の使用状況の分割にドリルダウンする場合は、行の [詳細の表示（View Details）] アイコンをクリックすると、適切な凡例を持つグラフの形式で分割されたオーバーレイが表示されます。チャートのさまざまな部分にカーソルを合わせると、その特定の種類の使用方法の詳細を確認できます。[テーブル（Table）] タブをクリックして、下の表に示す住所使用率属性を表示します。

表 *14 :* アドレス使用率属性

使用属性	説明
Tenant	管理者に関連付けられているテナント組織またはグループ
aggregation-level	この使用率データの粒度。スコープ レベルは、DHCP サーバーから入手できる最も詳細なデータです。集計されたカウンタは、サブネット レベルおよび追加ブロック レベルまたはネットワーク レベルで報告されます。これらは、特定のサブネット、アドレスブロック、またはネットワーク内のスコープ レベルのデータの合計です。
合計アドレス数	
total-dynamic	予約済みリースを除く、リースの合計数。
total-reserved	予約済みリースの合計数。
Free Dynamic	

使用属性	説明
avail	クライアントに発行できる動的リースの数。
other-avail	DHCP フェールオーバー パートナーが現在クライアントに対して問題に使用できる動的リースの数。
Active Dynamic	
offered	現在クライアントに提供されているが、リースされているとしてまだ確認されていない動的リースの数。
leased	現在クライアントにリースされている動的リースの数。
expired	リースの有効期限を過ぎても、他のクライアントでは利用できない動的リースの数 (ポリシーの猶予期間が終了した後を除く)。
pend-avail	再発行しなかったフェールオーバー パートナーからの確認応答を待機している動的リースの数。
Reserved	
reserved-active	クライアントがアクティブに使用している予約済みリースの数。
reserved-inactive	クライアントがアクティブに使用していない予約済みリースの数。
Unavailable	
unavail	クライアントが拒否する予約されていない動的リースの数、またはサーバーがアドレスの競合 (通常は修正が必要な構成を示す) でマークします。
reserved-unavail	クライアントが拒否する予約済みリースの数、またはサーバーがアドレスの競合 (通常は修正が必要な構成を示す) でマークします。
Deactivated	
deactivated	クライアントがアクティブにリースしている (提供、期限切れ、リリースされていない) 動的リースと予約リースの数。
leased-deactivated	クライアントがアクティブにリースしている (提供、期限切れ、リリースされていない) 動的リースの数。
reserved-leased- deactivated	クライアントがアクティブにリースしている (提供、期限切れ、リリースされていない) が、管理者が非アクティブ化した予約リースの数。
Additional Attributes	
primary-subnet	このサブネット レベルまたはスコープ レベルの使用率データのプライマリ サブネット。

使用属性	説明
selection-tags	範囲レベルの使用率データに関連付けられた選択タグのコンマ区切りリスト。

アドレス ブロック、サブネット、アドレス タイプの表示

ネットワーク用に作成されたアドレス ブロックとサブネットを表示できます。

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

メニューからDesign 、Address Treeサブメニューの下DHCPv4で選択して[DHCPアドレスツリー]ページを開きます。

アドレス空間の深さのレベルを選択するには、左側の [アドレス ツリー] ペインでアドレスの 1 つをクリックします。住所の詳細がページに表示されます。[アドレスの種類] 列には、表示されるオブジェクトの種類、アドレスブロック、またはサブネットが示されます。所有者列はアドレス スペースの所有者を識別し、Region 列はアドレス スペースに割り当てられた領域を識別します。

動的に割り当てられたアドレス・スペースは、「Dアドレス・タイプ」列に「委任」の場合に示されます。この委任されたアドレス スペースは削除できません。

そうでない場合、[**更新**（Refresh）アイコンをクリックすると、表示がリフレッシュされます。

アドレスの種類は、追加、変更、および削除できます。メニューからDesign サブDHCP Settingsメニューの Address Types下を選択して、[アドレスの種類の一覧/追加] ページを開きます。左側の [アドレスの種類] ペインの [**アドレスの種類の追加**] アイコンをクリックして 、[アドレススペースの種類の追加] ページを開き、[アドレスの種類の編集] ページで設定を変更します。[リスト/アドレスの種類の追加]ページで、レプリカ アドレスの種類、プッシュ アドレスの種類、およびアドレスの種類の再利用を行うこともできます。

CLI コマンド

address-type name create [attribute=value] を使用して、アドレスタイプを作成します。

address-type name delete を使用して、追加タイプを削除します。

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。

- address-type < name | all > pull < ensure | replace | exact > cluster-name [-report-only | -report]
- address-type < name | all > push < ensure | replace | exact > cluster-list [-report-only | -report]
- address-type name reclaim cluster-list [-report-only | -report]

IPv6 アドレス空間の表示

Design メニューの、ローカルの高度な Web UI または地域の高度な Web UI で DHCPv6 サブメニューの下から Address Tree を選択し、DHCP v6 アドレスツリーページを表示します。この

ページは、IPv4 の DHCP アドレスツリーページに似ています（アドレス空間の表示 （125 ページ）を参照）。[統一されたv6アドレススペースの表示（View Unified v6 Address Space）] ページでは、次の操作を実行できます。

- Web UI の上部にある [設定（Settings）] ドロップダウン リストの下にある [VPN] サブメニューからアドレス空間の VPN を設定します。
- アドレスツリーペインで [IPv6プレフィックスの追加（Add IPv6 Prefix）] アイコンをクリックして接頭辞を追加し、プレフィックス名、アドレスを入力し、プレフィックスタイプとプレフィックステンプレートを選択します。Add IPv6 Prefix をクリックします（プレフィックスの作成と編集 （161 ページ）を参照）。
- アドレス ツリー ペインで名前を選択して、プレフィックスを編集します。これにより、[プレフィックスの編集（Edit Prefix）] ページが開きます（プレフィックスの作成と編集 （161 ページ）を参照）。
- プレフィックス空間の現在の使用状況を表示します（プレフィックスのアドレス使用率の表示 （129 ページ）を参照）。

プレフィックスのアドレス使用率の表示

接頭語の現在のアドレス使用率を表示できます。

ローカルアドバンスドおよびリージョン Web UI

この機能は、DHCP v6 アドレスツリーページアドレス空間の表示 （125 ページ）で使用できます。

ヒント DHCP v6 アドレス ツリー ページを使用して、プレフィックスをプッシュおよび再利用できます。目的の接頭辞の Push アイコンまたは Reclaim アイコンをクリックします。（詳細については、プレフィックスの作成と編集 （161 ページ）を参照してください）。

[現在の使用状況（Current Usage）] タブをクリックすると、使用率の詳細が表示されます。

（注） このページで適切なプレフィックスとサーバーのマッピングを行うには、関連するローカルクラスターとの整合性が取れるように、地域アドレス空間ビューを更新する必要があります。これを行うには、v6 レプリカ アドレス空間をプルするか、プレフィックスを DHCP サーバーにプッシュします。また、特定の DHCP サーバーが実行していることを確認します。

[現在の使用状況（Current Usage）] タブの下の他の列では、次の項目を識別します。

- Range：プレフィックスのアドレス範囲。
- Type：アドレス空間がプレフィックスかリンクか。
- Active Dynamic：DHCP によって管理されるダイナミック レンジの一部であり、現在リースされているが予約されていないアドレス。

- Active Reserved：地域クラスタにのみ表示されます。ダイナミック レンジの一部であり、予約されているアドレス。
- View Utilization History：地域クラスタにのみ表示されます。**レポート**アイコン（）をクリックすると、使用率の詳細ページが開きます。

[現在の使用状況（Current Usage）] タブでは、[使用率の詳細（Utilization Detail）] 列が展開可能なので、プレフィックスまたは親プレフィックスのデータを表示できます。この列のプレフィックスまたは親プレフィックスの横にある [詳細の表示（View Details）] アイコン（）をクリックすると、選択したアイテムの [使用率の詳細（Utilization Detail）] ページが開きます。

[使用率の詳細（Utilization Detail）] ページは読み取り専用のページで、プレフィックスまたは親プレフィックス（合計として識別される）の詳細なアドレス使用率属性を表示します。実際の使用状況の分割にドリルダウンする場合は、行の [詳細の表示（View Details）] アイコンをクリックすると、適切な凡例を持つグラフの形式で分割されたオーバーレイが表示されます。チャートのさまざまな部分にカーソルを合わせると、その特定の種類の使用方法の詳細を確認できます。[テーブル（Table）] タブをクリックして、下の表に示す住所使用率属性を表示します。

表 *15：*アドレス使用率属性

使用属性	説明
Tenant	このプレフィックスのテナント所有者。
aggregation-level	使用率データの粒度。プレフィックス レベルは、データが特定のプレフィックス用であることを示します。totals は、データが親プレフィックスの場合、そのプレフィックス レベルのカウンターの合計であることを示します。
dhcp-type	DHCP アドレス割り当てタイプは、dhcp (ステートフル)、ステートレス (オプション構成)、プレフィックス委任、またはインフラストラクチャ (クライアント アドレスをアドレス プールのないリンクにマップ) です。
Total Addresses	
active-dynamic	アクティブな使用中の動的リースの総数 (リース、提供、リリース、期限切れ、または取り消し済み)。アクティブ動的カテゴリには、これらのリースの状態が表示されます。
total-reserved	予約済みリースの合計数。
Active Dynamic	
offered	現在クライアントに提供されているが、リースされているとしてまだ確認されていない動的 (予約されていない) リースの数。
leased	現在クライアントにリースされている動的リースの数。

使用属性	説明
expired	リースの有効期限を過ぎても、他のクライアントでは利用できない動的リースの数 (ポリシーの猶予期間が終了した後を除く)。
revoked	クライアントが使用できなくなったが、他のクライアントが使用している可能性がある動的リースの数。
Reserved	
reserved-active	クライアントがアクティブに使用している予約済みリースの数。
reserved-inactive	クライアントがアクティブに使用していない予約済みリースの数。
Unavailable	
unavail	クライアントが拒否する予約されていない動的リースの数、またはサーバーがアドレスの競合 (通常は修正が必要な構成を示す) でマークします。
reserved-unavail	クライアントが拒否する予約済みリースの数、またはサーバーがアドレスの競合(通常は修正が必要な構成を示す)でマークします。
Deactivated	
deactivated	クライアントがアクティブにリースしている(提供、期限切れ、リリースされていない) 動的リースと予約リースの数。
leased-deactivated	管理者が非アクティブ化した動的リースの数。
reserved-leased- deactivated	管理者が非アクティブ化した予約済みリースの数。
プレフィックス委任リース数	
max-pd-balancing-length	プレフィックス委任リースのカウントに使用されるプレフィックス長。
prefixes-in-use	使用中の最大-pdバランシング長プレフィックス長のプレフィックスの数。
prefixes-available	このサーバー上の任意のクライアントが使用できる、最大-pdバランシング長プレフィックス長のプレフィックスの数。
prefixes-other-available	フェールオーバー パートナー上の任意のクライアントで使用できる、最大 pd バランシング長プレフィックス長のプレフィックスの数。
prefixes-in-transition	フェールオーバーパートナー間の遷移における、最大pdバランシング長プレフィックス長のプレフィックスの数。
フェールオーバー関連	

使用属性	説明
available	このサーバー上の任意のクライアントが使用できるプレフィックス委任リースの数。これは、リース オブジェクトの数であり、特定のプレフィックス長のプレフィックスの数ではありません。
other-available	このサーバーがパートナーが任意のクライアントで使用できると考えるプレフィックス委任リースの数。
pending-available	保留中の状態にあるリースの数。
pending-delete	保留中の削除状態にあるリースの数。
追加属性	
cluster-id	この使用率データを報告したローカル クラスター。
link-name	このプレフィックス レベルの使用率データのリンク。
owner	このプレフィックスまたはリンクに関連付けられている所有者。
region	このプレフィックスまたはリンクに関連付けられたリージョン。
selection-tags	プレフィックス レベルの使用率データに関連付けられた選択タグのコンマ区切りリスト。
timestamp	この使用率データが収集された時刻。

使用率履歴レポートの生成

サブネット内に割り当てられたアドレスの数と空きアドレス空間を確認できるように、使用率履歴データを抽出できます。追加の管理機能を使用して、レコードの使用率データベースをトリミングおよび圧縮し、データベースのサイズを管理することができます。

関連項目

使用率履歴データの照会 （132 ページ）
使用率履歴データのトリミングと圧縮 （134 ページ）

使用率履歴データの照会

ローカルが地域またはデフォルトのポーリング(1時間ごと)または手動ポーリングで登録されている場合、DHCP使用率データが収集されます。使用可能なすべてのスコープとプレフィックス情報は、地域サーバーによって収集されます。

既定では、クラスターオブジェクトの作成時にこれらの値が設定解除されるため、すべてのクラスターでポーリングが有効になります。地域 CCM サーバーのグローバル設定は、値が設定されていない場合に、すべてのクラスターのポーリング間隔とオフセットを制御します。ローカルクラスタでこれらの値を設定すると、サーバーのデフォルト値が上書きされます。addrutil ポーリング間隔が 0 に設定されている場合、そのクラスターのポーリングは無効になります。

リージョン Web UI

ステップ 1 使用率データを照会するための選択基準を設定する必要があります：[詳細（Advanced）] モードの Operate メニューで、Reports サブメニューの下から DHCP Utilization History を選択します。[クエリ使用率の履歴（Query Utilization History）] ページが開きます。

ステップ 2 v4 の履歴または v6 の履歴ラジオ ボタンを有効にすることで、v4とv6の使用率の履歴を照会できます。また、次の条件に基づいて使用率履歴を照会することもできます。

1. **時間範囲**：リース履歴データの時間範囲を次の中から選択します。
   - 今日
   - 過去 10 日間
   - 過去 30 日間
   - 過去 60 日間
   - 過去 90 日間
   - 期間指定(90 日まで)

     この値を選択する場合は、ドロップダウンリストから [開始日（Start Date）] と [終了日（End Date）] の月、日、および年も選択します。結果は、addrutil-poll-interval 属性の値によって異なります。
2. **所有者**：隣接するドロップダウン リストから所有者を選択します。
3. **リージョン**：隣接するドロップダウン リストからリージョンを選択します。
4. **集計レベル**：隣接するドロップダウン リストから集計レベルを選択します。
5. **サブネット**：横にあるドロップダウン リストからサブネットを選択します。
6. **クラスタ**：横にあるドロップダウン リストからクラスタを選択します。

ステップ 3 ドロップダウンリストからフィルタ属性とタイプを選択し、[値（Value）] フィールドで選択したフィルタ タイプの値を入力します。+ アイコンをクリックしてフィルタを追加します。既存のカスタム フィルタがある場合は、その横にある [X] アイコンをクリックしてフィルタを削除できます。

ステップ 4 [フィルタの適用（Apply Filter）] をクリックして結果を表示します。[詳細の表示（View Details）] 列の [レポート（Report）] アイコン（  ）をクリックすると、[使用状況の履歴の詳細（Utilization History Details）] ページが開きます。

使用率履歴データのトリミングと圧縮

サブネットおよびプレフィックスの使用率履歴データベースは自動的にトリミングされます。CCM サーバーは、一定の経過時間より古い使用率データを一定の間隔で切り取る、地域クラスターでバックグラウンドトリミングを実行します。トリミング間隔は24時間にプリセットされ、年齢(トリミング前にどれくらい戻るか)は24週に設定されます。

また、特定の経過時間より古いレコードを圧縮して、保存される履歴の量を減らすこともできます。コンパクトな間隔ごとに最初のデータ ポイントのみが保持されます。その他のデータ ポイントはすべて削除されます。

データベースの値を調整し、使用率データベースのトリミングと最適化を実行するには、データベース サブロールを割り当てられた中央の構成管理者である必要があります。

リージョン詳細Web UI

ステップ 1 メニューからOperate サブServersメニューのManage Servers下を選択して、[サーバーの管理] ページを開きます。

ステップ 2 左側LocalCCMのServer[サーバーの管理] ウィンドウでリンクをクリックして、[ローカル CCM サーバーの編集] ページを開きます。

ステップ 3 [アドレス使用率の設定] で、次の属性を設定します。

a) addrutil-poll-interval：サブネットとプレフィックスの使用率をすべての DHCP サーバーから収集する頻度。0 に設定すると、ポーリングは無効になります。

b) addrutil-poll-retry：ポーリングが失敗した場合に、指定されたポーリング間隔の再試行回数。

c) addrutil-poll-offset：サブネット使用率のポーリングに対して固定の時刻を指定します。この時間は、0 が午前 0 時の時刻オフセットとして解釈され、ポーリング間隔が 24 時間未満で、オフセット値がポーリング間隔より小さい場合に限ります。オフセット値がポーリング間隔より大きい場合、または間隔が 24 時間を超える場合、オフセットは無視されます。

ポーリングのスケジューラは、最初のポーリング イベントがオフセット時に発生することを確認します。たとえば、間隔を 4 時間に設定し、オフセットを午前 2 時に設定すると、投票は午前 2 時、午前 6 時、午前 10 時、午後 2 時、午後 6 時、午後 10 時に行われます。

d) addrutil-trim-interval：古いサブネットとプレフィックスの使用率データを自動的にトリミングする頻度。デフォルトではデータをトリミングしません。バックグラウンドトリミングをトリガーするには、この値を設定する必要があります。制限値は 0 ~ 1 年で、単位は秒 (s)、分 (m)、時間 (h)、日 (d)、週 (w)、月 (m)、および年 (y) で使用できます。

e) addrutil-trim-age：古いサブネットとプレフィックスの使用率データを自動的にトリミングするために遡る期間。プリセット値は 24 週間です。(ただし、トリミングを有効にするには、addrutil-trim-interval値を 0 以外に設定する必要があります。制限値は 24 時間から 1 年で、単位は秒 (s)、分 (m)、時間 (h)、日 (d)、週 (w)、月 (m)、および年 (y) で使用できます。

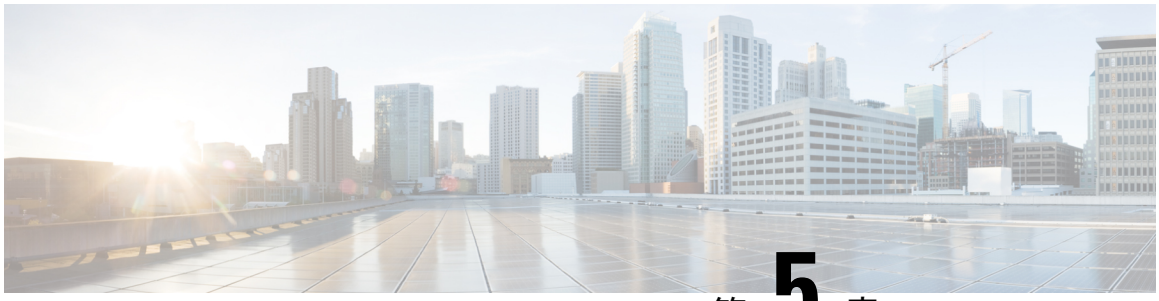
ステップ 4 また、即時のトリミングと圧縮を強制することができます。トリミング/圧縮セクションを見つけます。

a) Trim/Compact age：データのトリミングするために遡る期間。この値に対する境界はありません。ただし、非常に小さい値(1m など)を設定すると、最新のデータをトリミングまたは圧縮しますが、これは望ましくない場合があります。実際、ゼロに設定すると、収集されたデータがすべて失われます。値

を大きくし過ぎる (10y など) に設定すると、データのトリミングや圧縮が行えなくなる可能性があります。

b) Compact interval：Trim/Compact age よりも古いサブネットとプレフィックス使用率レコードを圧縮する時間間隔。この間隔は、ポーリング間隔の倍数になる場合があります。たとえば、コンパクト間隔がポーリング間隔の 2 倍に設定されている場合、その間隔は 1 つおきに削除されます。

ステップ 5 すぐにトリミングする場合は、ページのTrim All Utilization Data 下部にあるコントロールをクリックします。データを圧縮する場合は、 をCompact All Utilization Dataクリックします。



第 5 章

スコープ、プレフィックス、リンク、ネットワークの管理

動的ホスト構成プロトコル (DHCP) は、IP 構成をデバイスに自動的に割り当てるための業界標準のプロトコルです。DHCP は、アドレスの割り当てにクライアント/サーバー モデルを使用します。管理者は、1 つまたは複数の DHCP サーバーを設定し、IP アドレスの割り当てや、その他の TCP/IP 指向の設定情報をデバイスに提供することができます。DHCP を使用すると、IP アドレスを各クライアントに手動で割り当てるという作業を省くことができます。DHCP プロトコルは RFC 2131 で説明されています。プロトコルの概要については、を参照してくださいダイナミック ホスト コンフィギュレーションの概要 （1 ページ）。

この章では、スコープ、プレフィックス、およびリンクを設定する方法について説明します。クライアントがアドレス割り当てに DHCP を使用できるようにするには、少なくとも 1 つのスコープ (動的アドレス プール) またはプレフィックスをサーバーに追加する必要があります。

スコープの管理

この項では、DHCP サーバーのスコープを定義および設定する方法について説明します。スコープは、DHCP サーバーが管理するサブネット内の 1 つ以上の動的アドレス範囲で構成されます。DHCP サーバーがクライアントにリースを提供する前に、1 つ以上のスコープを定義する必要があります。(リースの一覧表示とスコープのリース予約の定義の詳細については、リースの管理 （227 ページ）

関連項目

スコープ テンプレートの作成と適用 （173 ページ）

スコープの作成 （138 ページ）

スコープの作成

スコープの作成は、ローカル クラスタ関数です。各スコープには、次の項目が必要です。

- [名前（Name）]
- リース時間、猶予期間、およびオプションを定義するポリシー
- ネットワーク アドレスとサブネット マスク
- 範囲、またはアドレスの範囲

スコープはローカル クラスタでのみ構成できます。Web UI ページは、ローカルの基本モードと高度なモードで異なります。

ローカルの基本 Web UI

ステップ 1 [**デザイン**] メニューからScopesサブメニューDHCPv4を選択し、[DHCP スコープの一覧/追加] ページを開きます。

ステップ 2 必要に応じて、Web UI の上部にある[**設定**] ドロップダウン リストから、スコープの VPN を選択します。

ステップ 3 [スコープ] ウィンドウの [**スコープの追加**] アイコンをクリックし、スコープ名を入力して、サブネットの IP アドレスを入力し、ドロップダウン リストからマスク値を選択します。

ステップ 4 必要に応じて、ドロップダウン リストからスコープの構成済みサービス クラス (クライアント クラス) を選択します。

ステップ 5 Add DHCP Scope をクリックします。

ステップ 6 DHCP サーバーをリロードします。

（注） スコープが Basic モードで作成されると、範囲とルーター アドレスが自動的に追加されます。これらのモードを変更する場合は、基本モードでは設定できないため、モードを [詳細設定] に変更する必要があります。

ローカル アドバンスド Web UI

ステップ 1 [**デザイン**] メニューからScopesサブメニューDHCPv4を選択し、[DHCP スコープの一覧/追加] ページを開きます。

ステップ 2 必要に応じて、Web UI の上部にある[**設定**]ドロップダウン リストから、スコープの VPN を選択します。

ステップ 3 [スコープ] ウィンドウの [**スコープの追加**] アイコンをクリックするか、スコープ名を入力するか、または空白のままにして、スコープテンプレートのスコープ名式で定義されているスコープスコープテンプレートでの式の使用 （183 ページ）名式を使用します (ある場合は参照)。後者の場合は、スコープ テンプレートを選択します。スコープには常にサブネットとマスクを入力する必要があります。

ステップ 4 ドロップダウン リストからスコープのポリシーを選択します。ポリシーはデフォルトでデフォルトポリシーになります。

ステップ 5 Add DHCP Scope をクリックします。

ステップ 6 スコープ内のアドレスの範囲を追加します。範囲は、定義されたスコープの任意のサブセットにできますが、重複することはできません。ホスト番号だけを入力した場合、範囲はネットマスクを基準にします。ローカル ホストまたはブロードキャスト アドレス (通常は 0 と 255) を含む範囲を入力しないでください。範囲を追加し、Add Rangeをクリックします。

ステップ 7 DHCP サーバーをリロードします。

ヒント スコープに関連付けられているリースと予約を表示するには、「」リースの管理 （227 ページ）を参照してください。リースを検索するには、「」をサーバー全体のリースの検索 （241 ページ）参照してください。

関連項目

サーバー上のスコープ数の取得 （148 ページ）
複数のスコープの設定 （139 ページ）
スコープの編集 （146 ページ）
段階的な同期モード （147 ページ）

複数のスコープの設定

同じネットワーク番号とサブネットマスクを使用して、(分離されたアドレス範囲を持つ)複数のスコープを構成できます。既定では、DHCP サーバーは、同じサブネット上のすべてのスコープから使用可能なリースをプールし、リースを要求するクライアントにラウンドロビン方

式でリースを提供します。ただし、各スコープに割り当て優先順位を設定することで、このラウンドロビン割り当てを割り当て優先順位を使用した複数スコープの設定 （140ページ）回避することもできます(「」を参照してください)。

単一サブネットのアドレスを複数のスコープに構成すると、より自然な方法でアドレスを管理しやすくなっています。スコープごとに実質的に無制限の数のリースを構成できますが、数千のリースを持つスコープがある場合は、それらを並べ替えるのに時間がかかることがあります。これは、リースを複数のスコープに分割する動機となります。

リースの種類に応じて、スコープ間でリースを分割できます。各スコープは個別の予約リストを持つことができるので、1 つのオプションとリース時間のセットを持つポリシーを持つ 1 つのスコープに動的リースを配置し、別のスコープ内のすべての予約を別のスコープに置くことができます。複数のスコープの一部がローカルに接続されていない場合は、適切なヘルパーアドレスを使用してルーターを構成する必要があります (BOOTP リレーサポートを持つ) 必要があります。

関連項目

ラウンドロビン アドレス割り当てのための複数スコープの設定 （140 ページ）

割り当て優先順位を使用した複数スコープの設定 （140 ページ）

ラウンドロビン アドレス割り当てのための複数スコープの設定

既定では、DHCPサーバーはラウンドロビン方式で複数のスコープを検索します。このため、DHCP クライアント要求の種類によってスコープをセグメント化する必要があります。サブネット上でセカンダリ スコープを使用して複数のスコープが使用できる場合、DHCPサーバーは、受信DHCPクライアント要求を満たすものを検索します。たとえば、サブネットに3つのスコープがあり、そのうちの1つだけが動的BOOTPをサポートする場合、予約のないBOOTP要求は、動的 BOOTP をサポートする 1 つの要求によって自動的に処理されます。

また、DHCP 要求を許可しないようにスコープを構成することもできます (既定では、スコープの許可です)。これらの機能を組み合わせて使用することで、すべての DHCP 要求が 1 つのスコープ (およびアドレス範囲) から満たされるようにサブネット上のアドレスを簡単に構成できます。このようにして、DHCP クライアントをサポートするアドレス プールへの影響を最小限に抑えながら、動的 BOOTP をサポートできます。

割り当て優先順位を使用した複数スコープの設定

前項で説明したデフォルトのラウンドロビンの動作の代わりに範囲内で割り当ての優先順序を設定することができます。このようにして、配分プロセスをより詳細に制御できます。また、サブネット内から連続してアドレスを割り当て、DHCPサーバーのフェールオーバーを使用する場合にバックアップ サーバーに割り当てられるアドレスのブロックDHCP フェールオーバーの管理 （65ページ）を制御するように、DHCPサーバーを構成することもできます(を参照)。

通常のインストールでは、スコープの割り当て優先順位属性を使用して、すべてのスコープの割り当ての優先順位を設定します。インストールによっては、スコープで最初に使用可能な割り当て属性を有効にしたい場合もありますが、多くの場合は有効にしません。割り当て先使用

可能を使用する場合はパフォーマンスが低下する可能性が小さいため、絶対に必要な場合にのみ使用してください。

次の制御が可能です。

- 最初にアドレスを割り当てる必要があるスコープ間の階層。
- 最も最近アクセスされたアドレスの既定の動作ではなく、スコープで最初に使用可能なアドレスを割り当てるかどうかを指定します。
- スコープのフェールオーバー構成で、連続するターゲット アドレスを割り当てる。
- 優先度アドレス割り当てサーバー全体。
- スコープの割り当て優先順位が等しい場合、サーバーが使用可能なアドレスの数が最も多いアドレスまたは最も多いアドレスからアドレスを割り当てる必要があるかどうか。

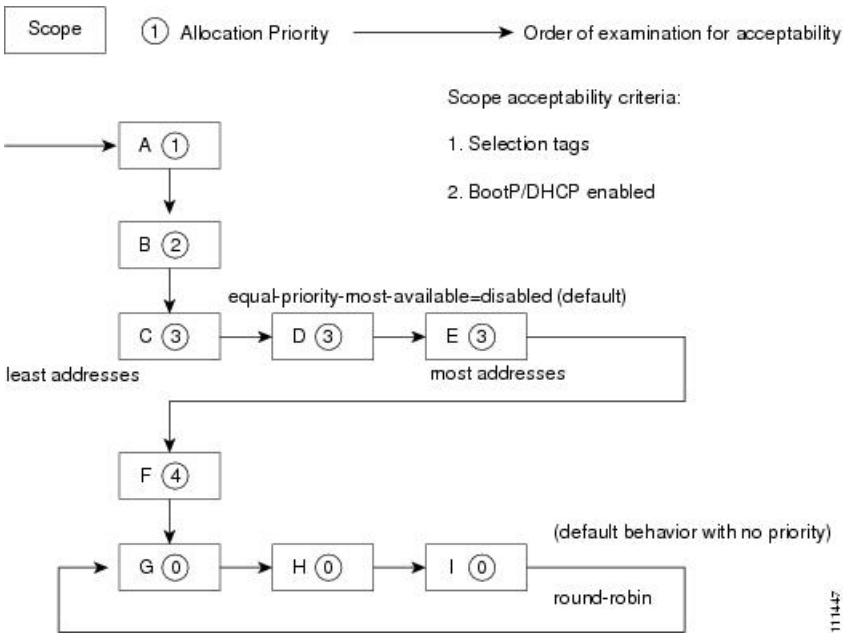
ネットワークに複数のスコープがある場合、DHCPは、既存のアドレスに関連付けられていない DHCP クライアントからの DHCPDISCOVER 要求を処理するときに、どのスコープから IP アドレスを割り当てるかを決定する必要があります。DHCPサーバーがこの割り当てを実行するために使用するアルゴリズムについては、次のセクションで説明します。

割り当て優先順位のアルゴリズム

DHCP サーバーは、ネットワーク内のスコープを 1 つずつ調べて、許容できるかどうかを判断します。受け入れ可能なスコープが見つかると、DHCPDISCOVER 要求を満たすために IP アドレスを割り当てようとします。割り当て優先順位のスコープ属性は、割り当て優先順位がない場合、DHCP サーバーがラウンドロビン順序でスコープを調べるため、DHCP サーバーにネットワーク内のスコープを特定の順序で調べるように指示するために使用されます。

下の図は、9つのスコープを持つネットワークの例を示しています(これは珍しいことですが、割り当て優先順位を使用するいくつかの可能性を示しています)。

図 *11* : スコープ割り当て優先順位



これらのスコープのうち6つは割り当て優先順位で構成され、そのうちの3つは設定されませんでした。サーバーは、割り当て優先順位で構成された6個を、優先順位の最も低いものから最も高い順に調べます。サーバーは、許容可能なスコープを検出すると、そこからIPアドレスを割り当てようとします。サーバーが成功すると、このアドレスを使用してDHCPDISCOVER要求の処理が終了します。そのスコープからアドレスを割り当てることができない場合は、スコープの調査を続行して別の受け入れ可能なスコープを探し、そこからアドレスを割り当てようとします。

このプロセスは、同じ割り当て優先順位が構成されているスコープがない場合は簡単ですが、例のように複数のスコープが同じ0以外の割り当て優先順位を持つ場合、サーバーは等しいスコープの間で選択する方法を持っている必要があります。優先順位。既定の動作では、使用可能なアドレスが最も少ないスコープから始まり、優先順位が等しいスコープを調べます。これにより、別のスコープの他のアドレスを使用する前に、1つのスコープ内のすべてのアドレスが使用されます。これは上の画像に示されている状況です。優先順位が最も高いDHCPサーバー属性を有効にすると、状況が逆転し、2つのスコープの優先順位が等しい場合に、使用可能なアドレスが最も多いスコープが最初に調べられます。これにより、スコープの使用率が広がり、割り当て優先順位が等しいすべてのスコープにアドレスの使用が均等に分散されます。

優先順位が同じスコープの処理に別の機能があるため、この優先順位が最も高い方法を使用できます。同じ優先順位のスコープが2つある状況では、サーバーがアドレスを割り当てようとしているDHCPDISCOVER要求にも制限IDが設定されている場合(オプション82の制限機能を使用しています。オプション82を使用したサブスクライバの制限（367ページ）したがって、同じ制限idを持つすべてのクライアントは、同じ優先順位のスコープ内の使用可能なアドレスの数や、優先順位が最も高いサーバー属性の設定に関係なく、同じスコープからアドレスを割り当てる傾向があります。

これを、最も利用可能な同等の状況に戻すために、最も利用可能な同等の優先度を構成し(かつ、いくつかの等しい優先順位スコープを持つ)、特定の制限idを持つ最初のDHCPクライアントが、使用可能なアドレスが最も多いスコープからアドレスを取得します(同じ制限idを持つクライアントが他に存在しないため)。その後、同じ制限IDを持つ後続のクライアントはすべて同じスコープに入るようになります。この構成の結果、最初のクライアントは許容可能な同等の優先順位のスコープに均等に分散され、後続のクライアントは同じ制限idを持つ既存のクライアントとクラスタ化されます。

同じネットワークに割り当て優先順位が構成されているスコープと、割り当て優先順位のないスコープが存在する場合、割り当て優先順位がゼロ以外のすべてのスコープが最初に受け入れ可能かどうかを調べます。その後、どのスコープも許容可能で、使用可能なIPアドレスも持っていない場合、割り当て優先順位のない残りのスコープはラウンドロビン方式で処理されます。このラウンドロビン検査は、現在のDHCPDISCOVERを送信する場合と同じ制限IDを持つ既存のDHCPクライアントがある場合を除き、このネットワークで最後に調べたスコープを超える次のスコープで開始されます。この場合、ラウンドロビンスキャンは、既存のクライアントIPアドレスの取得元のスコープから開始されます。これにより、そのスコープが許容範囲であり、割り当て可能なIPアドレスがある場合、その制限idを持つ後続のクライアントは、その制限idを持つ最初のクライアントと同じスコープからアドレスを引き出します。

アドレス割り当て属性

アドレス割り当てに対応する属性を次の表に示します。

表 *16 :* アドレスの割り当ての優先度の設定

属性	タイプ	説明
allocation-priority	スコープ (設定または設定解除)	定義されている場合、すべてのスコープのアドレスが使い果たされるまで、優先順位の高い許容範囲からアドレス割り当てが行われるような順序付けをスコープに割り当てます。割り当て優先順位が 0 (事前設定値) の場合、スコープに割り当て優先順位がないことを意味します。優先度 1 が最も高い優先順位で、増加する数の優先順位が低くなります。スコープと割り当て優先度を混在させることもできます。この場合、優先度のあるスコープは、優先順位のないスコープよりも先に受け入れ可能なスコープを調べます。 この属性を設定すると、DHCP サーバーの優先度- アドレス - 割り当て属性の設定が上書きされます。ただし、割り当て優先順位が設定解除され、優先順位アドレス割り当てが有効になっている場合、スコープの割り当て優先順位はそのサブネット・アドレスになります。割り当て優先順位が設定解除され、優先順位アドレス割り当てが無効になっていると、スコープは既定のラウンドロビン方式で調べられます。
割り当て先 - 使用可能	スコープ (有効または無効)	有効にすると、このスコープから新しいアドレスのすべての割り当てが、最初に使用可能なアドレスから割り当てられます。無効にした場合 (プリセット値)、最近アクセスしたアドレスが使用されます。この属性を設定すると、DHCP サーバーの優先度- アドレス割り当て属性の設定が上書きされます。ただし、設定解除と優先順位アドレス割り当てが有効な場合、サーバーは最初に使用可能なアドレスを割り当てます。割り当て先使用可能な設定解除と優先順位アドレス割り当てが無効になっていると、スコープは既定のラウンドロビン方式で調べられます。

属性	タイプ	説明
フェールオーバーバックアップ - 割り当て- 境界	スコープ（設定または設定解除）	最初に使用可能な割り当てが有効で、スコープがフェールオーバー構成にある場合、この値は、バックアップサーバーにアドレスを割り当てるポイントとして使用するIPアドレスです。この境界を下回るアドレスのみがバックアップサーバーに割り当てられます。この境界の下に使用可能なアドレスがない場合、その上のアドレスはバックアップ・サーバーに割り当てられます。実際の割り当てはこのアドレスから下がり、DHCP クライアントの通常の割り当てはスコープ内の最下位アドレスから上に働きます。 この属性が設定されていないか、ゼロに設定されている場合、使用される境界は、範囲の範囲内の最初と最後のアドレスの中間になります。この境界の下に使用可能なアドレスがない場合は、最初に使用可能なアドレスが使用されます。 この設定を使用して、スコープ内でアドレスを割り当てる方法の図については、図 12 : 割り当て先使用可能セットによるアドレス割り当てを参照してください。
優先アドレス - 割り当て	DHCP（有効または無効）	DHCP サーバー全体の優先アドレス割り当てを有効にする方法を提供します。(ただし、スコープ割り当て優先順位の設定は、この設定を上書きします。優先順位アドレス割り当てが有効で、スコープ割り当て優先順位属性が設定されていない場合、スコープサブネットアドレスが割り当て優先順位に使用されます。スコープ割り当て先利用可能が設定解除されると、優先順位アドレス割り当てが有効と見なされます。もちろん、このアドレス割り当ての全体的な制御を実行する場合、各スコープの実際の優先順位は、そのサブネット アドレスにのみ依存します。
同等の優先順位 - 最も利用可能	DHCP（有効または無効）	既定では、0 以外の割り当て優先順位が同じ 2 つ以上のスコープが検出されると、使用可能なIPアドレスが最も少ないスコープが使用され、新しいクライアントのアドレスが割り当てられます(そのクライアントが制限リストに含まれていない場合)。優先順位が最も高い場合に使用可能なスコープが有効で、2 つ以上のスコープの割り当て優先順位がゼロ以外の場合、使用可能なアドレスが最も多いスコープが使用され、新しいクライアントのアドレスが割り当てられます (そのクライアントが制限リストにない場合)。いずれの場合も、クライアントが制限リストに含まれている場合、同じ優先順位のスコープの中で、同じリスト内の他のクライアントを含むクライアントが常に使用されます。

スコープでのアドレスの割り当て

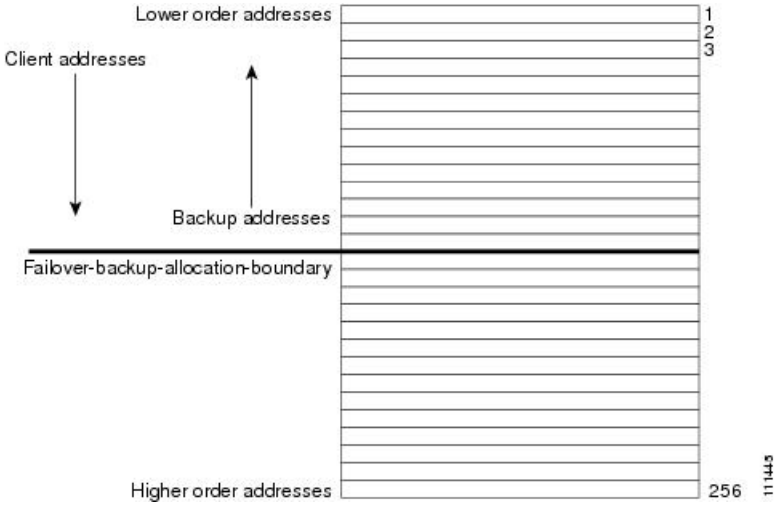
スコープ内から IP アドレスを割り当てようとする場合、DHCP サーバーの既定の動作では、使用可能なリースの一覧から、最も最近アクセスされたアドレスを最初に割り当てようとします。ただし、スコープ内のすべてのリースまたはすべてのリースのリスト表示、特定のリース

の要求 (nrcmd>**リース** addr)、リースの検索、リースの変更 (アクティブ化、非アクティブ化、または強制的に使用可能なリース) など、リースへのアクセスを必要とするすべての操作は、サーバーで利用可能なリースの一覧内のリースの順序に影響します。

リストの最後にリースする単一のリース場所で動作します。リースの一覧を作成すると、リースが数値順に配置され、最も低い番号のリースが使用可能なリストの最初に表示されます。リースクエリ要求など、サーバーがリースにアクセスする必要がある他の操作も、リースの順序に影響します。

したがって、一般的に、スコープ内のどの IP アドレスが特定の時点で割り当てられるかを予測する方法はありません。通常、これは難しい問題ではありませんが、より決定的な割り当て戦略が望まれる場合があります。完全に確定的なアドレス割り当て戦略を構成するには、スコープで最初に使用可能な割り当て属性を有効にします。これにより、最小の数値の使用可能なアドレスがDHCPクライアントに割り当てられます。したがって、最初のクライアントは最も低い範囲の最初のアドレスを取得し、2 番目のクライアントはその範囲の 2 番目のアドレスを取得します。これは下の画像に示されています。

図 *12*：割り当て先使用可能セットによるアドレス割り当て



この決定論的割り当て方法にはわずかなパフォーマンス コストが発生しますが、使用すべきでないほどではなく、必要でない場合は使用しないように十分な場合もあります。スコープがフェールオーバー関係にある状況でこの決定論的な割り当て方法を使用する場合、バックアップ サーバーに使用可能な IP アドレスを割り当てる方法がメイン サーバーに表示されます。既定では、スコープ内の最下位アドレスと最上位アドレスの中間にあるアドレスが、フェールオーバー-バックアップ-割り当て境界になります。バックアップ サーバーで使用可能なアドレスは、この境界から下方向に割り当てられます(その方向で使用可能なアドレスがある場合)。この境界の下にアドレスが使用できない場合は、境界の上にある最初のアドレスがバックアップ サーバーに使用されます。中間点とは異なるアドレス境界を設定する場合は、スコープのフェールオーバー-バックアップ-割り当て境界を構成できます。

必要以上に多くの IP アドレスを持つスコープを割り当てる場合は、決定的な割り当て戦略を使用し、割り当て先の使用可能な割り当てを構成します。アドレス空間を別のネットワークまたはサーバーに移動できるように、スコープ内の範囲を後で縮小することができます。非決定

的なアプローチでは、割り当てられたアドレスが範囲全体に散在し、スコープアドレスの半分を解放するようにDHCPクライアントを再構成するのは非常に困難です。ただし、割り当て先使用可能を構成すると、割り当てられたアドレスはスコープ範囲の低いクラスタに分類されます。その場合、そのアドレスを他の場所で使用できるように、必要ない範囲から範囲を削除する方が簡単でしょう。

スコープの編集

（注） スコープのサブネットに変更を加えることができるのは、現在のスコープ内、またはそれらのスコープのサブネットと同じ古いサブネットを持つ他のスコープに変更が適用される、変更と競合する予約または範囲がない場合のみです。

ローカル アドバンスド Web UI

ステップ 1 のスコープの作成 （138 ページ）説明に従って、スコープを作成します。

ステップ 2 DHCP サーバーをリロードします。

ステップ 3 [DHCPスコープの一覧/追加（List/Add DHCP Scopes）] ページの [スコープ（Scopes）] ペインでスコープ名をクリックして、[DHCPスコープの編集（Edit DHCP Scope）] ページを開きます(サーバーの再ロードが必要な場合は、ステータス メッセージが表示され、先に進む前に最初に再ロードする必要があります)。

ステップ 4 必要に応じて、フィールドまたは属性を変更します。スコープの名前を変更することもできます。

ステップ 5 スコープの埋め込みポリシーをスコープの埋め込みポリシーの設定 （148 ページ）編集するには、「」を参照してください。スコープのリースを一覧表示するには、「」リースの表示 （235 ページ）を参照してください。

ステップ 6 Save をクリックします。

ステップ 7 DHCP サーバーをリロードします。

CLI コマンド

スコープを作成したら、サーバー上のすべてのスコープのプロパティを調に入れ、使用(scopelistまたはscopelistnames scope listbrief 、 scope 、 scope get name show、 name属性) を使用します。次のアクションを実行します。

- 属性をリセットするには、scope 名前 set属性=値[属性=値..] を使用します。たとえば、scope name set name =new name を使用して範囲の名前をリセットできます。
- 属性を有効または無効にするにはscope、nameenable属性またはscope name disable 属性を使用します。

構文とscope属性の説明については、/docs ディレクトリの CLIGuide.html ファイルのコマンドを参照してください。

関連項目

段階的な同期モード （147 ページ）
スコープの埋め込みポリシーの設定 （148 ページ）
ネットワーク上の複数サブネットの設定 （149 ページ）
スコープの BOOTP の有効化と無効化 （150 ページ）
スコープの DHCP の無効化 （152 ページ）
スコープの非アクティブ化 （153 ページ）
スコープを更新専用に設定 （151 ページ）
スコープでの空きアドレス SNMP トラップの設定 （151 ページ）
アドレスを再利用しない場合のスコープの削除 （154 ページ）
アドレスを再利用しない場合のスコープの削除 （154 ページ）

段階的な同期モード

スコープの新しいスコープまたは変更は、ステージングモードと同期モードの2つのモードのいずれかになります。

- Staged: 既存のスコープに対する新しいスコープまたは変更はデータベースに書き込まれますが、DHCP サーバーが再ロードされるまで DHCP サーバーに伝達されません。
- Synchronous—ほとんどの新しいスコープとスコープの変更(削除を含む)は、直ちに DHCP サーバーに伝播されます(リロードは不要)。スコープの変更の一覧が変更できるわけではありません。たとえば、スコープのプライマリ サブネットの変更は許可されません (変更を反映するには再ロードが必要です)。さらに、リロードせずに、スコープ属性の変更のみを反映できます。たとえば、名前付きポリシーを変更するには、DHCP サーバーの再ロードが必要です。

ステージング モードでスコープを追加または変更した後、dhcp 編集モードを同期モードに変更すると、同期モードでの最初の変更は、そのスコープに対するすべての保留中の変更を適用します (同期モードで行われた変更だけでなく)。

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

現在の dhcp 編集モードを表示したり、dhcp 編集モードを変更したりするには、Web UI の上部にある[**設定**]ドロップダウン リストをクリックして、 を選択Session Settingsします。DHCP サーバーでスコープが最新の状態である場合、[**同期されたスコープの総数**] メッセージが [DHCP スコープの一覧/追加]ページ(詳細モード)に表示され、[**スコープ**名**の状態**: **同期**済み] メッセージが [DHCP スコープの編集]ページ(両方のモード)に表示されます。スコープが最新でない場合は、[**スコープ**名の**状態**: **必要な再読み込み**] メッセージが表示されます。

CLI コマンド

をsessionget使用dhcp-edit-modeして dhcp 編集モードを表示するか、 {sync sessionsetdhcp-edit-mode=を使用して DHCP 編集モードを設定する |staged}. DHCP サーバーと同期していないスコープを表示するには、 を使用しますscope report-staged-edits。次に例を示します。

```
nrcmd> scope report-staged-edits

100 Ok

example-scope: [reload-required]
```

サーバー上のスコープ数の取得

DHCP サーバーに関連付けられた作成されたスコープを表示できるため、Web UI でカウントを取得できます。

CLI コマンド

CLI を使用すると、 dhcp getScopeCount [vpn 名前|all]. VPN またはすべての VPN を指定できます。vpn 名前を省略すると、現在の VPN の数が返されます。フェールオーバー ペア名を指定すると、フェールオーバーペアのスコープとネットワークの合計が返されます。フェールオーバー ペア定義のマッチリストには明示的な VPN 設定が含まれているため、これらの数は現在の VPN だけに限定されません。

スコープを作成するには、scope 名前 createアドレスマスク[**テンプレート**=テンプレート名] [属性=値..]を使用します。各スコープは、そのネットワーク アドレスとマスクを識別する必要があります。スコープを作成すると、Cisco Prime Network レジストラーは、現在の仮想プライベート ネットワーク(VPN)session set current-vpnに配置されます。VPN は、スコープの作成時に設定した後は変更できません。

スコープのポリシーを設定するには、 scope name set policyを使用します。

スコープに IP アドレスの範囲を追加するには、scope名前addRangestart endを使用します。

スコープの埋め込みポリシーの設定

スコープを作成すると、Cisco Primeネットワーク レジストラーは、そのスコープに組み込まれたポリシーを自動的に作成します。ただし、組み込みポリシーには、有効または追加するまで、関連付けられたプロパティや DHCP オプションはありません。埋め込みポリシーは、スコープのルーターを定義する場合などに役立ちます。説明ポリシーのタイプ （200ページ）されているように、DHCP サーバーは、割り当てられた名前付きポリシーを参照する前に、スコープの埋め込みポリシーを参照します。

（注） スコープ ポリシーを削除すると、そのすべてのプロパティと属性が削除されます。

ローカル アドバンスド Web UI

ステップ 1 のスコープの作成 （138 ページ）説明に従って、スコープを作成します。

ステップ 2 [DHCP スコープの一覧/追加] ページの [スコープ] ウィンドウでスコープの名前をクリックして、[DHCP スコープの編集] ページを開きます。

ステップ 3 Create新Newしい埋Embeddedめ込みポリシーを作成するか、既存のポリシーが既に存在する場合は、[スコープの DHCP 埋め込みポリシーの編集] ページを開PolicyEditExistingEmbeddedPolicyきます。

ステップ 4 このページのフィールド、オプション、および属性を変更します。必要に応じて、属性を設定解除します。

ステップ 5 Save をクリックします。

CLI コマンド

最初にスコープを作成します。CLI では、scope-policyスコープ名を引数としてpolicy受け取ることを除いて、 と同じ構文を使用します。次に、次の手順を実行します。

- スコープに対して既に設定されている埋め込みプロパティ値があるかどうかをscope-policy判断し、scope-nameshowを使用します。
- 属性を有効または無効にする、scope-policy スコープ名 enable属性またはscope-policy スコープ名 disable属性を使用する。
- 属性を設定および設定解除し、scope-policy スコープ名 set属性=値[属性=値..] および scope-policy スコープ名 unset属性を使用します。
- ベンダーオプションの一覧、設定、および設定解除標準オプション定義セットの使用 （211 ページ）(「 」 を参照)

ネットワーク上の複数サブネットの設定

Cisco Prime Network レジストラーは、同じネットワーク セグメント上の複数の論理サブネットをサポートします。192.168.1.0/24 および 192.168.2.0/24 など、同じ物理ネットワーク上に複数の論理サブネットがある場合は、両方のプールからアドレスを提供するように構成できます。このようにアドレスをプールすることで、使用可能なリース数を増やすことができます。

2 つの論理サブネットを結合するには、2 つのスコープを作成し、1 つをプライマリに、もう 1 つをセカンダリに選択します。セカンダリ サブネットを構成すると、この物理ネットワーク上の新しいクライアントは、ラウンドロビンベースで 1 つまたは別のスコープからリースを取得します。

ローカル アドバンスド Web UI

ステップ 1 セカンダリ スコープを作成スコープの作成 （138 ページ）するスコープ (を参照) を作成します。

ステップ 2 [DHCPスコープの一覧/追加（List/Add DHCP Scopes）] ページの [スコープ（Scopes）] ペインでスコープ名をクリックして、[DHCPスコープの編集（Edit DHCP Scope）] ページを開きます。

ステップ 3 これをセカンダリ スコープにするには、[DHCP スコープの編集] ページの [プライマリ サブネット] 属性フィールドにプライマリ スコープのサブネットのネットワーク アドレスを入力します。

プライマリ サブネットは、プライマリ スコープのネットワーク アドレスに直接対応するのが一般的です。たとえば、192.168.1.0/24 ネットワークで作成された examplescope1 では、プライマリ サブネット =192.168.1.0/24 を使用して examplescope2 を関連付けます。(Cisco Prime Network レジストラーが、定義されたサブネットに関連するスコープがあることを検出した場合、マスク ビット定義は無視され、一致しない場合に備えてプライマリ スコープの定義を使用します)。ただし、プライマリ サブネットは、スコープが関連付けられていないサブネット アドレスである場合があります。

ステップ 4 Save をクリックします。

ステップ 5 サーバーを再起動または再ロードします。

CLI コマンド

セカンダリ スコープをプライマリ スコープに割り当てるscopeには、名前setprimary-subnet=値を使用してから、サーバーを再ロードします。

セカンダリ スコープを削除するには、 scope name unset primary-subnetを使用します。プライマリ サブネット属性を設定する場合は、スラッシュ表記を使用して、ネットワーク マスクのビット数を含めます。たとえば、ネットワーク 192.168.1.0 をマスク 255.255.255.0 で 192.168.1.0/24 として表します。マスク ビットは重要です。これらを省略すると、/32 マスク (単一 IP アドレス) が使用されます。

スコープの BOOTP の有効化と無効化

BOOTstrap プロトコル (BOOTP) は、ディスクレスコンピュータをロードするために作成されました。その後、ホストがインターネットを使用できるように、必要なすべての TCP/IP 情報を取得できるようにするために使用されました。ホストは BOOTP を使用して、ネットワーク上で要求をブロードキャストしたり、BOOTP サーバーから必要なデータを取得したりします。BOOTP サーバーは、着信要求をリッスンし、そのネットワーク上の BOOTP クライアントの構成データベースから応答を生成します。BOOTP は DHCP とは異なり、リースまたはリース期限の概念はありません。BOOTP サーバーが割り当てるすべてのアドレスは永続的です。

Cisco プライム ネットワーク レジストラー DHCP サーバーを BOOTP サーバーのように動作するように設定できます。さらに、BOOTP では通常静的アドレスの割り当てが必要ですが、アドレスを予約するか(静的割り当てを使用)、アドレスを動的に割り当てる (動的 BOOTP と呼ばれます) を選択することもできます。

BOOTP クライアントを移動または使用停止する必要がある場合は、リースの可用性を強制するだけで、リースを再利用できます。リースを強制的に使用可能にする （259ページ） を参照してください。

ローカル アドバンスド Web UI

[DHCP スコープの編集] ページの [BootP 設定] で、BOOTPの bootp 属性または動的ブート P の動的ブート属性を有効にします。デフォルトでは無効です。次に、Save をクリックします。

CLI コマンド

scope name enable bootp を使用して BOOTP を有効にし、scope name enable dynamic-bootp を使用して動的 BOOTP を有効にします。DHCP サーバーをリロードします(段階的な DHCP 編集モードの場合)。

スコープを更新専用に設定

既存のクライアントがリースを再取得することを許可するかどうかは制御できますが、新しいクライアントにリースを提供することはできません。更新のみのスコープでは、現在利用可能な IP アドレスを使用しているクライアントがリースを継続して使用することを許可する以外に、リースに関連付けられているクライアントは変更されません。

ローカル アドバンスド Web UI

[DHCP スコープの編集] ページの [その他の設定] で、更新専用属性を明示的に有効にします。次に、Save をクリックします。

CLI コマンド

scope名前enableをrenew-only使用して、スコープを更新専用に設定します。

スコープでの空きアドレス SNMP トラップの設定

SNMP トラップを設定して、トラップを有効にし、スコープの低しきい値と高しきい値を設定することで、予期しない空きアドレス イベントをキャプチャできます。スコープの代わりにネットワークと選択タグに基づいてトラップを設定することもできます。

しきい値を設定する場合は、下限値と高値の間の小さなオフセットを維持することをお勧めします。 Cisco プライムネットワーク レジストラー 11.0 管理ガイドオフセットは、たとえば、20% の低い値と 25% の高い値(プリセット値)の 5% までです。

これらの属性のサーバーとスコープの値を設定する方法のバリエーションを次に示します。

- 少なくとも 1 人の受信者が構成されている限り、サーバーの設定に基づいて、各スコープをトラップして解放アドレスの値をリセットします。
- スコープ レベルでトラップを無効にするか、スコープごとに異なる割合を指定します。
- サーバー上でトラップをグローバルに無効にしますが、スコープごとに有効にします。
- ネットワーク レベルまたは選択タグ レベルでトラップを設定します。

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 トラップ構成を作成するには、[**デプロイ**] メニューのサブTrapsメニューのDHCP下で [リスト/トラップ構成の追加] ページを開きます。

ステップ 2 [**トラップの追加**]アイコンをクリックし、トラップ設定の名前を入力scopeし、[**モード**]ドロップダウン リストから選択し、低しきい値と高しきい値を入力します(デフォルトでは、それぞれ20%と25%です)。Add AddrTrapConfig をクリックします。(必要に応じて、これらの値を編集するために戻ることができます。

ステップ 3 しきい値の設定を適用する作成されたスコープを編集します。[SNMP トラップ設定] の下の、フリーアドレス設定属性フィールドにトラップの名前を入力します。Save をクリックします。

ステップ 4 リージョン Web UI では、レプリカ トラップ構成をプルし、トラップ構成をローカル クラスターにプッシュするには、[リスト/トラップ構成の追加] ページを使用します。トラップ構成を再利用することもできます。

CLI コマンド

トラップaddr-trap構成を追加するには、namecreateを使用します。しきい値を設定するには、addr-trap 名前 setの方法を使用します (または、トラップの作成時にしきい値の設定を含めます)。次に例を示します。

```
nrcmd> addr-trap trap-1 create

nrcmd> addr-trap trap-1 set low-threshold

nrcmd> addr-trap trap-1 set high-threshold
```

フリー・アドレス・トラップを設定するには、scope 名前 set free-address-config=トラップ名を使用します。次に例を示します。

```
nrcmd> scope scope-1 set free-address-config=trap-1
```

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。

- **アドルトラップ**<名前|**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**]
- **アドルトラップ**<名前|**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**]
- **追加トラップ**名**再利用**クラスタリスト[-**レポートのみ**|-**レポート**]

スコープの DHCP の無効化

BOOTPのためだけに使用する場合は、スコープのDHCPを無効にできます。スコープのBOOTPの有効化と無効化 （150ページ）を参照してください。DHCP を無効にすることでスコープを

一時的に非アクティブにすることもできますが、BOOTP を有効にする場合は非アクティブ化が頻繁に使用されます。スコープの非アクティブ化 （153 ページ） を参照してください。

ローカル アドバンスド Web UI

[DHCP スコープの編集] ページの [BootP 設定] で dhcp属性を無効Saveにし、bootp属性を有効にして、 をクリックします。

CLI コマンド

DHCPscopeを無効にするには、名前disabledhcpを使用します。また、BOOTP を有効にして、サーバーをリロードする必要があります (段階的な dhcp 編集モードの場合)。

スコープの非アクティブ化

スコープ内のすべてのリースを一時的に非アクティブ化する場合があります。これを行うには、スコープの BOOTP と DHCP の両方を無効にする必要があります。

ローカル アドバンスド Web UI

[DHCP スコープの編集] ページの [その他の設定] で、非アクティブ化属性を明示的に有効にします。次に、Save をクリックします。

CLI コマンド

スコープscopeの BOOTP と DHCP を無効にするには、名前enabledeactivatedを使用します。DHCP サーバーをリロードします(段階的な DHCP 編集モードの場合)。

スコープの削除

注意 DHCP サーバーからスコープを削除するのは簡単ですが、注意が必要です。この操作を行うと、ネットワークの整合性が損なわれます。次のセクションで説明するように、アドレスを再利用するか、または使用しないかによって、サーバーからスコープを削除する方法はいくつかあります。

DHCP は、IETF によって定義されているように、特定の時間 (サーバー管理者によって定義される) クライアントにアドレスリースを提供します。その時間が経過するまで、クライアントはリースされたアドレスを自由に使用できます。サーバーは、リースを取り消して、クライアントがアドレスを使用するのを停止することはできません。したがって、DHCP サーバーからスコープを簡単に削除できますが、リースを取得したクライアントは、期限が切れるまで引き続き削除できます。これは、サーバーが更新の試行に応答しない場合でも、スコープが削除された場合に発生します。

削除したアドレスが何らかの方法で再利用されない場合、この方法では問題は発生しません。ただし、最後のリースの期限が切れる前に別のサーバーのアドレスが構成されている場合、2

つのクライアントが同じアドレスを使用する可能性があり、ネットワークが不安定になる可能性があります。

Cisco プライムネットワーク レジストラーは、削除されたスコープのリースを孤立したリースプールに移動します。スコープを作成する場合、孤立したリースは適切なスコープに関連付けられます。

アドレスを再利用しない場合のスコープの削除

アドレスを再利用しない場合は、スコープを削除できます。

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

スコープを再利用する予定がない場合は、[スコープの管理] ページまたは [DHCP スコープの一覧表示/追加] ページで、名前を選択した後、[スコープ] ウィンドウの[**スコープの削除**] アイコンをクリックし、削除を確認またはキャンセルします。

CLI コマンド

スコープ内のアドレスをすぐに再利用する予定がない場合は、scope名前deleteを使用して削除してください。

アドレスを再利用しない場合のスコープの削除

削除するスコープのアドレスを再利用する場合は、次の 2 つの方法があります。

- If you can afford to wait until all the leases in the scope expire—：スコープをサーバーから削除し、ポリシーで設定された最も長いリース時間が期限切れになるまで待機します。これにより、そのスコープのアドレスを使用しているクライアントが存在しません。その後、アドレスを安全に再利用できます。
- If you cannot afford to wait until all the leases in the scope expire：スコープを削除しません。代わりに、非アクティブ化します。スコープの非アクティブ化（153 ページ）を参照してください。削除されたスコープとは異なり、サーバーはすべてのクライアントの更新要求を拒否し、その多くが新しいリースを要求します。これにより、これらのクライアントは、削除されたスコープよりも非アクティブ化されたリースからより迅速に移動します。

  Windows のipconfigユーティリティを使用して、クライアントがそのリースを解放/releaseし、再取得/renewし、非アクティブ化されたリースからすぐにそのリースを移動させることができます。このユーティリティはクライアントコンピュータからしか発行できないので、何千ものリースが使用されているスコープでは実用的ではありません。ただし、Windows環境の最後の数個のクライアントをスコープ内の非アクティブ化リースから移動する場合に便利です。

DHCPv6 Addresses

Cisco Prime Network Registrar は、RFC 8415 に基づき 次の DHCP（DHCPv6）の IPv6 アドレスがサポートされています。

- Stateless autoconfiguration：DHCPv6 サーバーはアドレスを割り当てず、代わりに DNS サーバーデータなどの構成パラメータをクライアントに提供します。
- Stateful autoconfiguration：DHCPv6 サーバーは、非一時アドレスまたは一時アドレスを割り当て、クライアントに構成パラメータを提供します。
- Prefix Delegation：DHCPv6 サーバーはプレフィックスをクライアント（ルータ）に委任します。

（注） RFC 8415 は、以前の RFC である RFC 3315、RFC 3633、RFC 3736、RFC 4242、および RFC 7083 を組み込み、廃止しました。

DHCPv6 サービスは、次の機能を提供します。

- **アロケーション グループ**— 複数のプレフィックスを割り当ての観点から 1 つとして扱えるようにし、プレフィックスが使用される順序を制御できるようにします。
- **クライアント クラス化**:既知のクライアントまたはパケット ベースの式に基づいて、クライアントを分類し、プレフィックスを選択できます。
- DNS **更新**— DNS サーバーが (IPv4 経由で) DHCP アクティビティを更新します。
- **拡張機能**- C/C++ および Tcl 拡張機能を使用して、DHCP サーバーの処理を拡張します。
- **フェールオーバー**:要求しているクライアントにリースを提供できない場合に別のクライアントが引き継ぐように、DHCP フェールオーバー ペアを設定できます。
- LDAP:LDAPリポジトリ(Cisco Prime Network レジストラーの外部)でクライアント エントリルックアップを許可し、クライアントがクライアント予約を指定する場合があります。
- **リースクエリ**-リースクエリのサポートを提供します。
- **リンクとプレフィックス**— ネットワーク トポロジを定義する DHCPv4 ネットワークおよびスコープに似ています。各リンクには、1 つ以上のプレフィックスを付けることができます。
- **ポリシーとオプション**:リンク、プレフィックス、クライアントに属性とオプションを割り当てることができます。
- **プレフィックスの安定性**:クライアントは、ある CMTS から別の CMTS に移動したり、アドレス空間内で移動したりした場合でも、位置を変更した場合に委任されたプレフィックスを保持できます。適切なインフラストラクチャ サポート(CMTS、ルーター)を使用したプレフィックスの安定性により、別の委任されたプレフィックスを必要とせずに、サブスクライバを移動または移動できます。
- SNMP **トラップ**:プレフィックス内のリース数が一定の制限を超えた場合 (または一定の制限を下回った場合)、またはサーバーが重複アドレスを検出した場合など、イベントのトラップを生成します。
- **予約**—クライアントは、事前に決められたアドレスを受信できます。
- **統計収集およびロギング**-サーバー・アクティビティーのモニターを提供します。
- VPN **サポート**—複数のアドレス空間 (仮想プライベート ネットワーク) を提供します。

DHCPv6 サービスでは、サーバー・オペレーティング・システムが IPv6 をサポートし、システム上で IPv6 用に少なくとも 1 つのインターフェースを構成する必要があります。

IPv6 アドレス指定

IPv6 アドレスは 128 ビット長であり、コロンで区切られた 16 ビットの 16 進数のフィールドとして表現されます（:）。16 進数の A、B、C、D、E、および F は、大文字と小文字を区別しません。次に例を示します。

```
2001:db8:0000:0000:0000:0000:0000:0000
```

このアドレッシングに対するいくつかのショートカットは次のとおりです。

- フィールド内の先頭にある 0 は省略可能なため、09c0 は 9c0、そして 0000 は 0 と書き込むことができます。
- 連続した 0（任意の数）のフィールドは、2 つのコロン（::）で表すことができますが、アドレスで一度のみです。これは、二度以上使用すると、アドレス パーサーが 0 の各ブロックのサイズを識別できなくなるからです。これにより、アドレスの長さが減少します。たとえば、2001:db8:0000:0000:0000:0000:0000:0000:0000:0000:0000 に書き込むことができます。

  ```
  2001:db8::
  ```

リンク ローカル アドレスには、リンクに対して制限範囲があり、プレフィックス fe 80::/10 を使用します。ループバック アドレスにはアドレス ::1 があります。マルチキャスト アドレスには、ff00::/8 のプレフィックスが付きます（IPv6 にはブロードキャスト アドレスはありません）。

IPv6 の IPv4 に互換性のあるアドレスは、:: のプレフィックスが付く、IPv4 10 進数クアッド アドレスです。たとえば、IPv4 アドレスを 形式::c0a8:1e01::192.168.30.1で解釈して記述できます。

リンクとプレフィックスの決定

DHCPv6 サーバーは、DHCPv6 メッセージを受信すると、要求の処理に使用するリンクおよびプレフィックスを決定します。サーバー:

1. 送信元アドレスを検索します。
    1. クライアント メッセージがリレーされた場合、サーバーは、クライアントに最も近い (作業している) Relay-Forward メッセージから始まる最初の 0 以外のリンク アドレス フィールドに送信元アドレスを設定します。サーバーが送信元アドレスを見つけた場合は、**ステップ** 2に進みます。
    2. それ以外の場合、メッセージ・ソース・アドレスがリンク・ローカル・アドレスである場合、サーバーは、プレフィックスが存在するメッセージを受信したインターフェースの最初のアドレス (任意のアドレスの接頭部が見つかった場合は 0) にソース・アドレスを設定します。ステップ2に進みます。
    3. それ以外の場合、サーバーは送信元アドレスをメッセージ送信元アドレスに設定します。

(注) この動作は、拡張機能によって変更するか、クライアント/クライアント クラスの環境への追加属性を使用して、IPv6アドレスまたはプレフィックス名のリンク アドレスオーバーライド属性値を追加することによって変更できます。表 48 : 一般的な環境ディクショナリ データ項目 （462 ページ）を参照してください。

2. 送信元アドレスのプレフィックスを検索します。サーバーが送信元アドレスのプレフィックスを見つけられない場合、クライアントにサービスを提供できず、要求を破棄します。
3. プレフィックスのリンクを検索します。これは常に存在し、明示的に設定されたリンクか、プレフィックスアドレスに基づいて暗黙的に作成されたリンクのいずれかです。リンクはトポロジ リンクである必要があります (プレフィックス安定性 （158 ページ）を参照してください)。

これで、サーバーはクライアントリンクを決定できるようになったので、クライアント要求を処理できます。クライアント要求がステートフルかプレフィックス委任か、選択基準などの要因によって、サーバーはクライアント要求にサービスを提供するリンクに1つ以上のプレフィックスを使用する場合があります。

これは、DHCPv4 と DHCPv6 の間の違いの 1 つの領域です。DHCPv4 では、サーバーはクライアント要求にサービスを提供するために、ネットワークからスコープの 1 つだけを選択します。DHCPv6 では、サーバーはリンクのすべてのプレフィックスを使用できます。したがって、サーバーがクライアントにアドレスを割り当てたり、リンクの複数のプレフィックスからプレフィックスを委任したりすることができます（選択基準やその他条件が適用されます）。リンクの作成と編集 （167 ページ） を参照してください。

アドレスの生成

IPv6アドレスは128ビットアドレスです(IPv4の場合は32ビットのアドレスと比較されます)。ほとんどの場合、DHCPv6 サーバーは、これらのビットのうち 64 個のインターフェイス識別子 (EUI-64) 部分を割り当てます (RFC 4291 を参照)。クライアントの 64 ビット インターフェイス識別子または乱数ジェネレーターを使用して、アドレスを生成できます。インターフェイス識別子は、ステートレス自動設定がクライアントにアドレスを割り当てる方法をエミュレートします。残念ながら、その使用に関するプライバシーの懸念があり、クライアントのプレフィックスごとに 1 つのアドレスに制限されています。

デフォルトでは、Cisco Prime Network レジストラーは、RFC 4941 で説明されているアルゴリズムと同様のアルゴリズムを使用してアドレスを生成し、ランダムなインターフェイス識別子を生成します。これらのランダムインターフェイス識別子は、ユニバーサル/ローカルビットの値がゼロで、EUI-64ベースの識別子と区別されます。サーバーは、ランダムに生成されたインタフェース識別子を::0からスキップします。::ff にして、インフラストラクチャ デバイス (ルーターなど) に識別子を使用できるようにします。各プレフィックスに対して interface-identifier（使用可能な場合）を最初に割り当てるかどうかを設定できます（プレフィックスの allocation-algorithm 属性の interface-identifier フラグを使用）。（プレフィックスの作成と編集 （161 ページ） を参照）。インターフェイス識別子の使用を指定した場合、クライアントがア

ドレスを使用できない場合、またはクライアントがプレフィックスに複数のアドレスを要求する場合、サーバーはランダムに生成されたアドレスを使用する可能性があります。

サーバーは、プレフィックスが設定された範囲 (または範囲がない場合はプレフィックス アドレス) に基づいてアドレスを生成します。範囲プレフィックス長が 64 より短い場合、サーバーは 64 ビットのみを提供し、アドレス・インターフェース ID フィールドに入れられます。プレフィックス長が 64 より長い場合、サーバーはアドレスの残りのビットだけを提供します。したがって、/96 の範囲は、指定された範囲から 96 ビットを使用し、その後にクライアント インターフェイス識別子またはランダムに生成された値の 32 ビットを使用します。結果として得られるアドレスが使用できない場合 (別のクライアントまたは同じクライアントにリースされているが、別のバインディング上にリースされている場合など)、サーバーは別のアドレスを生成しようとします。このプロセスは最大 500 回まで繰り返されます。

DHCP フェイルオーバーが構成されている場合、サーバー生成のアドレスは、メインアドレス上で常に奇数アドレスになり、バックアップ上のアドレスも偶数になります。

（注） DHCP サーバーは、ランダムに生成されたインターフェイス識別子のみを::0 からの値に対してテストします。結果のアドレスではなく::ffにします。したがって、ランダムに生成されたアドレスは、xxxx :xxxx :xxxx :xxxx ::0を使用する可能性があります。xxxx : xxxx :xxxx :xxxx :xxxx ::ff アドレスは、プレフィックスの長さが /64 より長く、/64 境界を超えるプレフィックス ビットがすべてゼロである場合。

ヒント プレフィックスとプレフィックステンプレートに対して、追加のアドレス生成アルゴリズムから選択することもできます。「」を参照してくださいプレフィックステンプレートの作成と編集 （176 ページ）。

委任プレフィックスの生成

DHCPv6 サーバーは、委任されたプレフィックスを生成するときに、最適な最初の適合アルゴリズムを使用します。サーバーは、構成または要求された長さの最初の使用可能なプレフィックスを使用します。

DHCP フェールオーバーの場合、各サーバーは、使用可能な状態で委任されたプレフィックス リースのみを考慮します。サーバーが PARTNER-DOWN 状態の場合、サーバーは、一定時間の制限が過ぎた後に、他の使用可能な状態または保留中の状態でリースを使用することもできます。

プレフィックス安定性

プレフィックスの安定性 を使用すると、ネットワーク トポロジの接頭辞の安定性に関係なくプレフィックスの委任を。新しいリンク属性の種類は、リンクの種類を指定します。

リンクには、次の 3 種類があります。

- トポロジカル—トポロジリンク上のクライアントは、接続されているネットワークセグメントに基づいてリースが割り当てられます。
- 場所に依存しない:このリンク タイプは、ケーブルラボ DOCSIS 3.0 の概念である CMTS プレフィクスの安定性をサポートするために導入されています。CMTS のグループ内 (セントラル オフィスなど) 内のサービス プロバイダのロード バランシングおよび再構成イベントをサポートします。ロケーションに依存しないリンクでCMTS間で移動されたサブスクライバは、委任されたプレフィックスを保持できます。このリンクタイプは、単一のDHCPサーバー内での移動を可能にします。
- ユニバーサル:このリンク タイプは、加入者がネットワーク内の任意の場所で委任されたプレフィックスを保持できるように導入されます。このリンクの種類を使用するには、委任されたプレフィックスの管理割り当てと、クライアントまたはリース予約の使用が必要です。複数の DHCP サーバーに展開できます。

（注） プレフィックスの安定性の使用はルーティングに影響を与え、ルートをアドバタイズするためにリレー エージェント(つまり CMTS)からの適切なサポートが必要です。CMTS プレフィックスの安定性のために、これらは CMTS グループにローカライズされます。サービス プロバイダ ネットワーク全体でルートをアドバタイズする必要がある場合、ユニバーサル プレフィックスの安定性に対する影響は大きくなります。

CMTS プレフィックス安定性

ロケーションに依存しないリンクはCMTSプレフィクスの安定性のためのケーブルLabs DOCSIS 3.0の条件を実装します。すべてのプレフィックスが単一の DHCP サーバーによってサービスされる限り、CMTS プレフィックスの安定性は可能です。

特定の領域にCMTSプレフィックスの安定性を導入する場合は、次の手順を実行する必要があります。

- 既存のリンクを変更して、グループ内のすべてのリンクで同じリンク グループ名を指定します。各 CMTS(または CMTS バンドル)は個別のリンクを持っていますが、CMTS プレフィックスの安定性が望まれる領域内のこれらのリンクはすべて、同じリンク グループの一部にする必要があります。
- 場所に依存しないとしてフラグが設定され、このリンク グループの一部として作成された新しいリンクを作成します。この場所に依存しないリンクの下に1つ以上のプレフィックス委任プレフィックスを作成または移動します。
- 不要になった既存のリンクからプレフィックスの委任プレフィックスを削除します。ステートフル プレフィックス (dhcp タイプの dhcp) は削除しないでください。

（注） グループ内で使用できる場所に依存しないリンクは 1 つだけです。

クライアント要求を受信すると、サーバーは、最も長く一致するプレフィックスをチェックし、そのプレフィックスのリンクを使用して、リンクを検索します。ただし、このトポロジリンクがリンク グループの一部であり、そのグループにロケーションに依存しないリンクがある場合、場所に依存しないリンクのプレフィックスが最初にチェックされ、クライアントが要求する可能性のあるリースが確認されます。このロケーションに依存しないリンクからリースが利用できない場合にのみ、トポロジカルリンクが使用されます。これは、クライアントが要求したバインディングごとに使用されます。

リースメカニズム(リースまたはクライアント予約、最初の最適な状態、または生成/提供された拡張) は、CMTS グループにサービスを提供する単一サーバー内でのみリースが認識されるため、CMTS プレフィックスの安定性と共に使用できます。

ユニバーサル プレフィックス安定性

ユニバーサルプレフィックスの安定性を使用すると、接続先に関係なく、委任されたプレフィックスを保持できます。この機能を使用するには、デリゲートされたプレフィックスの予約を構成する必要があります。クライアントとリースの予約を使用できます。

クライアント予約では、DHCP サーバーが動的にアクセスする中央 LDAP リポジトリで、委任されたプレフィックスを指定できますクライアント予約の使用 （244 ページ）(を参照)。リース予約は、CCM リージョナル サーバー上で一元管理され、ユニバーサル リンクを使用して各ローカル DHCP にプッシュされます。リース予約を使用する場合、予約の完全な一覧は各サーバーでレプリケートされるため、大規模な展開ではクライアントの予約を検討する必要があります。

（注） 特定の VPN アドレス空間にユニバーサル リンクを 1 つだけ持つことができます。

ユニバーサル リンク タイプでリンクが設定されている場合、クライアントにリースを割り当てようとすると、そのリンクのプレフィックスが最初に考慮されます。リースが使用できない場合は、リンク グループ (存在する場合) のロケーション非依存リンク タイプのプレフィックスが使用されます。最後に、トポロジ リンクのプレフィックスが使用されます。

（注） CMTS プレフィックスの安定性とユニバーサルプレフィックスの安定性の両方を同時に有効にできますが、加入者に適用されるのは 1 つだけです。

プレフィックス割り当てグループ

プレフィックスアロケーション グループを使用すると、クライアントに対して複数のリース割り当てが行われないように複数のプレフィックスを定義し、プレフィックスを使用するプレフィックス割を制御できます。この動作を指定するために、割り当てグループおよび割り当てグループの優先順位属性が導入されます。

同じ割り当てグループ名を持つリンク上のすべてのプレフィックスは、その割り当てグループに属します。割り当てグループ名のないプレフィックスは、独自の割り当てグループ内にあります。バインディングごとに1つのリースが、同じアロケーション グループ内のすべてのプレフィックスに割り当てられます。

割り当てグループの優先順位設定は、使用するプレフィックスを制御します。数値が小さい場合は、優先順位が最も低い0(デフォルト)を除き、優先順位が高くなります。同じ優先順位のプレフィックスは、アクティブなリース数によって順序付けされ、カウント数が最も小さいプレフィックスが最も高い優先順位を持ちます。

（注） 割り振りグループ名は、リンクに固有のものです。異なるリンクで同じアロケーション グループ名を再利用できます。

クライアントがアロケーション グループ プレフィックスから取得できるリースの数を制御するには、DHCPポリシーのバインディングごとの最大リース属性を設定します。たとえば、バインディングあたりの最大リース数を1に設定すると、クライアントはアロケーション グループ プレフィックスからリースを 1 つだけ取得できます。さらに、同じ割り当てグループプレフィックスから複数のリースが既に割り当てられている場合、追加のリースは取り消されます(通常、最も古いリースは取り消されます)。

プレフィックスとリンクの設定

DHCPv6プレフィックスおよびリンクを直接設定することも、最初にプレフィックスまたはリンク テンプレートを作成することもできます。次のサブセクションを参照してください。

- プレフィックス テンプレートの作成と編集 （176 ページ）
- プレフィックスの作成と編集 （161 ページ）
- プレフィックスのアドレス使用率の表示 （129 ページ）

プレフィックスの作成と編集

プレフィックスを直接作成できます(また、必要に応じて既存のテンプレートを適用プレフィックス テンプレートの作成と編集 （176 ページ）することもできます)。設定できるプレフィックス属性は次のとおりです。

- name：このプレフィックスに名前を割り当てます。
- vpn-id：プレフィックスを含む VPN。
- address：IPv6アドレスの上位ビットを使用してインターフェイスが属しているプレフィックス（サブネット）。
- leasequery-send-all：プレフィックスについての説明。
- 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。
- 親—DHCP でプレフィックスを使用しない。ただし、子プレフィックスをグループ化するコンテナ オブジェクトとして使用します。親プレフィックスは、Web UI の IPv6 アドレス空間リストにのみ表示され、プレフィックスリストには表示されません。

- owner：プレフィックスの所有者。
- region：プレフィックスのリージョン。
- 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 （307 ページ）を参照してください）。
- range：サーバーがアドレス割り当てのプレフィックスを設定するために使用できるサブ範囲。使用されるプレフィックスは、dhcp-type属性に設定されている値によって異なります。設定されていない場合、プレフィックスアドレスが適用されます。この値は、割り当てに使用できるアドレスまたはプレフィックスの範囲を制限するために、プレフィックスアドレスより長いプレフィックスを指定できます。（詳細については、リンクとプレフィックス （5 ページ）を参照してください）。
- link：プレフィックス（サブネット）に関連付けられたリンク。単一のリンク上にあるプレフィックスをグループ化するために使用されます。
- 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。
- 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-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 off）：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、を拡張ポイントの使用 （433 ページ）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。
- restrict-to-admin-allocation：プレフィックスが次に使用可能なアドレスを割り当てる管理要求に制限されるかどうかを制御します。設定されている場合、サーバーは、クライアントに事前に割り当てられている場合にのみ、このプレフィックスからのアドレスを持つクライアントに応答します。
- 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にのみ作成します。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。

  ヒント　最大リースの値を予想される最大値に設定して、SNMP アドレス トラップが意味のある結果を返すようにします。

- 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。

- 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" の有効期限属性については、説明をプレフィックステンプレートの作成と編集（176 ページ）参照してください。
- free-address-config：このプレフィックスで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を識別します。構成されていない場合、サーバーは親リンクのフリー・アドレス構成属性値を探します。この属性が構成されていない場合、サーバーはv6-default フリー・アドレス設定属性を調べられます。
- 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。
- 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。
- allocation-group：この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。デフォルト値はゼロです。
- range-end：リースの割り当てに使用されるプレフィックスアドレス範囲内のエンドアドレスを指定します（これがDHCP タイプのプレフィックスである場合）。設定されていない場合、プレフィックスアドレス範囲の最後に使用可能なアドレスが終了アドレスとして使用されます(range-startがプレフィックスとして指定されている場合を除き、range-startで指定された接頭語の最後のアドレスが使用されます)。
- range-start：リースの割り当てに使用されるプレフィックスアドレス範囲内の開始アドレスか、またはこれがDHCPタイプのプレフィックスの場合は範囲として使用されるプレフィックス （この場合は range-end を指定しない）を指定します。設定されていない場合、プレフィックス アドレス範囲の最初の使用可能なアドレスが開始アドレスとして使用されます。

  範囲開始と範囲終了を使用すると、顧客は、ランダムなアドレスを割り当てる際にサーバーが使用するアドレスの範囲を制限できます。予約や拡張が提供するアドレスには影響しません。これらの属性は、完全に指定されたIPv6アドレスまたは非プレフィックスビット(プレフィックスの範囲またはアドレス属性に基づく)セットを持つIPv6アドレスです。たとえば、::1000 の場合、プレフィックスの範囲/アドレスが /96 以下です。

注
- 範囲開始も範囲終了も指定されていない場合、10.0より前の動作はランダムなアドレス割り当てに関して保持されます。
- 範囲開始または範囲終了のどちらかを指定した場合、インターフェースID割り振りアルゴリズムが指定されている場合は、使用不可になります。

• embedded-policy：プレフィックスに埋め込まれたポリシー。

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 [**デザイン**] メニューのPrefixesサブメニューDHCPv6の下で選択します。[DHCP v6 プレフィックスの一覧表示/追加] ページには、既存のプレフィックスが表示されます。

プレフィックスを作成するには、次の手順を実行します。

1. 現在の VPN 以外で作成する場合は、Web UI の上部にある [**設定**]ドロップダウン リストの VPN サブメニューから VPN を選択します。
2. [プレフィックス] ウィンドウの [**プレフィックスの追加**] アイコンをクリックし、プレフィックス名とアドレスを入力して、ドロップダウン リストからプレフィックスの長さを選択します。
3. プレフィックスのアドレス範囲を指定する場合は、サブネット アドレスを入力し、プレフィックスの長さを選択します。
4. DHCP タイプを選択します(このセクションの上部にある属性の説明を参照)。デフォルトは DHCP です。
5. 事前設定済みのプレフィックス テンプレートを適用する場合は、ドロップダウン リストから選択します。(適用されたテンプレートの属性値は、接頭辞に設定された値を上書きします)。
6. をAddIPv6クリックPrefixすると、リストに接頭辞が追加されます。
7. DHCP サーバーをリロードします。[DHCPv6 プレフィックスの一覧/追加] ページに戻ると、同期されるプレフィックスの数を示すメッセージが表示されます。

ステップ 2 プレフィックスから逆引きゾーンを作成するには、[逆方向の領域] タブをクリックします。このタブで、ゾーン テンプレートを選択し、 をReportクリックしますRun。

ステップ 3 プレフィックスを作成すると、[リース] タブをクリックして、プレフィックスのリースを表示および管理できます。[リース]タブで、クライアントルックアップキーのリースを表示し、名前をクリックして各リースを個別に管理できます。

ステップ 4 [予約] タブをクリックすると、プレフィックスの予約を表示および管理できます。各予約 IP アドレスとルックアップ キーを追加し、ルックアップ キーが文字列かバイナリAdd Reservationかを指定して、 をクリックします。

ステップ 5 プレフィックスを編集するには、[プレフィックス]ペインで名前をクリックします。[プレフィックスの編集]ページで、プレフィックス属性を編集するか、グループにプレフィックスを割り当てて優先順位を設定するか、新しいポリシーを作成するか、既存の埋め込みポリシーを編集します。

グループにプレフィックスを割り当て、優先順位を設定するには、次の手順に従います。

1. 配賦グループ属性フィールドにグループの名前を入力します。
2. [配賦グループ優先順位属性] フィールドに優先順位値を入力します。ここで値を入力しない場合は、既定値(0)が割り当てられ、このプレフィックスはグループ内で最も低い優先順位になります。

   これらの属性は、詳細モードのアロケーション グループにありますプレフィックス割り当てグループ（160 ページ）(を参照)。

埋め込みポリシーを管理するには

1. またはCreateEmbedded、[NewPolicyプレフィックスのEditDHCP 埋め込みポリシーの編集] ページを開きます。 Existing Embedded Policy
2. 埋め込みポリシーのプロパティDHCPv6 ポリシー階層 （202 ページ）を変更します (「」を参照)。
3. Modify Embedded Policy をクリックします。次回[DHCPv6 プレフィックスの編集]ページが表示されたら、そのプレフィックスの埋め込みポリシーを編集できます。
4. Save をクリックします。

ステップ 6 地域 Web UI では、プレフィックスをローカル クラスターにプッシュし、[DHCPv6 プレフィックスのリスト/追加] ページでプレフィックスを再利用できます。

- プレフィックスをプッシュするには、目的のプレフィックスを選択しPush 、クリックして[IPv6 プレフィックスをプッシュ]ページを開きます。プレフィックスをプッシュするクラスタテンプレートまたはプレフィックステンプレートを選択し、 をクリックしますPush Prefix。プレフィックスがプッシュされると、プレフィックスの予約はプレフィックスでプッシュされます。また、プレフィックスがリンク上にある場合、親プレフィックスがローカル クラスタに存在しない場合は、そのプレフィックスがプッシュされます。
- プレフィックスを再利用するには、目的のプレフィックスを選択し、Reclaim クリックして [IPv6 プレフィックスの再利用]ページを開きます。プレフィックスを再利用するクラスタテンプレートまたはプレフィックステンプレートを選択し、 をクリックしますReclaim Prefix。プレフィックスが再利用されると、アクティブなリースがない場合、またはforceオプションが指定されている場合は、予約はプレフィックス付きで削除されます。それ以外の場合、プレフィックスは非アクティブになります。

  （注） プレフィックスがユニバーサル リンクにある場合、そのプレフィックスは複数のクラスタにプッシュすることができ、ローカルでの変更は次のサーバーのリロードまで有効になりません。

CLI コマンド

prefix name create ipv6address/length を使用します。(このprefixコマンドは、以前のリリースdhcp-prefixのコマンドのシノニムです。DHCP サーバーをリロードします。次に例を示します。

```
nrcmd> prefix example-prefix create 2001:0db8::/32 [attribute=value]
nrcmd> dhcp reload
```

プレフィックスの作成中にプレフィックステンプレートを適用するには、 prefix name create ipv6address/length template=name を使用します。既存のプレフィックス定義にテンプレートを適用するには、prefix 名前 applyTemplateテンプレート名を使用します。次に例を示します。

```
nrcmd> prefix example-prefix create 2001:0db8::/64 template=preftemp-1

nrcmd> prefix example-prefix applyTemplate template=preftemp-1

nrcmd> dhcp reload
```

上記の属性は通常の方法で設定および有効化できます。prefix name addReservation ipv6address/length lookup-key [–blob | –string] を使用して予約を追加します。prefix名前listLeases を使用してリースを一覧表示します。

ヒント　追加のIPv6 リースの再設定 （258 ページ）構文については、 を参照してください。

dhcp [getPrefixCount vpn 名前|all]. VPN またはすべての VPN を指定できます。vpn 名前を省略すると、現在の VPN の数が返されます。

地域クラスターに接続すると、次のプッシュ・コマンドと再要求コマンドを使用できます。プッシュの場合、通常は1つのクラスターまたはフェールオーバーペアのみを指定でき、フェールオーバー ペアのクラスタを再利用できません。ただし、プレフィックスがユニバーサル リンクにある場合は、クラスターとフェールオーバー ペアの一覧を指定できます。

- **プレフィックス名プッシュ**クラスター/フェールオーバーペア リスト[-template=テンプレート名] [-omitparents] [-omitchildren] [-**レポート**]
- **プレフィックス名の再利用**[クラスター/フェールオーバー ペア リスト] [-force] [-omitchildren] [-report-only] [-report-only]

リンクの作成と編集

リンクを直接作成できます。リンクに設定できる属性は次のとおりです。

- name：ユーザーがリンクに割り当てた名前。
- vpn-id：リンクを含む VPN。
- description：リンクの説明テキスト。
- 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：リンクの所有者。
- region：このリンクのリージョン。

- free-address-config：このプレフィックスで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を識別します。構成されていない場合、サーバーはv6-default フリー アドレス設定属性を調べています。
- interface：このリンクに関連付けられたルーターインターフェイス。
- 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：クライアントに応答する際に使用される、単一で特定のリンク オブジェクト内に埋め込まれているポリシー。

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 メニューから Design 、DHCPv6サブメニューの下の を選択Linksします。[DHCP v6 リンクの一覧表示/追加] ページには、既存のリンクが表示されます。

ステップ 2 リンクを追加するには、[リンク] ウィンドウの [**リンクの追加**] アイコンをクリックします。

ステップ 3 リンクの名前を入力します。

ステップ 4 リンクが接頭部の安定性の場合は、リンクタイプ (type) を選択し、リンクグループ名 (group-name) を指定します。リンクの種類は、既定ではトポロジ的です。これらの属性は、[DHCP v6 リンク テンプレートの編集] ページの [プレフィックスの安定性プレフィックス安定性 （158 ページ）] 領域でも確認できます (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)。

（注） リンク グループにはロケーションに依存しないリンクが 1 つ、VPN アドレス空間に 1 つのユニバーサル リンクしか設定できません。また、ユニバーサルタイプのリンクをリンクグループに割り当てることはできません。

ステップ 5 [リンクの追加（Add Link）] をクリックします。

ステップ 6 新しいリンクの [リンクの編集] ページで、[使用可能] フィールドから [選択済] フィールドにリンクを移動して、リンクの定義済みのプレフィックスを選択します。

ステップ 7 リンクに新しいプレフィックスを追加するには、各プレフィックス名とアドレスをページの下部に入力し、範囲を指定し、DHCP タイプとテンプレートを選択しますApply Prefix(必要な場合)。

ステップ 8 Save をクリックします。

ステップ 9 地域 Web UI では、ローカル クラスターへのリンクをプッシュし、[DHCP v6 リンクの編集] ページでリンクを再利用し、[DHCP v6 リンクの一覧/追加] ページの [レプリカ IPv6 アドレス空間] をプルできます。

- リンクをプッシュするには、目的のリンクを選択しPush、(ページの上部にある)をクリックして、プッシュリンクページを開きます。リンクをプッシュするクラスターまたはリンクテンプレートを選択し、をクリックしますPush Link。リンクがプッシュされると、リンク上のすべてのプレフィックスと、プレフィックス上のすべての予約もプッシュされます。
- リンクを再利用するには、目的のリンクを選択し、(ページの上部にある)をクリックReclaimして[リンクの再利用]ページを開きます。リンクを再利用するクラスタまたはリンクテンプレートを選択し、をクリックしますReclaim Link。リンクが再利用されると、アクティブなリースがない場合、予約、プレフィックス、およびリンクはローカル クラスターから削除されます。アクティブなリースが見つかっ

た場合、プレフィックスは無効になります。forceオプションを使用すると、アクティブなリースがある場合にリンクとそのプレフィックスを削除できます。

（注） ユニバーサル リンクのみを複数のクラスターにプッシュできます。

- レプリカ IPv6 アドレススペースをプルするには、左側のリンクペインの上部にある[**データのプル**]アイコンをクリックして、[プル レプリカ IPv6 アドレススペースの選択]を開きます。データ同期モード(更新、完了、または正確) を選択Reportし、 をクリックします。

ローカルでの変更は、次のサーバーの再ロードまで有効になりません。

CLI コマンド

link 名前 create を使用します。（link コマンドは、以前のリリースからの dhcp-link コマンドと同義です）。次に例を示します。

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

リンクの作成時にリンクテンプレートを適用するには、link名前名template-root-prefix=[address]を使用createtemplate=し、テンプレートが複数のプレフィックスを作成できる場合は、テンプレートルートプレフィックスを指定します。既存のリンク定義にテンプレートを適用するには、link 名前 applyTemplateテンプレート名[テンプレートルートプレフィックス] を使用します。

上記の属性は通常の方法で設定および有効化でき、リンクの表示とリスト表示を行うことができます。リンクに関連付けられたプレフィックスまたはプレフィックス名を一覧表示するにはlink、名前listPrefixesまたはlink名前listPrefixNamesを使用します。

リージョナル クラスタに接続する場合は、下記の push コマンドや reclaim コマンドを使用することができます。プッシュの場合、通常は1つのクラスターまたはフェールオーバー ペアのみを指定でき、クラスターまたはフェールオーバーペアを再利用できません。ただし、リンクがユニバーサル リンクの場合は、クラスターとフェールオーバー ペアの一覧を指定できます。

- **リンク**名**プッシュ**クラスター/ フェールオーバーペア リスト[-template= プレフィックステンプレート名] [-omitparents] [-omitchildren] [-report]
- **リンク**名**の再利用**[クラスター/フェールオーバー ペア リスト] [-force] [-report]]

DHCP ネットワークの管理

スコープを作成する場合は、サブネットとマスクに基づいてネットワークも作成します。スコープは同じサブネットを共有できるため、関連付けられたネットワークとスコープを表示すると便利です。これらのネットワークの管理は、ローカル クラスタ機能のみです。また、作成されたネットワークの名前を編集することもできます。

関連項目

ネットワークの一覧 （170 ページ）
ネットワークの編集 （170 ページ）

ネットワークの一覧

[ネットワークの一覧]ページでは、スコープによって作成されたネットワークを一覧表示し、ネットワークが関連付けるスコープを決定できます。ネットワークは名前でリストされ、Web UI はサブネットとマスクから作成します。このページでは、ネットワークを展開したり折りたたんだりして、関連するスコープを表示または非表示にできます。

基本モードで、[**設計**] メニューから Networks[ ] から [ ] をDHCPv4 選択して [DHCP ネットワーク ツリー] ページを開きます。このページでは、次の作業を行うことができます。

- - ネットワークは名前のアルファベット順に表示されます。 List the networks サブネットと割り当てられた選択タグを識別できます。ネットワークの横にある+プラス記号()をクリックすると、関連するスコープが表示されます。

  すべてのネットワーク ビューを展開するにはExpand All、 をクリックします。ネットワークビューをすべて折りたたんでネットワーク名だけを表示するには、Collapse Allをクリックします。

- **ネットワーク名の編集**-ネットワーク名をクリックします。ネットワークの編集 （170 ページ）を参照してください。

DHCPv6 アドレス空間内のネットワークを表示するには、**設計** > DHCPv6メニューから**ネットワーク**を選択して、DHCPv6 ネットワーク ツリー ページを開きます。このページでは、「DHCPv6 リンクのリスト/追加」ページと同様に、テンプレートおよびテンプレート・ルート接頭部を使用して DHCPv6 リンクを追加できます。リンクを追加すると、[DHCPv6 リンクの追加]ページが開きます。リンクを作成した後は、編集用の [DHCPv6 ネットワークの表示]ページでリンクを選択できます。

ヒント DHCP v6 ネットワーク ツリー ページを使用して、リンクをプッシュおよび再利用できます。目的のリンクの**プッシュ**または**再利用**アイコンをクリックします。詳細については、リンクの作成と編集 （167 ページ）の項を参照してください。

ネットワークの編集

ネットワーク名を編集できます。元の名前は、スコープで指定されたサブネットとマスクに基づいています。この名前は任意の説明文字列に変更できます。

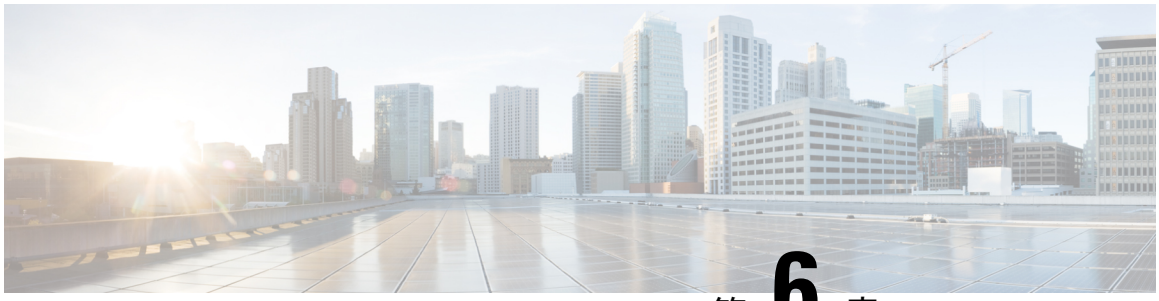
[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

ステップ 1 [**デザイン**] メニューから、[DHCPv4]サブメニューから[**ネットワーク**] を選択するか、DHCPv6サブメニューから[**ネットワーク**] を選択して、[DHCP ネットワーク ツリー] ページ (DHCP v4) または DHCP v6 ネットワーク ツリー ページ (DHCP v6) を開きます。

DHCPv6 の場合、DHCP v6 ネットワークページはネットワークを作成するためのものです。ネットワークの名前を入力し、必要に応じてテンプレートを選択し、テンプレートルートプレフィックス名を入力して、[**リンク**を追加ネットワークの一覧 （170 ページ）]をクリックします(を参照)。

ネットワークを編集する場合は、編集するネットワークの名前をクリックします。[DHCP v6 リンクの編集] ページが開きます。

ステップ 2 [保存（Save）] をクリックします。



第 6 章

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

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 （183 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 レジストラー 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 （183 ページ）
追加のスコープ テンプレート属性 （174 ページ）
スコープ テンプレートの編集 （174 ページ）
スコープ テンプレートのスコープへの適用 （174 ページ）
スコープ テンプレートの複製 （175 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スコープ テンプレートの作成と適用 （173 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 （183 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スコープが作成されます。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 （188 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コンテ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 （433 ページ）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にのみ作成します。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 （307 ページ）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（188 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[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| 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（192 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 （192 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 （158 ページ）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 （433 ページ）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 *17*：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: **(concat "ISP-" subnet)** --> ISP-192.168.50.0/24 **(concat subnet "-" (+ template.ping-timeout 10))** --> 192.168.50.0/24-110 **(concat "ISP-" subnet "-" (+ template.ping-timeout 10))** --> ISP-192.168.50.0/24-110 スコープ名の式の例 （187 ページ）も参照してください。
Create Option Operation	

式関数	説明

(create-option opt val)	[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。
	
	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。
	
	次に、例を示します。
	
	`(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))`
	
	埋め込みポリシー オプション式の例 （188 ページ）も参照してください。

Create Vendor Option Operation	

(create-vendor-option set-name opt val)	[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。
	
	次に、例を示します。
	
	`(list (create-option "routers" (create-ipaddr subnet 1))`
	
	`(create-vendor-option "dhcp-cablelabs-config" 125`
	
	`(concat "(tftp-servers 2 " (create-ipaddr subnet 2)")")))`

Create Range Operation	

式関数	説明
(create-range start end)	[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で、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。 例 (サブネット= 192.168.50.0/26): **(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 範囲の式の例 （187 ページ）も参照してください。
Create IP Operation	
(create-ipaddrネットホスト)	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。 例： **(create-ipaddr subnet 4)**
List Operation	
(list oper1 ... opern)	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ません。 例： **(list (create-option "routers" "10.10.10.1")** **(create-option "domain-name" "example.com"))** **(list (create-range 1 5) (create-range 10 20))**

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すexpression必要があります
- Range —IP アドレスを返Expressionす必要がある
- 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と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 （433 ページ）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 *18*：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 *19*：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、create-v6-optionまたはcreate-prefix-range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)	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。 範囲値 - プレフィックス長の増加。 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。 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。 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、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。 例： **(create-prefix-range 32 0x1)**
Create Option Operation	

式の演算子	説明
(create-option opt val)	オプション expr属性で使用される DHCPv6 オプションを作成します。opt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。val 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。 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。 例： **(list (create-option "dns-servers" (create-prefix-addr prefix "::2"))** **(create-option "domain-list" "sales.example.com,example.com"))** （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。
Create Vendor Option Operator	

式の演算子	説明

(create-vendor-option set-name opt val)	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。
	例：
	`(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))))"))`
	（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。

（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 （433ページ）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 21 : 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 19 : 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 *20* : 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 *21* : リンク テンプレート式演算子

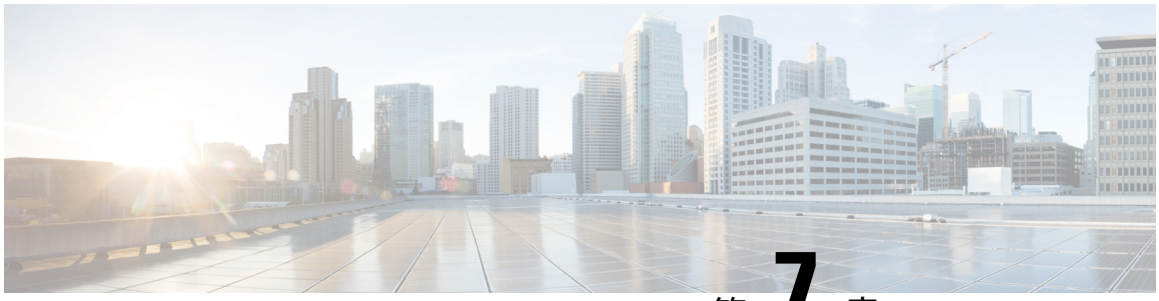
式の演算子	説明
Arithmetic (符号なし整数Operators引数のみ)	
(+ arg1 arg2)	(+ 2 3) などの 2 つの引数値を加算します。

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)	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create-v6-option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。 例： **(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)))**
Create Prefix Operator	
(create-prefix template prefix)	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)。 prefix引数は、プレフィックス名にすることもできますが、create-prefix-addr演算子のcreate-prefix-range値も指定できます。この関数をlist使用して、複数の操作を結合できます。 例： **(create-prefix "cm-prefix" (create-prefix-range 32 0x1))**
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)	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。 範囲値 - プレフィックス長の増加。 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。 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。 サイズとnは 0 より大きくなければなりません。 nはサイズ以下にする必要があり、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。 例： **(create-prefix-range 32 0x1)**
Create Option Operator	

式の演算子	説明
(create-option opt val)	オプション expr属性で使用される DHCPv6 オプションを作成します。opt 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。val 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。 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。 例： **(list (create-option "dns-servers" (create-prefix-addr prefix "::2"))** **(create-option "domain-list" "sales.example.com,example.com"))** （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。ただし、(作成オプション)を使用することをお勧めします。
Create Vendor Option Operation	

(create-vendor-option set-name opt val)	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・オプションのオプション定義セットを指定します。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。
	次に、例を示します。
	`(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))))"))`
	（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。ただし、使用することをお勧めします (create-vendor-option)。



第 7 章

ポリシーとオプションの管理

この章では、DHCP ポリシーとオプションを設定する方法について説明します。クライアントがアドレス割り当てに DHCP を使用する前に、少なくとも 1 つの DHCPv4 スコープ (動的アドレス プール) または DHCPv6 プレフィックスをサーバーに追加する必要があります。ポリシーの属性とオプションは、スコープまたはプレフィックスに割り当てられます。

DHCP ポリシーの設定

すべての DHCPv4 スコープまたは DHCPv6 プレフィックスには、定義された 1 つ以上のポリシーが必要です。ポリシーは、DHCP オプションと呼ばれるリース期間、ゲートウェイ ルーター、およびその他の構成パラメータを定義します。ポリシーは、ポリシーを 1 回定義するだけで済むため、スコープまたはプレフィックスが複数ある場合に特に便利です。

このセクションでは、特定の属性とオプション定義を持つ名前付きポリシーを定義する方法、またはシステムのデフォルトポリシーまたは組み込みポリシーを使用する方法について説明します。

関連項目

ポリシーのタイプ （200 ページ）

DHCPv4 ポリシー階層 （202 ページ）

DHCPv6 ポリシーの設定

DHCPv6 ポリシー属性は、次のように編集できます。

- affinity-period：リースアフィニティ （233 ページ）を参照してください（プリセット値なし）。
- allow-non-temporary-addresses：非一時（IA_NA）アドレスを要求する DHCPv6 クライアントを有効または無効にします（プリセット値は有効）。
- allow-rapid-commit：高速コミットが有効な状態で、クライアントはコミットされたアドレスに関する情報を（要求時に）受け取ります。その後で、クライアント要求で迅速にコミットされます（プリセット値は有効）。Rapid Commit は、1 台の DHCP サーバーがクライアントにサービスを提供している場合にのみ使用します。(このDHCPv6 ポリシー階層 （202 ページ）属性の特別な処理については、プレフィックスの埋め込みポリシーまたは名前付きポリシーで使用する場合は、サポートの再構成を参照してください)。
- allow-temporary-addresses： 一時（IA_IA）アドレスを要求する DHCPv6 クライアントを有効または無効にします（プリセット値は有効）。
- default-prefix-length-length：プレフィックスの委任では、クライアントまたはルータが明示的に要求していない場合は、委任されたプレフィックスのデフォルトのプレフィックス長。これは常に、プレフィックス範囲のプレフィックス長（プリセット値は 64 バイト）以下にする必要があります。
- reconfigure：リンク上のプレフィックスのプレフィックス ポリシー（組み込みまたは名前付き）を確認する際に、ポリシー階層の処理時に特別な処理を有効にします（IPv6 リースの再設定 （258 ページ）を参照）。
- preferred-lifetime：リースの優先ライフタイムのデフォルトの最大値（プリセット値は 1 週間）。
- v6-reply-options：クライアントへの応答で返される DHCPv6 オプション（プリセット値なし）。(プレフィックスDHCPv6 ポリシー階層 （202 ページ）の埋め込みポリシーまたは名前付きポリシーで使用する場合のこの属性の特別な処理については、を参照してください。
- valid-lifetime：リースの有効ライフタイムのデフォルトの最大値（プリセット値は 2 週間）。

ヒント　再設定属性の詳細については、を参照IPv6 リースの再設定 （258 ページ）してください。

サポートの再構成 (DHCPv6)

DHCPv6 の場合、サーバーは RECONFIGURE メッセージをクライアントに送信して、サーバーに新規または更新された構成パラメーターがあることをクライアントに通知できます。許可され、適切な認証を通じて許可された場合、クライアントは、サーバーとの更新、再バインド、または情報要求応答トランザクションを直ちに開始し、クライアントが新しいデータを取得できるようにします。このサポートがない場合、クライアントは、構成の更新を取得するためにリースを更新するまで待機する必要があります。

サーバーで再設定パケットをユニキャストするか、リレーエージェントを介してパケットを配信できます。どちらの方法を指定しない場合は、クライアントのクライアント クラス ポリシー、要求されたリースのプレフィックスポリシーまたはリンクポリシー、または system_default_policy (クライアント ポリシーではなく) によって優先される方法が決まります。ユニキャスト方式が使用できない場合 (クライアントに有効なアドレスリースがない場合) は、サーバーはリレーエージェントを使用します。リレーエージェントがない場合、サーバーはユニキャストを試みます。両方のエラーが発生すると、エラーになります。ユニキャスト方式では、指定されたリースが使用できない場合、サーバーは有効期間が最長のリースを選択します。

サーバーとクライアントは、再構成キーの追加セキュリティを使用して再設定サポートをネゴシエートします。内部プロセスは基本的に次のとおりです。

1. クライアントは、再設定受け入れオプション (20) を含む REQUEST、要請、または ADVERTISE パケットをサーバーに送信し、クライアントが再設定メッセージを受け入れることを示します。(逆に、DHCP サーバーは、クライアントが再構成メッセージを受け入れるかどうかについて、クライアントに再構成受け入れオプションを送信できます。このオプションは、再構成のサポートに必要です。

2. クライアントの Cisco Prime Network レジストラーポリシーで再設定属性allowがrequireordisallowに設定されている場合、DHCP サーバーはパケットを受け入れ、クライアント用の再設定キーを生成します。(サーバーは、キー値とその生成時間を、クライアント再構成キーおよびクライアント再構成キー生成時間属性に記録します。

3. サーバーは、再設定受け入れオプションとともに、認証オプション (11) の再設定キーを使用してクライアントに応答パケットを送信します。

4. クライアントは、再構成キーを記録して、サーバーからのメッセージの再構成を認証します。

5. サーバーは、クライアントを再構成する際に、再設定メッセージオプション(19)と、パケットと再構成キーから生成されたハッシュを含むauthオプションを使用して再設定パケットを送信します。再設定メッセージオプションは、クライアントが更新または情報要求パケットで応答する必要があるかどうかをmsg-typeフィールドで示します。

6. パケットを受信すると、クライアントはauthオプションに有効なハッシュが含まれていることを検証し、更新、再バインド、または情報要求パケットを返します。このパケットには、特定のオプションの更新を示すオプション要求(oro)オプション(6)が含まれています。(サーバーが、事前に構成されたタイムアウト値2秒でクライアントから応答を受信しない場合、サーバーは再設定メッセージを8回まで再送信し、クライアントの再構成プロセスを中止します。
7. サーバーは、構成パラメータのオプションを含む応答パケットをクライアントに送信します。パケットには、クライアントが要求しなかった場合でも、他の構成パラメータのアドレスと新しい値を含むオプションが含まれる場合もあります。クライアントは、これらの変更を記録します。

ポリシーのタイプ

ポリシーには、システムのデフォルト、名前付き、および埋め込みの3種類があります。

- **システムデフォルト** (system_default_policy)- すべてのスコープまたはプレフィックスに対して特定のオプションのデフォルト値を設定する場所を1つ指定します。システムのデフォルト・ポリシーを使用して、DHCPサーバーがサポートするすべてのネットワーク上のすべてのクライアントに共通の値を持つ属性および標準DHCPオプションを定義します。システムのデフォルト・オプションとその値を変更できます。システムのデフォルトポリシーを削除すると、元のDHCPオプションの一覧とシステム定義値を使用して再表示されます (下の表を参照)。

表 *22:* システムデフォルトポリシーオプションの値

システムのデフォルト オプション	定義済みの値
all-subnets-local	いいえ（False）
arp-cache-timeout	60 秒
broadcast-address	255.255.255.255
default-ip-ttl	64
default-tcp-ttl	64
dhcp-lease-time	604800 秒（7d）
ieee802.3-encapsulation	いいえ（False）
interface-mtu	576 バイト
mask-supplier	いいえ（False）
マックス・ドグラム再構成	576 バイト
非ローカル ソース ルーティング	いいえ（False）

システムのデフォルト オプション	定義済みの値
パス-mtu エージング タイムアウト	6000 秒
パス-mtu-プラトーテーブル	68, 296, 508, 1006, 1492, 2002, 4352, 8166, 17914, 32000
マスク検出の実行	いいえ（False）
ルーター発見	[はい（True）]
ルーター勧誘アドレス	224.0.0.2
tcp-キープアライブゴミ	いいえ（False）
tcp キープアライブ間隔	0 秒
トレーラーカプセル化	いいえ（False）

- Named-名前で明示的に定義するポリシー。通常、名前付きポリシーには、関連するスコープ、プレフィックス、またはクライアント グループに基いて名前が付けられます。たとえば、ポリシーには、ルーターなど、サブネットに固有の属性とオプションが割り当てられ、適切なスコープまたはプレフィックスに割り当てられる場合があります。

  Cisco プライムネットワーク レジストラーには、DHCP サーバーをインストールするときに名前が付けられたdefaultポリシーが含まれています。サーバーは、新しく作成されたスコープとプレフィックスにこのポリシーを割り当てます。この既定のポリシーは削除できません。

- Embedded:名前付きスコープ、スコープテンプレート、プレフィックス、プレフィックス テンプレート、クライアント、またはクライアント クラスに埋め込まれたポリシー(および制限付き)ポリシー。埋め込みポリシーは、対応するオブジェクトを追加 (または削除) するときに暗黙的に作成 (または削除) されます。埋め込みポリシー オプションには既定値がなく、最初は未定義です。

ヒント 埋め込みポリシーを作成または変更するオブジェクト (スコープ、プレフィックス、クライアント、またはクライアント クラス) を保存してください。この操作を行わないことは、Web UI を使用する場合によく発生するエラーです。埋Modifyめ込みポリシーと親オブジェクトの両方をクリックします。

ポリシー階層

DHCPv4 ポリシー階層

さまざまなレベルで設定されている競合する属性とオプションの値を排除するために、Cisco Prime Network レジストラー DHCP サーバーはローカルの優先度の方法を使用します。この関数は、よりグローバルなレベルで定義された属性値を無視しながら、ローカルに定義された属性とオプションの値を最初に採用し、それ以外の場合は定義されていないデフォルトの値を含みます。DHCP サーバーが DHCPv4 クライアントの処理決定を行う場合、次の順序で属性とオプションに優先順位を付けます。

1. クライアントの埋め込みポリシー。
2. クライアント名のポリシー。
3. クライアントクラスの組み込みポリシー。
4. クライアント クラスの名前付きポリシー。
5. クライアントのスコープ組み込みポリシー、またはサブネットの埋め込みポリシーをアドレス ブロックします。
6. クライアントの名前付きポリシー (または、名前付きポリシーがスコープに適用されていない場合は既定のポリシー) またはサブネットの名前付きポリシーをアドレス ブロックします。
7. system_default_policy内の残りの未対応の属性とオプション。属性には、ほとんどの場合にローカル ポリシーのデフォルト値が適用されます。

（注） DHCPv6 ポリシーの優先順位付けについてはDHCPv6 ポリシー階層 （202 ページ）、を参照してください。

DHCPv6 ポリシー階層

DHCPv6 は、追加の DHCPv6 固有の属性 (DHCPv4 の属性にほぼ類似) を含む既存のポリシーオブジェクトを使用します。DHCPv6 の場合、階層は次のようになります。

1. クライアントの埋め込みポリシー
2. クライアント名のポリシー
3. クライアントクラスの組み込みポリシー
4. クライアントクラスの名前付きポリシー
5. プレフィックス埋め込みポリシー
6. プレフィックス名付きポリシー
7. リンク埋め込みポリシー
8. リンクの名前付きポリシー
9. system_default_policy

属性の場合、最も多くのローカルポリシーの既定値が適用されます。この階層は、追加のリンク ポリシーとプレフィックスポリシーがスコープポリシーを置き換える点を除いて、DHCPv4 の場合と同じです。(DHCPv4 ポリシー階層との比較については、「」DHCPv4 ポリシー階層（202 ページ）を参照してください。

階層は、サーバーが単一のプレフィックスのコンテキストで処理するほとんどのポリシー属性に適用されます。ただし、サーバーは複数のプレフィックスのコンテキストで、いくつかの属性 (特に、高速コミット、再設定、 v6-reply-options、 v6-options 、 v6-vendor-options) を処理します。 このような場合、プレフィックス レベル (手順 5 と 6) での処理は少し異なります。

- サーバーがクライアントの再構成を必要とするか、許可するか、または許可しないかを制御する再設定属性の場合、サーバーはクライアントが使用できるリンク上のすべてのプレフィックスの埋め込みポリシーと名前付きポリシーをチェックします(選択タグに基づいて)。プレフィックスポリシーのいずれかに再設定属性disallowが設定されている場合、またはrequireに設定されている場合、サーバーはその設定を使用します。それ以外の場合、少なくとも 1 つのallowポリシーが に設定されている場合は、再構成が許可されます。それ以外の場合、サーバーは階層内の残りのポリシーをチェックします。（詳細については、IPv6 リースの再設定 （258 ページ）を参照してください）。
- クライアントが Rapid Commit を要求DHCPv6 サーバー属性の編集 （28 ページ）した場合 (を参照)、サーバーは、クライアントが使用できるリンク上のすべてのプレフィックスの埋め込みポリシーと名前付きポリシーをチェックします(選択タグに基づいて)。これらのポリシーの1つが、高速コミットの許可を無効にしている場合、サーバーは、Rapid Commit が要求の一部ではないかのようにクライアント要求を処理します。少なくとも1つのポリシーで高速コミットが有効になっている場合、クライアントは高速コミットを使用できます。この属性が設定されているプレフィックスポリシーがない場合、処理はステップ7で続行されます。
- オプション関連の属性については(をDHCPv6 オプションの設定 （220 ページ）参照してください)、サーバーはステップ 5 および 6 で特別な処理も行います。サーバーは、リンク上の各プレフィックスの埋め込みポリシーと名前付きポリシーをチェックします。次に、v6-reply-option属性が設定された最初の属性、またはv6-optionsまたはv6 ベンダーオプションの設定値を持つ最初の属性を使用します。
- サーバーは、プレフィックスを大文字小文字を区別しないアルファベット順にチェックします。
- サーバーは、ロケーションに依存しないリンクやユニバーサルリンク、およびその下のプレフィックスに関連するポリシーを無視します。トポロジ リンク (およびそれらのリンクの下のプレフィックス) のみが考慮されます。

ヒント　リンク上に複数のプレフィックスが設定されている構成では、プレフィックスポリシーのRapid Commit プロパティとオプション プロパティを設定することは避け、代わりにリンク ポリシーまたはその他のポリシーに設定します。

DHCP ポリシーの設定と適用

ここでは、DHCP サーバー レベルでポリシーを作成し、それを参照する特定のスコープまたはプレフィックスを許可する方法について説明します。ポリシーは、次の要素で構成されます。

- Name- 大文字と小文字を区別せず、一意である必要があります。
- 永続リース attribute— 永続リースは無期限です。
- Lease : DHCP サーバーでリースを更新する前に、クライアントが割り当てられたリースを使用できる期間 (組み込みポリシーではリース時間属性は使用timeできません。システムのデフォルトポリシーとデフォルトポリシーのデフォルトのリース時間は、7 日間(604800 秒)です。ポリシーには、クライアントリース時間とサーバーリース時間の 2 つのリース時間が含まれます。
  - - クライアントがリースが有効であると判断する期間を決定します。 Client lease time (ポリシー属性ではなく、DHCP オプションを使用してクライアントのリース時間を設定します。
  - - サーバーがリースを有効と見なす期間を決定します。 Server lease time サーバーのリース期間は、リース猶予期間とは関係なく行われます。リース期間と猶予期間が終了するまで、サーバーはリースを別のクライアントに割り当てません。

> ⚠ **注意** Cisco Prime Network レジストラーでは、特殊な状況で 2 回のリース時間の使用がサポートされていますが、一般に、サーバー リース時間属性を使用しないことを推奨します。

クライアント DNS 名に関する情報を保持し、リースを頻繁に更新する場合は、これら 2 つの異なるリース時間を設定できます。1 つのリース期間を使用して、有効期限が切れると、サーバーはそのクライアント DNS 名を保持しなくなります。ただし、クライアントリース時間が短く、サーバーのリース期間が長い場合、クライアントのリース期限が切れた後でも、サーバーはクライアント情報を保持します。リースの詳細については、を参照リースの管理 （227 ページ）してください。

- — リースの期限が切れてから再割り当てができない期間 (組み込みポリシーでは使用できません)。 Lease grace period
- — DNS 更新の構成では、実行する DNS 更新のタイプ、関連するゾーン、更新する DNS サーバー、および関連するセキュリティを指定します。 DNS update configuration ポリシーは、DNS の前方更新構成オブジェクトと逆引き DNS 更新構成オブジェクトを決定し、DNS サーバーが複数のゾーンをホストする場合に使用する転送ゾーンを指定することもできます。(DNS 更新の構成の詳細についてはDNS 更新設定の作成 （317 ページ）、を参照してください)。

- - オプション値を追加するには、「」を参照してくださいDHCP options。 ポリシーの DHCP オプションと属性の設定 （207 ページ）

ローカル基本または詳細とリージョン Web UI

ステップ 1 [**デザイン**] メニューのPolicies [DHCP **設定**]サブメニューの下で [DHCP ポリシーの一覧/ 追加] ページを開きます。

ステップ 2 デフォルトのポリシーとsystem_default_policyは既に提供されています。

ステップ 3 [ポリシー] ウィンドウの [**ポリシーの追加**] アイコンをクリックし、ポリシーに一意の名前を付けます (必須)。

ステップ 4 オファーのタイムアウト値と猶予期間の値を設定するか、空のままにします。

ステップ 5 必要に応じてDHCP リース時間をAdd DHCP Policy入力し、名前付きポリシーを追加する場合にクリックします。

ステップ 6 [DHCP ポリシーの編集] ページでは、次の操作を実行できます。

- 必要な DHCP オプションをポリシーの DHCP オプションと属性の設定 （207 ページ）追加します(次の例を参照してください。
  - **リース時間**—dhcp リース時間(51)オプションを設定します。
  - **制限数**—式の使用方法 （389 ページ）を参照してください。
  - **予約にクライアント ID を使用**するクライアント ID の上書き （471 ページ）(を参照してください)。

  ベンダー固有のオプションを設定するには、「」を参照してください標準オプション定義セットの使用 （211 ページ）。
- 詳細モードで、次のようなポリシー属性を設定します。
  - **使用不可タイムアウト**-使用不可リースのタイムアウトの設定 （264ページ）を参照してください。
  - **すべての更新を禁止**するリース更新の抑制 （259 ページ）— を参照してください。
  - **再起動時にすべての更新を禁止する**
  - **永久リース**(推奨されません)
  - **リース保持制限**
- DNS 更新に含める転送ゾーンまたは逆ゾーンを決定する DNS 更新構成を設定します (DNS **更新設定**)。次の設定を行うことができます。
  - **転送 *dns* 更新**: 転送ゾーンの更新構成の名前。転送ゾーンと逆ゾーンに対して異なる更新設定を設定できます。
  - ***forward-zone-name***:必要に応じて、更新構成の転送ゾーンを上書きします。DNS サーバーが複数のゾーンをホストしている場合に使用します。

- **逆引き *dns* 更新**- 逆ゾーンの更新構成の名前。クライアント要求に適用可能なポリシー階層内のポリシーに設定されていない場合 (DHCPv4 ポリシー階層 （202 ページ）を参照) 、DHCP サーバーは forward-dnsupdate 構成を使用します。

ステップ 7 Save をクリックします。

ステップ 8 DHCP サーバーをリロードします。

地域 Web UI では、レプリカ ポリシーをプルし、ローカル クラスターにポリシーをプッシュすることもできます。(地域DHCP ポリシーの設定 （197 ページ）政策管理については、を参照してください。

CLI コマンド

ポリシーpolicyを作成するには、名前createを使用します。次にpolicy、set名前offer-timeout=値と名前policy値を使用して、これら 2 つの値を設定します。 set grace-period=

ポリシー オプションを設定するにはpolicy、名前setOption<opt-nameを使用する |id>値[-blob] [-**ラウンドロビン**]

- setLeaseTime policy — 名前の時間を使用する- 時間を指定します。 Lease time
- policy -名前dhcpの値enableとget-subnet-mask-from-policyを組み合わせて使用します。 setOption subnet-mask Subnet mask

-roundrobinが有効な場合、DHCP サーバーは、異なる回転順序で複数の値を含むオプションデータを返すように指示します。特定のクライアントは常に同じ順序を取得しますが、異なるクライアントは、クライアント識別子に基づいてオプションに対して構成された複数の値の順序の異なる「ローテーション」を取得します。

オプションの設定を確認するには、policy 名前 listOptionsまたはpolicy 名前 getOption<opt-name を使用します。id>.

永続的なリースを有効にするには(推奨されませんpolicy)、名前enablepermanent-leasesを使用します。永久リースを有効にすると、dhcp-リース時間オプション(51)が無限に設定されます。

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。

- **ポリシー** <名前|**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**]
- **ポリシー**<名前|**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**]
- **ポリシー**名**クラスタリストを**再利用する [-**レポートのみ**|-**レポート**]

関連項目

ポリシーのタイプ （200 ページ）

ポリシーの複製

CLI では、policy clone-name create clone=policy を使用して既存のポリシーからポリシーを複製してから、そのクローンを調整できます。次に例を示します。

```
nrcmd> policy cloned-policy create clone=example-policy-1 offer-timeout=4m
```

ポリシーの DHCP オプションと属性の設定

DHCP オプションは、DHCP クライアントにドメイン、ネームサーバー、サブネット ルーター アドレスなどの構成パラメータをDHCP オプション定義セットとオプション定義の作成 （210 ページ）自動的に提供します (を参照)。Cisco Prime Network レジストラーユーザー インターフェイスでは、クライアントに返されるパケットには実際には影響しないオプション値をポリシーに設定できます(ホスト名やdhcp-server-identifierなど)。

サーバーは、次の BOOTP 属性値と DHCP 属性値を順番に検索し、応答パケット内で最初に出現した値を返します。

- パケット・シアドは、siaddrパケット・フィールドに戻されます。
- ファイルフィールドに返されるパケット ファイル名
- snameフィールドに返されるパケットサーバー名

関連項目

オプション値の追加

DHCPオプション値を表示、設定、設定解除、および編集できます。オプション値を設定すると、DHCPサーバーは、指定されたオプション名に必要に応じて、既存の値を置き換えるか、新しい値を作成します。Cisco Prime Network レジストラー DHCP オプションはカテゴリにグループ化され、さまざまな使用状況で設定する必要があるオプションを識別するのに役立ちます。カスタム オプション定義を作成して、カスタム オプション値の入力をカスタム オプション定義の作成 （213 ページ）簡略化することができます (を参照)。

ローカル基本または詳細とリージョン Web UI

ステップ 1 ポリシーを作成する (をDHCP ポリシーの設定と適用 （204 ページ）参照)。

ステップ 2 [DHCP ポリシーの編集] ページで、ドロップダウン リストで DHCP オプションの番号と名前を選択して、各 DHCP オプションをポリシーに追加します。選択肢は、オプション値のデータ型を示しますオプション定義データ型と繰り返し回数 （221 ページ）(を参照)。

ヒント オプションは、名前、番号、または (DHCPv4 の場合) レガシー (グループ化) でソートできます。

ステップ 3 [値] フィールドに適切なオプション値を追加します。Web UI は、入力された値に基づいてエラーチェックを行います。たとえば、ポリシーのリース期間を追加するには、[数値] ドロップダウン リストの[51] dhcp-リース時間 (符号なし時間)オプションをクリックし、[値] フィールドにリース時間の値を追加します。(オプションにはプリセット値がありません。

ヒント 別のユーザーがオプション定義を編集している間に、ポリシーのオプションを構成する場合は、セッションからログアウトし、ログインし直して新しいオプション定義を取得します。

ステップ 4 各Add Optionオプションをクリックします。値を指定する必要があります。

ステップ 5 Save をクリックします。

ヒント 新しいオプション値を追加する場合や既存の値を編集する場合は、 をSaveクリックしてポリシーオブジェクトを保存してください。

CLI コマンド

オプション値を表示するには、policy名前getOption<opt-nameを使用します。id>policyと名前 listOptions. オプション値を設定するには、policy 名前 setOption<opt-nameを使用する |id>値[-blob] [-**ラウンドロビン**] オプション値を設定すると、DHCP サーバーは、指定されたオプション名に対して、既存の値を置き換えるか、必要に応じて新しい値を作成します。-roundrobinが有効な場合、DHCP サーバーは、異なる回転順序で複数の値を含むオプションデータを返すように指示します。特定のクライアントは常に同じ順序を取得しますが、異なるクライアントは、クライアント識別子に基づいてオプションに対して構成された複数の値の順序の異なる「ローテーション」を取得します。オプション値の設定を解除するにはpolicy、名前unsetOption<opt-name を使用する |id>.

サブオプションの複雑な値の追加

サブオプションなど、より複雑なオプション値を追加する場合は、括弧付きのストリング形式を使用します。この形式では、次のことが必要です。

- 各オプション・レベル (オプション、サブオプション、サブサブオプション) を括弧で囲みます。
- 複数の値を指定する場合は、カンマで区切ります。

- パックされたデータのデータ フィールド(サブオプション コードまたは長さが欠落している) をセミコロンで区切ります。

たとえば、cablelabs クライアント設定オプション(122)には、通常、サブオプションとサブオプションが 10 個あります。この例では、サブオプション 1、2、3、および 4 のデータ値を設定する構文を示し、サブオプション 3 の 2 つのサブサブオプションとサブオプション 4 の 3 つのサブサブオプション (データがパックされ、コード番号がない) が含まれています。

```
(primary-dhcp-server 1 10.1.1.10)
(secondary-dhcp-server 2 10.2.2.10)
(provisioning-server 3 (flag 0; provisioning-server server.example.com.))
(as-backoff-retry 4 (as-backoff-retry-initial-time-ms 10;
as-backoff-retry-max-time 10s; as-backoff-retry-count 100))
```

サブオプション名 (プライマリ dhcp-server など) はオプションです。そのため、多くの場合、コード番号とデータ値(またはパックされたデータのデータ値)のみを使用して、誤植エラーや解析エラーを最小限に抑える方が安全です。サブオプション名を取り除く前の例の最適化された (そして優先される) バージョンは次のとおりです。

```
(1 10.1.1.10) (2 10.2.2.10) (3 (0;server.example.com.)) (4 (10;10s;100))
```

数値コード値を使用する場合でも、サブオプションを表示する際に、Cisco Prime Network レジストラーには必ず同等の名前が含まれます(を参照)。DHCP オプション定義セットとオプション定義の作成 （210 ページ）

エンタープライズ ID を含むサブオプション (オプション 125 など) を含めるには、ポリシー・オプション値を入力する場合などに、次の形式を使用します。

```
(enterprise-id 1((1 10.1.1.1) (2 10.2.2.2) (3 www.cisco.com)))
```

かっこは、エンタープライズID 自体、サブオプションをグループとして囲み、各サブオプションを囲みます。

MAP-T および 4rd オプション

オプション値フィールドでカプセル化された DHCPv6 オプション (つまり、最上位のオプション) を指定できるようになりました。そのため、ソフトワイヤ MAP や 4 番目のオプションなどのオプションを指定できます。次に例を示します。

```
nrcmd> policy softwire setv6option s46-cont-mapt "(s46-rule (flags 0; ea-len 12;
prefix4-len 24; ipv4-prefix 10.1.2.0; prefix6 1234::/64 (s46-portparams (offset 10;
psid-len 10; psid 43))))(s46-dmr 2345::/64)"
```

s46-portparamsをs46 ルールと同じレベルにする場合は、次の手順を実行します。

```
nrcmd> policy softwire setv6option s46-cont-mapt "(s46-rule (flags 0; ea-len 12;
prefix4-len 24; ipv4-prefix 10.1.2.0; prefix6 1234::/64)) (s46-portparams (offset 10;
psid-len 10; psid 43))(s46-dmr 2345::/64)"
```

サブオプションとカプセル化されたオプションの構文が変更されました。以前は、idを要求し、名前が存在する場合は名前を無視していました。今、私たちはもはやidを必要としません.nameが存在する場合は、有効でなければなりません (無視されません)。nameとidが存在する場合、名前の id はidと一致する必要があります。データがオプション ID である場合は、"nameid data"として指定する必要があります。

カプセル化オプションの場合、許可されたオプションのみが指定されていることを確認するチェックは行いません。任意のオプションを指定できます。

組み込みポリシーの作成と編集

埋め込みポリシーは、DHCPv4 スコープまたはスコープ テンプレート、DHCPv6 プレフィックスまたはプレフィックス テンプレート、クライアント、またはクライアント クラスに埋め込まれています。埋め込みポリシーを作成または編集できます。

ローカル アドバンスド Web とリージョン UI

ステップ 1 Designメニューから、ローカル Web UI で DHCPv4 または DHCPv6 に表示される、 Scopes Scope Templates 、 Clients Client-Classes、 Prefixes、 Links、 または のいずれかを選択します。(地域の Web UI には、 Scope Templates Client-Classes、 Prefixes、Linksおよび の選択を含めることができます)。

ステップ 2 左側のペインでオブジェクトの名前をクリックして、そのオブジェクトの編集ページを開きます。

ステップ 3 ページCreate New Embedded PolicyのEdit Existing Embedded Policy埋め込みポリシー セクションの下をクリックします。これにより、オブジェクトの [DHCP 埋め込みポリシーの編集] ページが開きます。

ステップ 4 必要に応じて値を変更し、Modify Embedded Policyをクリックします。

ステップ 5 オブジェクトの [編集] ページで、[保存] をクリックして変更を**保存**します。

CLI コマンド

コマンドがオブジェクト名の後にclient-class-policy-policy を続けて開始する場合は、組み込みコマンド (クライアント クラス名set属性=値など) を使用します。

DHCP オプション定義セットとオプション定義の作成

Cisco Prime Network レジストラーでは、リース時間やルータ アドレスなどのポリシーにオプション値を設定します。RFC 2132 以降では、多くの RFC が DHCP オプション値のフォーマットを記述しています。Web UI および CLI では、オプション定義を使用して、ポリシー内のオプション値のフォーマット設定を制御します。

DHCPv6 オプションで DHCPv4 オプションは使用しないでください。これらは一意で、独立しています。現在、約 46 の DHCPv6 オプションがあります。これらのオプションのほとんどは DHCPv6 プロトコル インフラストラクチャ オプションであり、ユーザー定義はできません。16 ビットのオプション コードと 16 ビットの長さを使用します (DHCPv4 では、両方に 8 ビットしか使用しません)。ポリシーでのオプションの設定および設定されたオプションの動作は、DHCPv4 の場合と似ています。ポリシーDHCPv6 オプションの設定 （220 ページ）階層に関連するクライアント処理の詳細については、「」を参照してください。

DHCPv4アドレス・スペースとDHCPv6アドレス・スペースに対して、以下のようにオプション定義を個別に定義できます。

- **標準(組み込み)オプション**-RFCによって定義されます。Web UIでは、これらはdhcp-configおよびdhcp6-config定義セットに含まれています。CLIには、非表示になっているが、特に呼び出すとアクセス可能なdhcp-defaultおよびdhcp6 **デフォルト**の定義セットが追加されています。（標準オプション定義セットの使用 （211 ページ） を参照）。
- **カスタム オプション**-指定されたdhcp-config定義セットまたはdhcp6-config定義セット内の定義を新規作成または変更しました。Web UI で定義を追加または変更すると、CLI の dhcp-custom定義セットまたはdhcp6 **カスタム**定義セットに追加されます。（カスタム オプション定義の作成 （213 ページ） を参照）。
- **ベンダー固有のオプション**- 独自の定義セットで定義されます。ケーブルラボ定義セット(dhcp-**ケーブルラボ設定**およびdhcp6 **ケーブルラボ設定**)は、Cisco Prime Network レジストラーで事前設定されています。CLIには、dhcp-**ケーブルラボデフォルト**、dhcp6-**ケーブルラボデフォルト**、dhcp-**ケーブルラボ-カスタム**、およびdhcp6 **ケーブルラボカスタム**定義セットも含まれています。（標準オプション定義セットの使用 （211 ページ） を参照）。

関連項目

標準オプション定義セットの使用 （211 ページ）

カスタム オプション定義の作成 （213 ページ）

ベンダー固有オプション定義の作成 （213 ページ）

オプション定義データ型と繰り返し回数 （221 ページ）

サブオプション定義の追加 （222 ページ）

オプション定義セットのインポートとエクスポート （223 ページ）

オプション定義セットのローカル クラスタへのプッシュ （224 ページ）

レプリカ データからのオプション定義セットのプル （224 ページ）

ポリシーのオプション値の設定 （219 ページ）

標準オプション定義セットの使用

Cisco Prime Network レジストラーでは、DHCPv4 dhcp-configdhcp6-configおよび DHCPv6 オプション定義にそれぞれ2つの標準の組み込みオプション定義セットと が用意されています。これらのセットに新しいオプション定義を作成することも、既存のオプション定義を上書きすることもできます。新しいオプション定義または上書きされたオプション定義は、アスタリスク(*) で識別されます。これらの定義を削除することができ、削除確認は行いません。ただし、上書きされた定義を削除した後にセットを保存すると、元の定義がセットに再表示されます。

注意 標準定義を任意に変更する(またはサブオプション定義を追加する)と、構成に悪影響を及ぼす可能性があります。

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 メニューからDesignOptionsDHCPv4またはDHCPv6サブメニューの下で選択し、「DHCP オプション定義セットのリスト/追加」ページを開きます。(DHCP オプション定義は基本モードでは使用できません。

ステップ 2 (DHCPv4)dhcp-configまたはdhcp6-config(DHCPv6) リンクをクリックして[DHCP オプション定義セットの編集]ページを開き、[**オプション定義**]タブをクリックします。[DHCP オプション定義の一覧/追加] ページで定義済みの定義を表示します。これらは、ポリシーに追加するオプション値のフォーマットを制御する定義です。サブオプション定義がある場合は、それらを展開して表示することができます。

ステップ 3 定義を追加するには、[DHCPAddOptionDefinitionオプション定義の編集] ページのアイコンをクリックします。オプションに、数、名前、説明、タイプ、および繰り返し回数を指定します(オプションの複数のインスタンスが許可されているか、必須であるかに関係なく)。(データ型と繰り返しカウント値の詳細についてはオプション定義データ型と繰り返し回数 （221 ページ）、を参照してください。

（注） 既に存在するオプション番号または名前に対してオプション定義を追加することはできません。ただし、ページ上にハイパーリンクとして表示されるオプション定義は変更できます。

ステップ 4 Add Option Definition をクリックします。次に、[DHCP オプション定義セットの一覧/追加]Saveページで、をクリックします。

ステップ 5 標準セットの元の定義に戻すには、[**キャンセル**]ボタンをクリックします。

ステップ 6 地域 Web UI では、レプリカ定義セットとローカル クラスターへのプッシュ定義セットをプルすることもできます。（レプリカ データからのオプション定義セットのプル （224 ページ） およびオプション定義セットのローカル クラスタへのプッシュ （224 ページ） を参照）。

CLI コマンド

標準の DHCP オプション定義の一覧をすべて表示option-setdhcp-configするにはshow、[] option-set dhcp6-config showまたは []、またはoption{id |名前}オプションセットshowを使用して特定の定義を表示します。次に例を示します。

```
nrcmd> option-set dhcp-config
nrcmd> option subnet-mask dhcp-config show
```

セットに定義を追加するにはoption 、idオプションセット createオプション名型[属性=値] を使用します。既に存在するオプション ID(番号) または名前の定義を追加することはできません。たとえば、dhcp-config オプション セットに名前 example-option を指定して、文字列タイプを指定してオプション番号 222 を追加するには、次のように使用します。

```
nrcmd> option 222 dhcp-config create example-option AT_STRING
```

特定のオプション属性値を取得するには、 option {id|名前}オプションセット get属性。オプション属性を変更するにはoption、{id |名前}オプションセット set属性=値。オプション属性の設定を解除することもできます。

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| option-setすべて>プル<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| option-set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リスト [-レポートのみ | option-set -レポート]

カスタム オプション定義の作成

標準セット内にカスタム オプション定義を作成できます。[DHCP dhcp-configdhcp6-configオプション定義セットの一覧/追加]ページでをクリックするか、または[DHCP オプション定義セットの追加]ページで設定します。次に、**のステップ**3に標準オプション定義セットの使用 （211 ページ）進みます。

ベンダー固有オプション定義の作成

ベンダ固有のオプション データを要求する DHCP クライアントに送信できます。

（注） ベンダー固有のオプションには、いくつかのオプション コードが用意されているので、ベンダ固有のオプション定義を作成するオプション コード番号を明示的に指定する必要があります。

Cisco Prime Network レジストラーでは、web UI または CLI で id オプションoptionセット名createを使用してベンダー固有のオプション定義を作成できます。(オプションデータ型の詳細については、を参照オプション定義データ型と繰り返し回数 （221 ページ）してください。

ベンダー固有のオプションは、次の DHCP オプションで送信されます。

- **ベンダー・カプセル化オプション *(43)***-これをバイナリー・データ・タイプに設定し、ベンダー固有のサブオプション定義を追加します。(親オプション定義のデータ型はプレースホルダのみです。サブオプション定義は、有効なオプション値のフォーマットを定義します。
- ***v-i-vendor-info (125)* または *DHCPv6 の vendor-options (17)***-これを vendor-opts データタイプに設定し、ベンダー固有のサブオプション定義を追加します。

DHCPv4 オプション 43 および 125、および DHCPv6 オプション 17 について、ベンダー固有のオプション定義を作成できます。作成するベンダーオプション定義セットに、ベンダー固有のオプション定義を追加します。

注意 オプション定義のプロパティを変更したり、オプション定義を完全に削除したりすると、ポリシーに予期しない副作用が生じる可能性があります。カスタムオプション定義を削除する場合は、オプション値を含むポリシーも確認してください。オプション定義を変更すると、保存されるものではなく、表示方法が変更されるため、ポリシーが異なる形式のオプション値を返す必要がない場合は、ポリシー値を変更する必要はありません。いくつかのオプションの種類は非常に似ていますが、それらのオプションの種類を変更すると、副作用が起きることができます。たとえば、文字列と DNS 名はどちらもユーザー インターフェイスに文字列値として入力されますが、書式設定されたオプションの値は大きく異なります。

（注） Cisco Prime Network レジストラーは、ベンダー固有のオプション定義セットdhcp-cablelabs-configとdhcp6-cablelabs-configベンダー固有のオプション定義セットに、別個の CableLabs(エンタープライズ ID 4491)オプション定義を事前に設定します。

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 [**デザイン**] メニューの[DHCPv4]または[DHCPv6]サブメニューの下にある[**オプション**]を選択して、[DHCP オプション定義セットの一覧/追加] ページを開きます。既存の DHCPv4 または DHCPv6 オプションを表示します。

ステップ 2 [オプション] ウィンドウの [**オプションの追加**] アイコンをクリックして、[オプション定義セットの追加] ダイアログ ボックスを開きます。

ステップ 3 オプション定義セットの名前を入力し、[DHCP **タイプ**]ドロップダウンリストから[DHCPv4]または[DHCPv6]を選択します。

ベンダー固有のオプション定義を作成する場合は、以下を使用します。

- オプション 43 で、ベンダー オプション文字列フィールドに値を入力します。(オプション 43 のベンダー オプション セットとベンダー オプション値の作成のサンプル手順については、以降のセクションを参照してください。
- DHCPv4 のオプション 125、または DHCPv6 のオプション 17 に、有効なエンタープライズ・オプション・エンタープライズ ID 値を入力します。

ステップ 4 [**オプション定義セットの追加**] をクリックします。

ステップ 5 左側のペインで追加されたオプション定義セット名をクリックします。

ステップ 6 [DHCP オプション定義セットの編集]ページで、[**オプション定義**] タブをクリックします。既存のオプション定義は、このページに表示されます(新規または変更された標準定義はアスタリスクでマークされます)。

ステップ 7 [オプション定義の追加（Add Option Definition）] アイコンをクリックします。オプション定義の ID 番号と、その名前と説明を入力します。クライアントがベンダー固有のオプション定義を認識するには、IDは 43、125、または 17 (DHCPv6 の場合) である必要があります。オプション名は、RFC で指定された名前と一致する必要はありません。

ステップ 8 データタイプと繰り返し回数を選択します(または次のフィールドに絶対繰り返し回数を入力します)。データ型は次の値にする必要があります。

- オプション 43 のバイナリ (AT_BLOB)。
- オプション 125 (DHCPv4 の場合) およびオプション 17 (DHCPv6 の場合) の vendor-opts (AT_VENDOR_OPTS)。

(データ型と繰り返しカウント値の詳細についてはオプション定義データ型と繰り返し回数（221ページ）、を参照してください。

ステップ 9 [オプション定義の追加（Add Option Definition）] をクリックします。次に、[DHCP オプション定義の一覧] ページで [**保存**] をクリックします。

ローカル拡張 Web UI を使用して、オプション 43 のベンダー オプション セットとベンダー オプション値を作成します。

ステップ 1 [設計（Design）] メニューで、[DHCPv4] または [DHCPv6] サブメニューから [オプション（Options）] を選択し、[DHCPオプション定義セットの一覧/表示（List/Add DHCP Option Definition Sets）] ページを開きます

ステップ 2 [オプション（Options）] ペインの [オプションの追加（Add Options）] アイコンをクリックし、[OptionDefinitionSetの追加（Add OptionDefinitionSet）] ダイアログボックスを開きます。

ステップ 3 次の属性の値を入力します。

- [名前（Name）]：オプション定義セットの名前（AP1130 など）。
- [DHCP タイプ（DHCP Type）]：このセット内のすべての子のタイプ識別子のバイトサイズ。ドロップダウン リストからDHCP v4を選択する必要があります。
- [ベンダーオプション文字列（Vendor Option String）]：DHCP クライアント デバイス ベンダーが提供するオプション 60 からのベンダークラス識別子の文字列を正確に指定します。たとえば、Cisco AP c1130 です。

ステップ 4 [OptionDefinitionSetの追加（Add OptionDefinitionSet）]をクリックします

[DHCP オプション定義セットの一覧/追加] ページが表示されます。

ステップ 5 [AP1130] をクリックすると、表示されるオプション定義セットの名前が表示されます。

[DHCP オプション定義セット AP1130 の編集] ページが表示されます。

ステップ 6 [**オプション定義**] タブをクリックし、[**オプション定義の追加**] アイコンをクリックします。

ステップ 7 次の属性の値を入力します。

- [番号（Number）]：オプションコードの番号。43 を入力する必要があります。
- [名前（Name）]：オプションコードの名前。43 と入力する必要があります。
- [タイプ（Type）]：この属性の名前。たとえば、ap1130-オプション-43。

ステップ 8 [オプション定義の追加（Add Option Definition）] をクリックします。

このボタンをクリックしても、オプション定義セットに加えた変更は保存されません。[DHCP オプション定義の一覧] ページに設定されているオプション定義のみが一覧表示されます。

ステップ 9 [オプション定義] タブで、新しいオプション定義の名前 (ap1130-option-43) をクリックし、[**サブ オプション定義の追加**] をクリックします。

ステップ 10 次の属性の値を入力します。

- [番号（Number）] : このサブオプションのオプションコード。この例では、241 と入力する必要があります。
- [名前（Name）] : この属性の名前。たとえば、「ap1130-サブオプション-241」です。
- [タイプ（Type）] : サブオプション値のデータ型。この例では、ドロップダウン リストから [IP アドレス] を選択する必要があります。
- [繰り返し（Repeat）] : このタイプの繰り返し回数。この例では、ドロップダウン リストから [1+] を選択する必要があります。

ステップ 11 [オプション定義の追加（Add Option Definition）] をクリックし、[保存（Save）] をクリックします。

ステップ 12 [**デザイン**] メニューの[DHCP **設定**]サブメニューの下の [**ポリシー**] を選択して、[DHCP ポリシーの一覧表示/追加] ページを開きます。

ステップ 13 このオプションを設定するポリシーを選択します。または、詳細モードで新しいポリシーを追加します。

選択内容に応じて、[DHCP ポリシーの編集] policy_nameまたは [DHCP ポリシーの追加] ページが表示されます。

ステップ 14 [DHCPv4 **ベンダ オプション**] ドロップダウン リストから、オプション定義セットの名前 (AP1130) を選択し、[**選択**] をクリックします。

ステップ 15 [**名前**]ドロップダウン リスト(「ap1130-option-43」)からオプション定義を選択し、[値]フィールドに値を入力します。次に例を示します。

```
(241 3.3.3.3,4.4.4.4)
```

ステップ 16 [オプションの追加（Add Option）] をクリックし、[保存（Save）] をクリックします。

ステップ 17 DHCP サーバーをリロードします。

例: Cisco AP デバイスのベンダー オプション セットの作成

このセクションで説明するサンプル手順を使用して、Cisco アクセス ポイント(AP)デバイス、SunRay デバイス、および Cisco 79xx IPPhone 用の CLI からベンダー のオプション セットとベンダー オプション値を作成できます。

ライトウェイト アクセス ポイント プロトコル(LWAPP)AP にオプション 43 を使用するには、DHCP サーバーとして Cisco Prime Network レジストラーを使用している場合、ベンダー オプション 43 が必要です。この例は、Cisco Aironet 1130 シリーズに固有のものです。この例を変更して、Cisco Aironet 1200 シリーズや Cisco Aironet 1240 シリーズなど、他のベンダー オプションのオプション 43 を設定できます。

ステップ 1 次の内容の .txt ファイルを作成します。

```
#
# Version: 1
# 6.2+ Option-set example for Option 43 with suboptions for Cisco APs
#
# NOTE: Need to edit vendor option string to Exact match AP Model string in Option-60.
#
# For compatibility with pre-6.2 vendor options ensure that
# name=vendor-option-string. (Not True in this test example.)
# ===================================================================
{
( id-range = 1 )
( vendor-option-string = Cisco AP c1130 )
( name = APtest )
( children = [
{
( id = 43 )
( name = pxe-sample )
( desc = )
( base-type = AT_BLOB )
( children = [
{
( id = 241 )
( name = controller )
( desc = ap controller )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
} ]
)
} ]
)
}
```

ステップ 2 次の場所にオプションセットCiscoAP.txtとしてファイルを保存します。

/opt/nwreg2/local/usrbin

ステップ 3 インポート オプション セット ファイル コマンドを使用して CLI からオプションセットCiscoAP.txt ファイルをインポートします。次に例を示します。

```
nrcmd> import option-set OptionSetCiscoAP.txt
```

(オプション定義セットのインポートについては、「」を参照オプション定義セットのインポートとエクスポート （223 ページ）してください)。

ステップ 4 policy name setVendorOption <opt-name | id> opt-set-name value [-blob] コマンドを使用してポリシーにベンダー固有のオプションデータを設定します。

たとえば、オプションセット APtest のベンダー オプション 43 データを値 (241 3.3.3,4.4.4.4) に設定するには、名前テストを持つ既存のポリシーで、次のコマンドを使用します。

```
nrcmd> policy test setVendorOption 43 APtest "(241 3.3.3.3,4.4.4.4)"
nrcmd> save
```

ステップ 5 DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

例: SunRay デバイスのベンダー オプション セットの作成

次のサンプル手順を使用して、SunRayデバイス用の複数のサブオプションを使用してベンダーオプションセットを作成します。

ステップ 1 次の内容の .txt ファイルを作成します。

```
#
# Option Definition Set Export/Import Utility
# Version: 1
# 6.2 Option-set example for Option 43 with suboptions for Sun SunRay.
#
# NOTE: Need to edit vendor option string to match Option-60
#
# For compatibility with pre-6.2 vendor options ensure that
# name=vendor-option-string.
# ========================================================================
{
( id-range = 1 )
( vendor-option-string = sunray )
( name = sunray )
( children = [
{
( id = 43 )
( name = option43 )
( desc = )
( base-type = AT_BLOB )
( children = [
{
( id = 21 )
( name = AuthSrvr )
( desc = AuthSrvr )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
}]
{
( id = 35 )
( name = AltAuth )
( desc = AltAuth )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
}
{
( id = 36 )
( name = BarrierLevel )
( desc = BarrierLevel )
( base-type = AT_SHORT )
}
]
)
} ]
)
}
```

ステップ 2 次の場所にオプションセットサンレイ.txtとしてファイルを保存します。

/opt/nwreg2/local/usrbin

ステップ 3 インポート オプション セット ファイル コマンドを使用して CLI から OptionSetSunRay.txt ファイルをインポートします。次に例を示します。

```
nrcmd> import option-set OptionSetSunRay.txt
```

(オプション定義セットのインポートについては、「」を参照オプション定義セットのインポートとエクスポート （223 ページ）してください)。

ステップ 4 policy name setVendorOption <opt-name | id> opt-set-name value [-blob] コマンドを使用してポリシーにベンダー固有のオプションデータを設定します。

たとえば、オプションセット APtest のベンダー オプション 43 データを値 (241 3.3.3,4.4.4.4) に設定するには、名前テストを持つ既存のポリシーで、次のコマンドを使用します。

```
nrcmd> policy test setVendorOption 43 APtest "(241 3.3.3.3,4.4.4.4)"
nrcmd> save
```

ステップ 5 DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

例: Cisco 79xx IP Phone のオプション セットの作成

Cisco 79xx IPPhone のオプション セットを作成するには、次のサンプル手順を使用します。

ステップ 1 オプションを定義します。

```
nrcmd> option 150 dhcp-custom create voip-tftp-server AT_IPADDR desc="VOIP Option-150 Server"
repeat=ONE_OR_MORE
```

ステップ 2 構成済みのオプションを表示します。

```
nrcmd> option dhcp-config list
```

ステップ 3 ポリシーのデフォルト setを使用してポリシーを設定するオプション voip-tftp-server ip アドレス. 次に例を示します。

```
nrcmd> policy default setOption voip-tftp-server 192.168.1.254
```

ステップ 4 ポリシー設定を確認します。

```
nrcmd> policy default getOption voip-tftp-server
```

ステップ 5 DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

ポリシーのオプション値の設定

ポリシーにオプション値を入力します。サーバー構成のオプション定義は、入力する形式と値を制御します。

ローカルアドバンスドおよびリージョン Web UI

[DHCP ポリシーの一覧表示/追加] ページで、ポリシーをクリックして編集します。(基本モードでは、ポリシーのオプションを設定できないことに注意してください。[DHCP ポリシーの編集] ページで、次の操作を行います。

- ポリシーの標準の DHCPv4 または DHCPv6 オプション値を入力するには、[DHCPv4 **オプション**] または[DHCPv6 **オプション**] ドロップダウン リストから選択し、オプションの値を設定します。Add Option をクリックします。
- ポリシーのベンダー固有の DHCPv4 または DHCPv6 オプション値を入力するには、DHCPv4 **ベンダー オプション**または DHCPv6 **ベンダー**オプション ドロップダウン リストで Select オプション定義セットを選択し、をクリックします。ページが変更され、オプションを含むドロップダウン リストが表示されます。を選択し、Add Option をクリックします。

このページでポリシー属性を編集することもできます。[ ]Modify Policy をクリックします。

構成済みのポリシー オプションを編集するには、[DHCP ポリシーの編集] ページで構成済みオプションの名前をクリックし、[DHCP ポリシー オプションの編集] ページを開きます。新しい値を入力し、Modify Option をクリックします。

CLI コマンド

次のいずれかのコマンドを使用します。

```
nrcmd> policy name setOption {opt-name | id} value [-blob] [-roundrobin]
nrcmd> policy name setV6Option {opt-name | id}[.instance] value [-blob] [-roundrobin]
nrcmd> policy name addV6Option {opt-name | id}[.instance] value [-blob] [-roundrobin]
nrcmd> policy name setVendorOption {opt-name | id} opt-set-name value [-blob]
nrcmd> policy name setV6VendorOption {opt-name | id} opt-set-name value [-blob]
```

ポリシーのオプションを一覧表示するには、次のいずれかのコマンドを使用します。

```
nrcmd> policy name listOptions
nrcmd> policy name listV6Options
nrcmd> policy name listVendorOptions
nrcmd> policy name listV6VendorOptions
```

サブオプション値を追加するには、「」を参照してくださいサブオプションの複雑な値の追加（208 ページ）。

DHCPv6 オプションの設定

プレフィックスのポリシー (埋め込みまたは名前付き) を作成または編集する場合は、DHCPv6 オプションとベンダー オプションを設定します。(プレフィックスDHCPv6 ポリシー階層 （202 ページ）の埋め込みポリシーまたは名前付きポリシーで使用する場合の v6-options および v6 ベンダー オプション ポリシー属性の特別な処理については、を参照してください。

Cisco Prime Network Registrar は、少なくとも 10,000 バイトまでのオプションをサポートします。

（注） DHCP サーバーのパケットサイズを大きくし、クライアントに配信するために IPv6 のフラグメンテーションが必要となるためにネットワークの問題を引き起こす可能性があるため、非常に大規模なオプションを使用することは推奨しません。大規模なデータセットで通信する必要がある場合は、クライアントが HTTP を介して情報を取得できる URL や、大規模なデータ交換用に設計された他の送信メカニズムを提供するなど、他のメカニズムを検討してください。

ローカル アドバンスド Web UI

DHCPv6 オプションは、[DHCP ポリシーの一覧/追加] ページまたは [DHCP ポリシーの編集] ページの DHCPv4 オプションと共に共存します。ベンダーオプションは、これらのオプションを作成した場合にのみ表示されますDHCP オプション定義セットとオプション定義の作成（210 ページ）(「」を参照してください)。

ドロップダウン リストからオプションを選択することができます。オプションの説明が存在する場合は、[名前] と [番号] の見出しの下に表示され、クリックしてエントリを並べ替えることができます。

CLI コマンド

policy名前setV6Optionを使用 {opt-name|id}[.インスタンス]値[-blob] [-roundrobin] またはpolicy名前setV6VendorOption{opt-name|id}opt-set-nameの値[-blob]-roundrobinが有効な場合、DHCP サーバーは、異なる回転順序で複数の値を含むオプションデータを返すように指示します。特定のクライアントは常に同じ順序を取得しますが、異なるクライアントは、クライアント識別子に基づいてオプションに対して構成された複数の値の順序の異なる「ローテーション」を取得します。オプションの設定には、オプション名 (または ID) と値が必要です。次に例を示します。

```
nrcmd> policy dhcpv6-policy setV6Option dns-servers 2222::1,2222::2

nrcmd> policy foo setV6VendorOption 17 dhcp6-cablelabs-config "(32 2222::3,2222::4)"
```

オプション定義データ型と繰り返し回数

使用できるデータ型の値を次の表に示します。

表 *23*：オプション定義データ型

符号なし 8 ビット AT_INT8	符号なし 16 ビット AT_SHORT	符号なし 32 ビット AT_INT	文字列 AT_STRING
AT_SINT8署名 8 ビット	AT_SSHORT署名された16ビット	AT_SINT署名された32ビット	文字列 AT_NSTRING (終了なし)
DNS 名を AT_DNSNAMEする	AT_SHRTI符号なし 16 ビット (インテル)	AT_INTI符号なし 32 ビット (インテル)	バイナリ AT_BLOB

相対 DNS 名 AT_RDNSNAME	AT_SSHRTI署名済み 16 ビット (インテル)	AT_SINTI署名された 32 ビット (インテル)	AT_DATE日
AT_VENDORクラスのベンダークラス	AT_IPADDR IP アドレス	AT_BOOLブール値	未署名時刻 AT_TIME
AT_VENDOR_NOLEN ベンダーノレン	IPv6 アドレスを AT_IP6ADDRする	AT_MACADDR MAC アドレス	AT_STIME署名時刻
AT_VENDOR_OPTS vendor-opts	AT_ZEROSIZEゼロサイズ	IPv6 可変レッグ接頭部 AT_VPREFIX	

これらのタイプは、を使用option listtypesして CLI で表示できます。

繰り返し回数を設定するには、繰り返しカウント属性を次のいずれかに設定するか、絶対数を入力します。

- ZERO_OR_MORE：Web UI の0+
- ONE_OR_MORE：Web UI の1+
- EVEN_NUMBER：Web UI の2n

たとえば、CLI では次のコマンドを使用します。

```
nrcmd> option 200 ex-opt-def-set set repeat-count=ZERO_OR_MORE

nrcmd> save
```

サブオプション定義の追加

[DHCPオプション定義の編集（Edit DHCP Option Definition）] ページで Add Suboption Definition をクリックして、オプション定義のサブオプション定義を設定できます。[DHCPオプション定義の編集（Edit DHCP Option Definition）] ページが開き、オプション定義と同じ値を追加できます。作成するサブオプション定義は、その親オプション（または親サブオプション）定義に関連付けられます。最大6つのオプションレベルとサブオプション・レベルを定義できます。

（注） サブオプション定義は、Web UI のみを使用して追加できます。現在、CLI を使用して実行することはできません。

サブオプション定義形式は、パックまたはタイプ/長さ/値 (TLV) にできます。

- Packed：ID値がゼロで暗黙的なデータ・タイプを持つサブオプション。オプション値はパケット内の唯一のデータです。DHCPv6オプションは、ほとんどすべてパックされたデータで定義されます。タイプまたは長さのマーカーはなく、データのレイアウトはオプション定義に固有です。パック・サブオプションに対してこれ以上のサブオプション定義を持つことはできません。
- TLV：タイプ、長さ、および値を含む値が 1 から 255 (または 65535) のサブオプション。パケット内のデータの種類と長さが、値の前にあります。

ほとんどの場合、同じオプションに対して TLV サブオプションを含む混合は行いません。

（注） DHCP サーバーはサブオプション 0 の定義 (DHCPv4 vendor-encapsulated-options (43) および v-i-vendor-opts (125) オプション、および DHCPv6 vendor-opts (17) オプション) をサポートしていません。ID 値がゼロのサブオプションは、上記のようにパックされたデータを指定するために DHCP サーバーによって使用されます。

ポリシーの編集時にサブオプション値を入力するには、「サブオプションの複雑な値の追加（208 ページ）」を参照してください。

オプション定義セット

オプション定義セットのインポートとエクスポート

オプション定義セットのインポートとエクスポートは、サーバー間でコピーする方法です。CLI では、importoption-setファイルと名前ファイルを使用してオプションセットをインポートexportoption-setおよびエクスポートできます。

たとえば、プレブート実行環境 (PXE) クライアントのオプション セットをインポートするには、次のように、/examples/dhcp ディレクトリにあるサンプル ファイルを変更してインポートします。

```
nrcmd> import option-set /examples/dhcp/OptionSetPXE.txt
```

注意 組み込みオプション定義セット (dhcp-config や dhcp-cablelabs-config など) をエクスポートしてから、再インポートしないでください。TAC 支援なしで編集済みオプション定義セットを再インポートすると、サーバーが異常終了する可能性があります。

ファイル形式のガイドラインには、次のようなものがあります。

- ファイル内のバージョン文字列は、インポートユーティリティのバージョンと一致する必要があります。
- このユーティリティは、ファイル内の最初のオプション定義セットのみをインポートします。
- 角かっこ ( ){ }を使用してオブジェクトを区切る 、かっこ ( )( )を使用する属性、[ ]角かっこ ( ) を使用して属性内のオブジェクトの一覧を区切ります。引用符 (" ") を使用して文字列値の属性を区切ります。

テキストファイルを編集して、オプション定義セットに若干の変更を加えることもできます。Cisco プライムネットワーク レジストラーは、例/dhcp ディレクトリ、OptionSetJumpStart.txt および OptionSetPXE.txt に 2 つのサンプル オプション定義セット テキスト ファイルを提供します。

- OptionSetJumpStart.txt: ベンダー オプション文字列を編集して、JumpStart クライアントが送信する dhcp クラス識別子 (オプション 60) と一致させます。
- OptionSetPXE.txt:ベンダーオプション文字列を編集して、ブート前実行環境(PXE)クライアントが送信するdhcpクラス識別子(オプション60)と一致させます。

オプション定義セットのローカル クラスタへのプッシュ

地域クラスターから作成したオプション定義セットを、任意のローカルクラスターにプッシュできます。特定のオプション定義セットをクラスタにプッシュする場合は、[DHCPPush Option Definitionオプション定義セットの一覧/追加] ページで [セット] をクリックします。

このページでは、プッシュするデータ、ローカルクラスターと同期する方法、およびプッシュ先のクラスターを示します。データ同期モードは次のとおりです。

- 保証（Ensure）(プリセット値):既存のデータに影響を与えずに、ローカル クラスターに新しいデータが含まれるようになります。
- Replace-ローカルクラスタに固有の他のオブジェクトに影響を与えずにデータを置き換えます。
- Exact-「すべてプッシュ」操作でのみ使用できます。データを上書きし、ローカル クラスターに固有の他のオブジェクトを削除するため、この方法は注意して使用してください。

[使用可能（Available）]フィールドで宛先クラスタを選択し、[選択済み（Available）]フィールドに移動します。

ヒント　同期モードとクラスタ選択の設定は、現在のログインセッションの間は永続的であるため、変更しない限り、このページにアクセスするたびに有効になります。

これらの選択を行った後Push Data to Clusters、 をクリックします。これにより、[プッシュ DHCP オプション定義セット データ レポートの表示] ページが開きます。

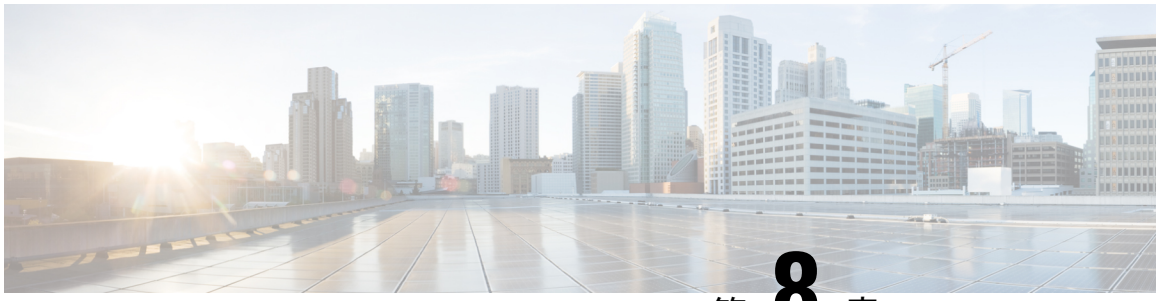
レプリカ データからのオプション定義セットのプル

明示的に作成するのではなく、ローカル クラスタのレプリカ データからオプション定義セットをプルすることもできます。(最初に、クラスタ名の横にある [**レプリケート**] アイコンをクリックして、オプション定義セットのレプリカデータを更新する必要がある場合があります)。Web UI でオプション定義セットをプルするには、Pull Replica Option Definition Sets クリックして [レプリカの DHCP オプション定義セットをプルする] ページを開きます。

このページには、ローカル クラスタのオプション定義セットのリージョン サーバー レプリカ データのツリー ビューが表示されます。ツリーには、ローカル クラスタ用と各クラスタのスコープテンプレート用の2つのレベルがあります。個々のオプション定義セットをクラスタからプルすることも、すべてのオプション定義セットをプルすることもできます。個々のクラスタをプルするには、クラスタのツリーを展開し、Pull Option Definition Set その名前の横にあるをクリックします。クラスタからすべてのクラスタを取得するには、 をPull All Option Definition

Sets from Clusterクリックします。オプション定義セットをプルするには、同期モードも選択する必要があります。

- Ensure-既存のデータに影響を与えずに、リージョン クラスタに新しいデータが含まれることを確認します。
- Replace(プリセット値) - 地域クラスターに固有の他のオブジェクトに影響を与えずにデータを置き換えます。
- Exact- 「すべてプル」操作でのみ使用可能です。データを上書きし、地域クラスターに固有の他のオブジェクトを削除するため、このオプションは慎重に使用してください。



第 8 章

リースの管理

リースは、Dynamic Host Configuration Protocol（DHCP）の中心となるものです。これらは、一定期間、個々のクライアントに割り当てられる IP アドレスです。DHCP サーバーは、有効な IP アドレス範囲を含む適切に構成されたスコープで、これらのリースを自動的に割り当てます。2つのクライアントが同じリースアドレスを持つ可能性はありません。予約とは、常に同じ IP アドレスを取得するリースです。

この章では、ネットワーク内のリースと予約を管理する方法について説明します。

リース ステータス

次の表に、IPv4 または IPV6 のリース状態を示します。

IPv4 リース状態

IPv4 リースは、次の表に示す状態のいずれかになります。

表 *24 : IPv4* リース ステータス

状態	説明
Available	リースできる IP アドレス。
[取得不可（Unavailable）]	見知れない。DHCP使用不可としてマークされているリースの処理 （263 ページ）サーバーがリースを利用不可に設定する方法については、「」を参照してください。
リース済み	クライアントが保持します。
受信コンタクト（Offered）	クライアントに提供されます。
[期限切れ（Expired）]	リース猶予期間が期限切れになったときに使用できます。
非アクティブ化（Deactivated）	リースの期限が切れた後は、更新または再生できません。リースの無効化 （239 ページ）を参照してください。
保留中の利用可能	フェールオーバーに関連する。サーバーがフェールオーバーパートナーと状態を同期するとすぐに、保留中の状態のリースが使用可能になります。 DHCP フェールオーバーの管理 （65 ページ）を参照してください。

IPv6 リース状態

リースは、次の表に記載されている状態のいずれかになります。

表 *25 : IPv6* リース ステータス

状態	説明
Available	リースできる IP アドレス。
受信コンタクト（Offered）	クライアントに提供されます。
リース済み	クライアントが保持します。
[期限切れ（Expired）]	リース猶予期間が期限切れになったときに使用できます。
[取得不可（Unavailable）]	見知れない。何らかの競合のために利用できなくなりました。
[解放（Released）]	クライアントはリースを解放しましたが、サーバーはリースに猶予期間を適用するように構成されています。猶予期間が切れるまで、リースは利用できません。

状態	説明
その他利用可能	フェールオーバー関連。フェールオーバー パートナーによる割り当てには使用できますが、このサーバーでは割り当てには使用できません。
保留中の利用可能	フェールオーバーに関連する。サーバーがフェールオーバー パートナーと状態を同期するとすぐに、保留中の状態のリースが使用可能になります。プレフィックス委任リースのみに使用されます。
保留中の削除	フェールオーバー関連。保留中の削除状態のリースは、サーバーがフェールオーバー パートナーと状態を同期するとすぐにクライアントとの関連付けを解除されます。

リース期間のガイドライン

リース時間に適切な値を定義するには、ネットワーク上で次のイベントを検討します。

- DHCP オプションおよびデフォルト値に対する変更の頻度。
- IP アドレスを要求するクライアントと比較した、使用可能な IP アドレスの数。
- ネットワーク インターフェイス エラーの数。
- コンピュータがネットワークに追加および削除される頻度。
- ユーザーによるサブネット変更の頻度。

これらのイベントはすべて、クライアントがIPアドレスを解放したり、DHCPサーバーでリースが期限切れになる原因となる場合があります。その結果、アドレスは、再利用のためにフリー アドレス プールに戻る可能性があります。ネットワークで多くの変更が発生する場合、アクティブなネットワークには1~3日、非アクティブなネットワークには4日から10日の間のリース期間が推奨されます。このようなリース時間を割り当てると、クライアントがサブネットから離れるのに合った速さでIPアドレスが再割り当てされます。

もう1つの重要な要因は、接続されているコンピュータに対する使用可能なアドレスの比率です。たとえば、使用可能なアドレスが254個のクラスCネットワークでは、アドレスの再利用の要求が少なく、そのうち40個しか使用されません。このような状況では、2か月などの長いリース期間が適切な場合があります。一度に接続しようとしているクライアントが240~260人いれば、需要ははるかに高くなります。このような場合は、より多くのアドレス空間を構成する必要があります。その前まで、DHCP リース時間を 1 時間以下にしてください。

ヒント リース期間が短くなると、クライアントはリースを頻繁に更新するため、DHCPサーバーを継続的に使用できるようにすることが求められるようになります。DHCPフェールオーバー機能は、このようなレベルの可用性を保証するのに役立ちます。

永続的なリースを持つポリシーを作成する場合は注意が必要です。安定した環境でも、クライアント間で一定の売上高が発生します。ポータブル ホストの追加と削除、デスクトップ ホストの移動、およびネットワーク アダプタ カードの交換が可能です。永続的なリースを持つクライアントを削除する場合、IPアドレスを再利用するためには、サーバー構成に手動で介入する必要があります。管理者の介入なしにアドレスが最終的に回復されるように、6 か月などの長いリースを作成することをおお方が良いでしょう。

リース期間の推奨事項は次のとおりです。

- ケーブル モデムのリース期間を 7 日間 (604800 秒) に設定します。リースはプライベート アドレス空間から取得する必要があり、ケーブル モデムはめったに動かないはずです。
- 顧客宅内機器 (CPE) またはラップトップのリースは、パブリック アドレス空間から取得し、サーバーの負荷を軽減するためにできるだけ長いリースで、ユーザーの人口の習慣と一致する必要があります。
- リース時間を短くするには、より多くのDHCP要求および応答バッファーが必要です。最適なスループットを得るために要求バッファと応答バッファをDHCP要求と応答パケットバッファの設定 （84 ページ）設定します (を参照)。
- リース時間の優先許可ポリシー属性が無効 (通常の既定値) であることを確認して、サーバーがリース期間を決定できるようにします。有効にした場合でも、クライアントは、サーバーに対して構成した時間よりも短いリース時間のみを要求できます。一部のクライアントは、常に固定リース時間 (1 時間など) や以前と同じ時間を要求します。この種の要求は、クライアントが完全なリース時間を取得しなくなるという問題を引き起こし、サーバーに対するトラフィックを増やす可能性があります。
- リースの中間マークの前にリースを更新しようとするクライアントのリース延長を延期します。詳細については、リース拡張の保留 （25 ページ） を参照してください。

リース日の制限

リース日の制限は、次の属性を使用して指定できます。

- lease-retention-max-age
- lease-retention-min-age

リース保持期間-最大年齢属性は、リース時間が制限されている過去 (現在の時刻から) の最長時間を指定します。これは、プライバシー保護のためのデータ保存制限を満たすために使用できます。指定しない場合、リース時間がどの程度前に戻るかに制限は適用されません。リースに対してリース保持制限を適用するには、リース保持期限をゼロ以外にする必要があるだけでなく、個々のリース自体がそのポリシーでリース保持制限属性を設定するポリシーに該当する必要があります。この値は、構成されている場合は 8 時間より大きくする必要があります。0 以外で 8 時間未満に設定されている場合は、8 時間に設定されます。

リース保持期間 -期限属性は、リース時間を制限できる最短の時間を、過去に指定します。その値は、リース保持 - 最大年齢より少なくとも 6 時間少なくする必要があります。この属性が有効で、ゼロ以外の値に設定されている場合、保有期間の制限の対象となるリース時間は、

リース保持期間 -max-ageより古くなることはできません。リース保持期限(max-age)に向けて進むにつれて、過去には定期的にリース保持-最小年齢にリセットされます。この属性の構成は、既定では、リース保持期間 -最大期間より 6 時間少なくなるので、オプションです。また、属性値の差が 6 時間未満の場合は、リース保持期間 -最大年齢から 6 時間を引いた値が使用されます。

リース保持-最小とリース保持-最大年齢の間のリース期間に古い時間を維持するには、いくつかの処理が必要であり、これらの2つの値が近いほど、これらの属性の絶対値に関係なく、この処理が行われる頻度が高くなります。リース保持期間の日数を、リース保持期限の数日前に設定すると、リース保持期間の制限に専念する追加のサーバー処理が最小限に抑えられます。

これらの保存期間に影響を受けるクライアントのポリシーを 1 つ以上変更する必要があります。すべてのクライアントに適用するようにsystem_default_policyでこれを構成できます。しかし、これが問題ではないデバイスがある場合は、より選択的に設定することをお勧めします。この機能を有効にしているクライアントが少ないほど、作業が少ないため、サーバーのパフォーマンスへの影響が少なくなります。

ポリシー属性のリース保持制限は、そのポリシーに関連付けられているクライアントがリース日付の制限の対象かどうかを示します。この属性が有効で、DHCP サーバーのリース保持期限がゼロ以外の値に構成されている場合、このポリシーの対象となるリース期間は、リース保持期間 -最大年齢よりも古くなることはできません。リース保持期限に向けて進むにつれて、過去には定期的にリース保持-最小年齢にリセットされます。

プライバシー保護機能の使用を検討する際に覚えておくべきことは次のとおりです。

- 最初に有効(または特定の再構成)を行った場合、既存のリース履歴レコードは、リース保持限界フラグが設定されていないため、この機能の対象になりません。
- リース履歴のトリミング時間が調整される可能性があります。リース保持期間の上限とリース保存期間の差の約3分の2に設定されています。たとえば、6 時間の既定値を使用すると、トリミングは 4 時間ごとに行われます。
- システムのディスク入出力レートが上昇します。これは、サーバーがアクティブなリースレコードと履歴リースレコードの古い時間を更新する必要があるためです。この影響は、リース保持期間の最大値とリース保存期間の差を大きくすることで、ある程度まで減少します。
- スコープやプレフィックスの削除、範囲の調整などの構成変更が行われると、スコープまたはプレフィックスに関連付けられたリースは孤立したリースになります。これらの孤立したリースは、プライバシー保護の時間制限のためにトリミングされず、処理されません。孤立したリースを削除する必要があります。詳細については、孤立したリースの削除（241 ページ）を参照してください。

DHCPv6 クライアントとリース

DHCPv6 サーバーは、DHCPv4 のクライアントとリースに類似したクライアントとリースをサポートします。以下に、その主な違いを説明します。

- サーバーは、ハードウェア アドレスとクライアント ID を 1 つの一意のクライアント識別子に統合する DHCPv4 概念である DHCP 一意識別子 (DUID) によって DHCPv6 クライアントを識別します。
- DHCPv6クライアントは、複数のリースを持つことができます。つまり、複数のプレフィックスが単一のリンク上にあり、割り当てグループ属性を使用してグループ化されていない場合、サーバーは DHCPv4 のように、1 つのスコープからではなく、使用できる各プレフィックスからリースをクライアントに割り当てます。1つのリンク上の複数のプレフィックスが割り当てグループ属性を使用してグループ化されている場合、サーバーは、プレフィックスアロケーション グループ内で最も優先度の高いプレフィックスから、割 プレフィックス割り当てグループ （160 ページ）り当てグループごとに 1 つのリースのみをクライアントに割り当てます (を参照)。
- サーバーは、最初のリースをDHCPv6クライアントに関連付けると最初に作成し、リースが関連付けられていないときにクライアントを削除します。これは、DHCPv4クライアントが 1 つのリースしか持てることができない点を除いて、DHCPv4 の動作と同じです。
- DHCPv6リースは動的に作成されます。サーバーは、構成時に使用できる可能性のあるすべてのリースを作成するわけではありません。

リースは次の場合に使用できます。

- Nontemporary：長く、かつ再生可能な可能性がある標準IPv6ユニキャストaddressesアドレス。
- Temporary：標準 IPv6 ユニキャスト アドレスですが、有効期間が非常に限られています (addresses更新不可能)。一時的なアドレスは、IPv6 (RFC 3041 を参照) のプライバシーの問題を解決します。
- Delegated prefixes：プレフィックスの委任に使用されます（RFC 8415 を参照）。

リースには、優先存続期間と有効な有効期間の両方があります。

- Preferred：主にクライアントを使用する場合、有効なアドレスが優先lifetimeされる時間。優先存続期間が満了すると、アドレスは非推奨になります。
- Valid：クライアントとサーバーの両方で使用される、アドレスが有効な状態のままのlifetime時間です。有効期間は推奨期間より長いか、または同じである必要があります。有効期間が切れると、アドレスは無効になります。有効な有効期間が切れると、リースは削除される資格があります。これは、DHCPv4 のリース時間と基本的に同じです。

関連項目

DHCPv6 バインディング （233 ページ）
リース アフィニティ （233 ページ）
リースのライフ サイクル （233 ページ）
DHCPv6 リース予約 （251 ページ）

DHCPv6 バインディング

バインドは DHCPv6 の新機能であり、複数のアドレス グループをクライアントに割り当てることができます。クライアント・バインディングは、次の 3 つのタイプのいずれかで構成されます。

- 一時的でない (IA_NA)
- 一時(IA_TA)
- プレフィックス委任（IA_PD）

バインディングは、一意の ID アソシエーション ID (IAID) から構成されます。リースは常にバインディングの下に存在します。したがって、クライアントには 1 つ以上のバインディングがあり、バインディングには 1 つ以上のリースがあります。サーバーは、最初にリースを追加するときにバインディングを作成し、それ以上リースがない場合はバインディングを削除します。最初のバインディングを追加するときにサーバーはクライアントを作成し、バインディングがなくなったときにクライアントを削除します。

リース アフィニティ

DHCPv4 の場合、リースが期限切れになった場合、またはサーバーがリースを解放すると、サーバーは、別のクライアントに割り当てられていない限り、そのアドレスに対してクライアントを記憶します。DHCPv6 の場合、IPv6 アドレス空間が大きいため、アドレス生成手法によっては、アドレスを別のクライアントに再割り当てする前に eons が渡される可能性があります。したがって、Cisco Prime Network レジストラーは、有効期限前に更新を要求しなくてもクライアントが同じアドレスを取得できるように、アフィニティ期間属性を提供します。

アフィニティ期間は、一部の環境では望ましいが、アフィニティー時間がゼロまたは非常に小さい場合には望ましくありません。アフィニティー期間中、リースは AVAILABLE 状態で、最後にリースされたクライアントに関連付けられます。この期間中にクライアントがリースを要求した場合、サーバーは同じリースを許可します (または、更新が禁止されている場合、クライアントはそのリースを明示的に取得しません)。

リースのライフ サイクル

リースには、州によって制御されるライフサイクルがあります。リースはクライアントに関連付けられている間のみ存在し、サーバーはそのクライアントに関連付けられていないと削除します。ライフサイクルと状態遷移は次のとおりです。

1. リースが生まれ、サーバーが次の場合にアドレスに関連付けられます。
    1. リースの予約を作成し、リースを AVAILABLE 状態にして、RESERVED としてマークします。この状態に関連付けられているタイマーはなく、サーバーは予約されている限りリースを削除しません。
    2. クライアントに ADVERTISE メッセージを送信し、リースを提供状態にします。リースは、オファーのタイムアウト後に DELETED 状態に移行します。

3. クライアントに応答メッセージを送信し (要求、書き換え、または REBIND の場合)、リースをリース状態にします。リースの有効期間が経過すると、リースは期限切れ状態に移行します。

2. 提供されたリースは次の処理に移行します。
   1. LEASED 状態は、サーバーが REQUEST メッセージを受信し、リースの有効期間が経過した後に期限切れ状態に遷移します。
   2. 提供時間が経過した場合の DELETED 状態。
3. リースリース:
   1. サーバーが要求、書き換え、または REBIND メッセージを受信すると、更新されます。リースの有効期間が新たに経過した後、リースは期限切れ状態に移行します(新しい有効な有効期間は 0 である可能性があります)。
   2. サーバーが RELEASE メッセージを受信すると、RELEASE 状態に遷移します。リースは、リリース猶予期間が経過した後に AVAILABLE 状態に移行します。
   3. サーバーが辞退メッセージを受信すると、UNAVAILABLE 状態に遷移します。サーバーは、タイムアウト時間が経過した後にリースを削除します。
4. 期限切れリースは、猶予期間の後にいずれかの利用可能な状態に移行します。サーバーは、アフィニティー期間が経過した後にリースを削除します。
5. 利用可能なリース:
   1. DELETE 状態に遷移し、サーバーは、アフィニティ期間が経過した後、メモリとリース データベースから削除します。
   2. [予約済み] の場合は削除できず、使用可能なまま残ります。
6. サーバーは LEASED、EXPIRED、RELEASED、または AVAILABLE リースをクライアントに再提供できますが、現在の状態のままですが、タイムアウトは少なくともオファータイムアウトまで延長されます。

DHCP フェールオーバーは、一般的にパートナーが認識するまでこれらの遷移が発生する可能性がある状態遷移の一部を複雑にします。追加のライフサイクルと状態の遷移 (フェールオーバー関連) は次のとおりです。

- AVAILABLE (または他の AVAILABLE) 状態に移行するには、パートナーが移行を確認する必要があるため、承認がパートナーから受信されるまで、PENDING AVAILABLE 状態が使用されます。
- クライアントからのリースの関連付けを解除するには、パートナーからの確認応答も必要であり、したがって、パートナーが状態変更を確認するまで、PENDING DELETE 状態が使用されます。

スコープでのリースの設定

スコープの IP アドレス範囲を設定した後、DHCP 割り当てから生じるリースを監視および調整できます。

リースの表示

リースを表示するには、スコープ内でCisco Prime Network Registrar 11.0 クイックスタートガイドスコープの管理 （137 ページ）IP アドレスの範囲を作成する必要があります。

ローカルの基本 Web UI

[**デザイン**] メニューのScopes[DHCPv4]サブメニューの下で [DHCP スコープの一覧/追加] ページを開き、スコープの[**リース**]タブをクリックします。ページが開き、各リースをクリックして管理できます。

「 リース ステータス （227 ページ）状態」列の値の説明については、「」を参照してください。リースの有効期限に関するガイドラインについては、をリース期間のガイドライン （229 ページ）参照してください。

[DHCP スコープの編集] ページを開くには、リース IP アドレスをクリックします。

ローカル アドバンスド Web UI

メニューからDesignScopesDHCPv4サブメニューの下で選択し、[DHCP スコープの一覧/追加] ページを開きます。その後、スコープの[**リース**]タブをクリックします。または、スコープの名前をクリックして [DHCP スコープの編集] ページをLeases開き、ページのタブをクリックします。

CLI コマンド

[vpn-name/]ipaddr showを使用してlease、IP アドレスに基づいて特定のリースのプロパティを表示します。scope 名前付 listLeasesきスコープのすべてのリースを表示するには、名前を使用します。出力は両方のコマンドでほぼ同じです。特定の仮想プライベート ネットワーク (VPN) でリースを一覧表示できないことに注意してください。すべての VPN のすべてのリースがリストに表示されます。

リースに関連付けられた最新の MAC アドレス、または MAC アドレスに関連付けられているリースを表示できます。[vpn-name/]addr macaddrコマンドはlease、リースが予約されているかアクティブであるかにかかわらず、リースの MAC アドレスを表示します。lease list –macaddr addr [-vpn=vpn-name] コマンドは、その MAC アドレスの IP アドレスがアクティブにリースされた (予約されていない) 場合にのみリース データを一覧表示します。lease list -lansegmentまた、ADDRマスクおよびaddrマスクコマンドを使用してlease list -subnet、LAN セグメントおよびサブネット別のリースを一覧表示することもできます。

リース データのインポートとエクスポート

CLIを使用して、テキストファイルに対してリースデータをインポートしたり、テキストファイルからエクスポートしたりできます。

前提条件のインポート

リースをインポートする前に、次の構成手順を実行する必要があります。

1. インポートするリースの DHCP サーバーでスコープを構成します。
2. リースのホスト名をインポートの一部として DNS に動的に入力する場合は、DHCP サーバーからの動的更新を許可するように DNS サーバーのゾーンを構成します。
3. DHCP サーバーをインポートモードに設定して、リースインポート中に他のリース要求に応答しないようにします。
4. すべての時間フィールドに対して、1970 年 1 月 1 日の GMT の午前 0 時からの秒数、または日、月、日付、時刻、年の形式 (2002 年 4 月 15 日 15:35:48) のいずれかを使用します。
5. リースをインポートした後、DHCP サーバーをインポートモードから外し、他のリース要求に応答できるようにします。

（注） 永続リースオプションを無効にすると、永久リースのインポートは失敗します。必要に応じて policy name enable permanent-leasesを使用してこのオプションを有効にします。

インポートとエクスポート コマンド

コマンドとimport leasesコマンドは、特殊なファイル形式を使用します。 export leases ファイル内の各レコードまたは行は、1 つの DHCP クライアントを表します。

```
field-1|field-2|field-3|...|field-13
```

垂直線 (|) 区切り文字とフィールド値の間にスペースを使用しないでください。少なくとも最初の4つの必須フィールドを含める必要があります。さらに値を指定する場合は、13個のフィールドが存在するように、残りのNULLフィールドをすべて垂直線 (|) で区切る必要があります。フィールドは次の順序で示されます。

1. aaの MAC アドレス :bb:cc:dd:ee:ff形式 (必須)
2. MAC アドレス タイプ（必須）
3. MAC アドレスの長さ (必須)
4. ドット付き 10 進形式の IPアドレス、.b.c.d (必須)
5. リース開始時間 (グリニッジ標準時、GMT) (オプション)
6. リース有効期限 (GMT) (オプション)
7. 許容延長時間 (GMT) (オプション)
8. 最終トランザクション時間 (GMT) (オプション)
9. DHCP サーバーの IP アドレス（任意）
10. ホスト名（ドメイン無し）（任意）
11. ドメイン名（任意）
12. クライアント ID (オプション)
13. VPN 名 (省略した場合は、グローバル VPN が使用されます)

すべての時間フィールドに対して、1970 年以降の秒数または日月-日付/時刻の年形式(例:2007 年 4 月 9 日(月 9/16:35:48))を使用します。

リースをインポートする場合、DHCPサーバーがリースを受け入れないか、通信障害がリースパケットをドロップする可能性があります。後者の場合、サーバーはインポートを数回再試行し、約1分後に失敗を報告します。インポートが失敗した場合は、DHCPサーバーのログファイルを調べて、エラーの原因となったリースを見つけます。インポートファイルに戻り、問題のあるエントリを含めてすべてのリース エントリを削除し、リース のインポートを繰り返します。

を使用export leasesする場合は、現在のリースと期限切れのリースの状態を出力ファイルに書き込むか、現在のリースのみを書き込むか選択できます。次の例は、Cisco Primeネットワーク レジストラー DHCP サーバーからのリース データ エクスポートの一部を示しています。レコード間の空白行は、わかりやすくするために例に表示されます。実際の出力には含まれていません。

例: リースデータエクスポート

```
00:60:97:40:c1:96|1|6|204.253.96.103|Wed Aug 30 08:36:57 2000|Fri Sep 01 13:34:05 2000|
Wed Aug 30 08:36:57 2000|Fri Sep 01 09:34:05 2000|204.253.96.57|nomad|cisco.com|
00:d0:ba:d3:bd:3b|blue-vpn
00:d0:ba:d3:bd:3b|1|6|204.253.96.77|Thu Aug 17 13:10:11 2000|Fri Sep 01 14:24:46 2000|
Thu Aug 17 13:10:11 2000|Fri Sep 01 10:09:46 2000|
204.253.96.57|NPI9F6AF8|cisco.com|blue-vpn
00:d0:ba:d3:bd:3b|1|6|204.253.96.78|Fri Jun 23 15:02:18 2000|Fri Sep 01 14:11:40 2000|
Fri Jun 23 15:02:18 2000|Fri Sep 01 09:56:40 2000|
204.253.96.57|JTB-LOCAL|cisco.com|blue-vpn
```

インポート ファイルのリース期間

リース インポート要求の場合、DHCP サーバーが次の場合は、次のようになります。

- インポートモードで有効になっており、リースがまだクライアントにリースされていない場合、サーバーはクライアントが指定したリース時間を受け入れます。
- インポートモードでは、リースは既にクライアントにリースされ、サーバーに対して遅延リースエクステンションが有効になり(デフォルト)、要求は更新時刻(T1)より前に到着します。

  要求が T1 の後に到着すると、サーバーはクライアントに要求されたものを何でも与えます。有効期限から約 2 分以内に、遅延リース延長は動作しません。

- インポートモードに対して有効になっていませんが、サーバーで構成された時間よりも長いリース時間を受け入れることはありません。
  - 要求に適用可能なポリシーに対してリース時間の優先を許可が有効になっている場合、サーバーはクライアントからのリース時間を短く受け入れます。サーバーエキスパートモードのクライアント要求最小リース時間属性を設定して、リース時間のフロアを作成できる場合でも、リース時間を短くすることは、サーバーに許容されます。
  - 適用可能なポリシーでリース時間の優先を許可する機能が有効になっていない場合、サーバーは着信パケットの dhcp-lease-time 要求を無視し、サーバー設定を使用します。

インポート ファイルに DNS ゾーン名が指定されている場合、サーバーは DNS を更新するときにゾーン名を使用しません。ファイルがホスト名を指定する場合、クライアントまたはクライアント・クラスのエントリーのホスト名指定がホスト名をオーバーライドしない限り、サーバーは DNS の更新時にホスト名を使用します。

クライアントのホスト名は、DNS 更新に使用する DNS 更新構成オブジェクトに関連付けられているゾーン以外のゾーンにする必要があります。これは、クライアントまたはクライアントクラスのエントリでゾーンを指定することによってのみ、DHCP サーバーに表示できます。

アドレス提供前のホストへの ping 実行

DHCP サーバーでインターネット制御メッセージ プロトコル (ICMP) エコー メッセージ機能 (別名ping) を使用して、IP アドレスに応答するユーザーがいるかどうかを確認してから、それを割り当てる(ping-clients属性を使用) することができます。ping-clients属性は、サーバーがリースを提供する前にアドレスに対して ping を試行するかどうかを制御します。有効にした場合は、ping タイムアウト属性も設定する必要があります。このテストにより、DHCP サーバーは、アドレスを割り当てる前に、そのアドレスが使用されていないかどうかを確認できます。

をping使用すると、2 つのクライアントが同じアドレスを使用するのを防ぐことができます。クライアントが ping に応答すると、DHCP サーバーはそのアドレスを利用不可としてマークし、別のアドレスを提供します。このテストは、パワーアップされたクライアントに対してのみ機能します。クライアントがリースを持ち、電源を切ることは可能です。

DHCP サーバーでpingクライアント属性を構成することもできます。この属性は、スコープで明示的に構成されていない場合、スコープのping-clients属性の既定値を制御します。

（注） スコープを構成している場合は、スコープ固有の構成が優先されます。明示的な構成を持たないスコープは、グローバル設定を前提としています。

ping タイムアウト期間は重要です。ping は、特定の IP アドレスを使用しているクライアントがないことを確認するのに役立つため、各 ping はタイムアウト期間全体を待機する必要があります。この ping タイムアウト期間はオファーの前に来るので、指定された時間はサーバーのパフォーマンスに大きな影響を与えます。

- この時間を長く設定しすぎると、リースオファリングプロセスが遅くなります。
- この時間を短く設定しすぎると、IP アドレスを使用して別のクライアントを検出する ping パケットの有効性が低下します。

IP アドレスを提供する前に ping ホストを実装するには、次の方法でスコープを変更します。

- pingクライアント属性を有効にします。この機能はデフォルトでは無効になっています。
- pingタイムアウト属性を設定しています。デフォルトでは 300 ミリ秒です。

サーバーは、正常な ECHO 応答を受信する IP アドレスを使用できなくなります。DHCP サーバー属性のignore-icmp-errors (プリセット値) を有効にすることで、このアクションを制御でき

ます。DHCP サーバーは、IP アドレスを使用不可にする理由として、ICMP DEST_UNREACHABLEを使用し、ICMP ECHO 要求を送信した後に受信するエラーメッセージをTTL_EXPIREDします。

リースの無効化

リースを非アクティブ化すると、クライアントはリースから移動します。リースが使用可能な場合、このリースを非アクティブ化すると、DHCPサーバーがクライアントにリースを渡すことを防ぎます。リースがアクティブ (クライアントによって保持されている) の場合、非アクティブ化すると、クライアントがリースを更新し、サーバーが別のクライアントにリースを渡すことを防ぎます。リースを非アクティブ化できるのは、サーバーが実行中の場合だけです。DHCP サーバーは、リースを直ちに非アクティブ化します。

ヒント Windows クライアントがリースを強制的に解放するには、ipconfig /release をクライアントマシンで実行します。

（注） DHCPv4 リースの場合、リースは再びアクティブ化されるまで非アクティブ化されたままになります。DHCPv6 リース (アドレスまたはプレフィックスの委任) の場合、クライアントがリースから削除されると、リースが自動的にアクティブになるという動作が少し異なります。したがって、DHCPv6 非アクティブ化リースをアクティブ化する必要はありません。ただし、これは、現在のリースが終了した後にリースが使用可能であり、クライアントに関連付けられていないリースを非アクティブ化できないことを意味します。DHCPv6 予約が非アクティブ化された場合、その予約を再度使用するためには、その予約をアクティブにする必要があります。

ローカルの基本 Web UI または 高度な Web UI

リースを非アクティブ化するには、[スコープ] の [**リース**] タブでリースのアドレスリースの表示 （235 ページ）をクリックしDeactivate(を参照) 、[ ] をクリックします。リースが非アクティブ化として表示されるようになりました。リースを再アクティブ化するには、Activateをクリックします。同様の方法で、DHCPv6 リースを非アクティブ化することもできます。

CLI コマンド

リースを非アクティブ化するにはlease、[vpn-name/]ipaddr deactivateを使用します。リースを再アクティブ化するには、[lease vpn-name/]ipaddractivateを使用します。

DHCPv6 リースを非アクティブ化するには、lease6 [vpn-name/]addr **非アクティブ化**を使用します。DHCPv6 リースを再アクティブ化するには、lease6 [vpn-name/]addr **アクティブ化**を使用します (ただし、クライアントがリースから削除されたときに自動的にこれが行われるため、DHCPv6 リースは通常再アクティブ化する必要はありません)。

範囲からのリースの除外

IPアドレス範囲は、定義上、連続している必要があります。既存の範囲からリースを除外するには、範囲を2つに分割する必要があります。新しい範囲は、元の開始範囲と終了範囲のアドレスと除外するアドレスの間のアドレスで構成されます。

注意 除外されたアドレスに現在アクティブなリースがある場合は、まず のリースの無効化 （239ページ）手順に従って、そうでない場合は警告メッセージが表示されます。アクティブなリースを削除すると、削除されたアドレスが後で再構成され、再割り当てされた場合、重複するIPアドレスが生じる可能性があります。サーバーを再ロードした後、リースに関する情報は存在しなくなります。

ローカルの基本 Web UI

スコープ アドレス範囲からリースを除外するには、次の手順を実行します。

ステップ 1 Design メニューで、[DHCPv4] サブメニューから Scopes を選択し、[DHCPスコープの一覧/追加（List/Add DHCP Scopes）] ページを開きます。

ステップ 2 [スコープ] ウィンドウでスコープの名前をクリックして、[DHCP スコープの編集] ページを開きます。

ステップ 3 [範囲（Ranges）] 領域で、削除する IP アドレス範囲の横にある [削除（Delete）] アイコンをクリックします。

ステップ 4 除外された IP アドレスの直前に終了する範囲を追加します。

ステップ 5 除外された IP アドレスの直後に始まる別の範囲を追加します。

ステップ 6 [保存（Save）] をクリックしてスコープを保存します。

ステップ 7 DHCP サーバーをリロードします。

ローカル アドバンスド Web UI

スコープ アドレス範囲からリースを除外するには、基本モードと同じ操作が存在します。

CLI コマンド

スコープ アドレス範囲からリースを除外するには、リース範囲 (scope name listRanges) を検出 lease し、リースを非アクティブ化します ([vpn-name/]ipaddr deactivate) 、 その IP アドレス scope(name removeRange start end) の範囲だけを削除します。その後、結果の範囲が適切に分割されます。

次の例では、範囲から 192.168.1.55 アドレスを削除します。リースが VPN が定義されたスコープ内にある場合は、セッションに対して VPN を明示的に定義するか、または VPN プレフィックスを lease コマンドに含めることができます。

```
nrcmd> session set current-vpn=red
```

```
nrcmd> scope examplescope1 listRanges

nrcmd> lease red/192.168.1.55 deactivate

nrcmd> scope examplescope1 removeRange 192.168.1.55 192.168.1.55

nrcmd> scope examplescope1 listRanges
```

孤立したリースの削除

孤立したリースを削除するには、次の手順を実行します。

始める前に

スコープやプレフィックスの削除、または範囲の調整などの設定変更が行われると、スコープまたはプレフィックスに関連付けられているリースが孤立したリースになります。これらの孤立したリースは、日付の制限に違反しないように定期的に更新されません。

リース日付制限機能を使用する場合は、孤立リースが存在しないようにします (または定期的に消去します)。

ステップ 1 DHCP 属性の削除-孤立リースを有効にする:

```
nrcmd> dhcp enable delete-orphaned-leases
```

ステップ 2 DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

ステップ 3 DHCP 属性の削除-孤立リースの設定を解除する:

```
nrcmd> dhcp unset delete-orphaned-leases
```

ステップ 4 DHCP サーバーをリロードします。

```
nrcmd> dhcp reload
```

サーバー全体のリースの検索

Cisco プライムネットワーク レジストラーを使用すると、サーバー全体でリースを検索できます。検索は、ネットワーク用に構成された1つ以上のリースを対象とするリース属性の組み合わせを指定できるフィルターメカニズムです。リース履歴検索機能はローカルおよび地域の両方のクラスターで使用できますが、アクティブなリース検索機能はローカル クラスターでのみ使用できます。検索機能は、DHCPv4 と DHCPv6 のリースに対して個別に提供されます。

Cisco プライムネットワーク レジストラーを使用して、アクティブなリースを検索することもできます。

ローカル アドバンスド Web UI

DHCPv4 リースを検索するには、次の手順を実行します。

ステップ 1 メニューからOperateサブDHCPv4 Current Leases メニューの下をReports選択して、[DHCP リース検索] ページを開きます。

（注） DHCP リース検索ページを開くには、[DHCP リース履歴検索] ページの [検索] ボタンをクリックします (Reportsサブメニューの [DHCPv4 **リース履歴**] を選択して [DHCP リース履歴検索] ページを開きます)。このボタンをクリックすると、リース履歴検索ページとアクティブなリース検索ページを切り替えることができます。

ステップ 2 アドレスなど、ドロップダウン リストから、[フィルタ属性（Filter Attribute）] を選択します。DHCPv4 と DHCPv6 には、フィルター属性の個別のリストがあります。また、アクティブリースと履歴リースでは、フィルタ属性のセットが異なります。

属性は、要素として選択するとグレー表示されます。

ステップ 3 ドロップダウン リストから、フィルタ タイプを選択します。少なくともバイナリまたは正規表現を選択できますが、選択したフィルタ属性に応じて、リストに次の 1 つ以上を含めることができます。

- バイナリ - 値はバイナリ表記です。
- [日付の範囲] - 日付値の範囲、日付と時刻から日付と時刻を指定します。
- 整数 - 値は整数です。
- 整数の範囲 - 整数値の起き値から整数値の To 値。
- IP Address：値は IP アドレスです。
- IP 範囲:IP アドレスの [宛先] 値から IP アドレスの値。
- IP サブネット:値は IP サブネットです。
- 正規表現 - 値は正規表現構文の正規表現です。(正規表現の一般的な使用方法については、『』の「管理者の設定」の章 Cisco プライムネットワーク レジストラー 11.0 管理ガイドを参照してください。

ステップ 4 選択したタイプに基づいて値を入力します。フィルタをクリアするには、Clear Filter をクリックします。

ステップ 5 AddクリックElementすると、検索要素が [フィルター要素] リストに追加されます。フィルター表示を展開し、要素の横にある[**削除**] アイコンをクリックすると、要素を削除できます。

ステップ 6 要素のリストを作成したら、それらの要素を検索して、結果を得るための要素をまとめて検索できます。Search をクリックします。

ステップ 7 検索の結果として得られるリースのテーブルを確認し、各アドレス、状態、MACアドレス、ホスト名、フラグ、および有効期限を示します。必要に応じて、ページサイズを変更して、さらにエントリを表示します。リースは IP アドレスで順序付けられます。

ヒント　フィルターエレメントは、検索のために一緒に AND されます。検索結果が期待どおりの結果を得られない場合は、フィルター要素リストをもう一度確認し、結果を妨げる可能性のある要素を削除します。

ローカル アドバンスド Web UI

DHCPv6 リースを検索するには、次の手順を実行します。

ステップ 1　メニューからOperate、DHCPv6 Current Leasesサブメニューの下Reportsで選択し、DHCP v6 リース検索ページを開きます。

DHCPv6 LeaseHistoryサブメニューの下Reportsで選択した場合は、DHCP v6 リース検索ページに移動することもできます。DHCPv6 LeaseHistoryサブメニューの下でReports選択すると、DHCP v6 リース履歴検索ページが表示されます。[DHCP v6 リース検索] ページに移動するには、[検索] ボタンをクリックする必要があります。

ステップ 2　アドレスなど、ドロップダウン リストから、[フィルタ属性（Filter Attribute）] を選択します。

ステップ 3　ドロップダウン リストから、フィルタ タイプを選択します。少なくともバイナリまたは正規表現を選択できますが、選択したフィルタ属性に応じて、リストに次の 1 つ以上を含めることができます。

- バイナリ - 値はバイナリ表記です。
- [日付の範囲] - 日付値の範囲、日付と時刻から日付と時刻を指定します。
- 整数 - 値は整数です。
- 整数の範囲 - 整数値の起き値から整数値の To 値。
- IPv6 アドレス:値は IPv6 アドレスです。
- IPv6 プレフィックス:値は IPv6 プレフィックスです。
- 正規表現 - 値は正規表現構文の正規表現です。（一般的な正規表現の使用方法については、Cisco プライムネットワーク レジストラー 11.0 管理ガイドの「管理者の設定」の章を参照してください）。
- [次の値を含む] - 値は IPv6 アドレスまたはプレフィックスです(IPv6 アドレスでのみ使用できます)。クエリは、指定したアドレスまたはプレフィックスを含むリースを一覧表示します。

ステップ 4　選択したタイプに基づいて値を入力します。フィルタをクリアするには、Clear Filter をクリックします。

ステップ 5　AddクリックElementすると、検索要素が [フィルター要素] リストに追加されます。フィルター表示を展開し、要素の横にある[**削除**] アイコンをクリックすると、要素を削除できます。

ステップ 6　要素のリストを作成したら、それらの要素を検索して、結果を得るための要素をまとめて検索できます。Search をクリックします。

ステップ 7　検索の結果として得られるリースのテーブルを確認し、各アドレス、状態、MACアドレス、ホスト名、フラグ、および有効期限を示します。必要に応じて、ページサイズを変更して、さらにエントリを表示します。リースは IP アドレスで順序付けられます。

CLI コマンド

DHCPv4 空間でleaselist–macaddrリースを検索するには、mac-addr [–vpn=vpn-name] を使用します。リースの MAC アドレスを指定します。VPN 指定を省略すると、現在の VPN に基づいて検索を行います。

DHCPv4 空間のリースの場合は、次lease listの構文を使用します。

```
nrcmd> lease list [-macaddr=mac-addr] [-cm-macaddr=cm-mac-addr]
       [-reservation-lookup-key=key] [-mac | -blob | -string]]
       [-vpn=vpn-name] [-count-only]
```

DHCPv4 スペース内のリースの場合は、次の lease listbrief 構文を使用します。

```
nrcmd> lease listbrief [-macaddr=mac-addr] [-cm-macaddr=cm-mac-addr]
       [-reservation-lookup-key=key] [-mac | -blob | -string]]
       [-vpn=vpn-name] [-count-only]
```

DHCPv6 空間のリースの場合は、次lease6 listの構文を使用します。

```
nrcmd> lease6 list[-duid=client-id]
       [-lookup-key=key] [-blob | -string]]
       [-reservation-lookup-key=key] [-blob | -string]]
       [-macaddr=mac-addr]
       [-cm-macaddr=cm-mac-addr]
       [-vpn=vpn-name] [-count-only]
```

DHCPv6 スペース内のリースの場合は、次の lease6 listbrief 構文を使用します。

```
nrcmd> lease6 listbrief[-duid=client-id]
       [-lookup-key=key] [-blob | -string]]
       [-reservation-lookup-key=key] [-blob | -string]]
       [-macaddr=mac-addr]
       [-cm-macaddr=cm-mac-addr]
       [-vpn=vpn-name] [-count-only]
```

オプション–macaddrと–cm-macddrオプションは、CableLabs DOCSISvendor-opts オプション (DHCPv6 オプション 17) で識別されるリースを検索することです。たとえば、次の 2 つのコマンドの場合は、次のようになります。

```
nrcmd> lease6 listbrief -macaddr=01:02:03:04:05:06
nrcmd> lease6 listbrief -cm-macaddr=01:02:03:04:05:06
```

–macaddr 回線には、オプション 17 device-id サブオプション (36) に要求された MAC アドレスが含まれているリースがリストされます。–cm-macddr 行には、オプション 17 cm-mac-address サブオプション (1026) が要求された MAC アドレスと一致するリースがリストされます。(これらの番号順の DHCPv6 オプション一覧 （530 ページ）サブオプションの詳細については、を参照してください。

クライアント予約の使用

以前のバージョンの Cisco Prime Network レジストラーのバージョンでは、クライアントが必要とするリースを取得する唯一のオプションは、リース予約の作成 （248 ページ）リース予約を

作成することでした(を参照)。クライアント(ごとに予約を作成するのは必ずしも簡単ではありません。また、Cisco Primeネットワーク レジストラー予約をデータベースに同期するプロセスも非常に複雑です。クライアント予約機能は、この複雑さを軽減するのに役立ちます。

Cisco Primeネットワーク レジストラー DHCP サーバーが DHCPv4 クライアントに IP アドレスを割り当てる際にサポートされている現在の機能は次のとおりです。

- クライアントのリースベースの予約が存在し、リースが使用可能な場合は、その予約が使用されます。
- それ以外の場合、クライアントがアドレスを要求し、そのアドレスが使用可能な場合は、そのアドレスが使用されます。
- それ以外の場合は、クライアントが使用できるスコープの1つからランダムアドレスが使用されます。

クライアント予約機能を使用すると、クライアントエントリ(Cisco Prime Network レジストラーまたは LDAP に直接保存される)または拡張を通じて、アドレスを指定してプレフィックスを委任できます。また、クライアントは複数のスコープまたはプレフィックスに配置でき、サーバーはクライアントの場所に適したアドレスを選択します。

クライアント予約リースは、基本的に予約済みリースです。主な違いは、リースが予約されているクライアントが、クライアントの予約の場合にサーバーに知られていない点です。クライアント予約は、多数のクライアントのリースを構成する場合や、単一のクライアントに対して多数のリースを構成する場合に使用されます。

クライアントの予約は、次の3つの主要なメカニズムのいずれかを使用してCisco Primeネットワーク レジストラーに提供できます。

- 内部クライアント データベースの使用:リース予約と同じ問題がいくつか発生しますが、Cisco Prime Network レジストラー内部クライアント データベースが他の目的で既に使用されている場合は、より良いオプションになる場合があります。内部クライアント データベースが、クライアントを単独で維持する必要があり、予約を維持する必要が生じるためには、リース予約と比較すると、より有利になります。
- LDAP を使用する:Cisco Prime Network レジストラーは、LDAP リポジトリ(Cisco Prime Networkレジストラーの外部)でクライアントを検索することができ、クライアントがクライアント予約を指定する場合があります。
- エクステンションの使用:Cisco Prime Network レジストラーは、エクステンションを使用して外部サーバーまたはデータベースと通信するように設定できます。

Cisco Prime Network レジストラークライアント データベースまたは LDAP 内で維持されるクライアントエントリには、クライアントが使用するはずのアドレスとプレフィックスを含めることができます。クライアント予約を指定する属性は次のとおりです。

1. reserved-addresses- クライアント用に予約されているアドレスのリストを指定します。使用可能なスコープに一致する最初の使用可能なアドレス (予約への制限が有効になっている必要があります) がクライアントに割り当てられます。

2. reserved-ip6addresses：クライアント用に予約されているアドレスのリストを指定します。使用可能なプレフィックスに一致する使用可能なすべてのアドレス (予約に制限が有効になっている必要があります) がクライアントに割り当てられます。
3. reserved-prefixes- クライアント用に予約されているプレフィックスのリストを指定します。使用可能なプレフィックスに一致する使用可能なすべてのプレフィックス (予約制限が有効になっている必要があります) がクライアントに割り当てられます。

（注） 上記の属性は VPN を示すものではなく、(クライアントが接続できる)すべての VPN に適用されます。したがって、VPN でクライアント予約を使用する場合は、予約済みアドレスが適切な VPN でのみ有効であることを確認するか (含まれるスコープまたはプレフィックスが存在し、予約が制限されているすべての VPN に適用されるため)、VPN ごとに一意のクライアントを確保する必要があります。

属性の予約制限は、スコープ、スコープテンプレート、プレフィックス、およびプレフィックステンプレートの各オブジェクトに追加され、クライアント予約を指定します。

LDAP のクライアントの場合、LDAP 属性名と対応するクライアント属性名との間のマッピングをセットアップする必要があります。

LDAP アドレス属性にクライアントの IPv4 アドレスリストが含まれている場合、 ldap servername setEntry query-dictionary ldap-attribute=cnr-client-attribute を使用して、reserved-addresses 属性にマッピングします。次に例を示します。

```
nrcmd> ldap ldap-1 setEntry query-dictionary addresses=reserved-addresses
```

ローカル アドバンスド Web UI

スコープをクライアント予約に制限するには、次の手順を実行します。

1. [**デザイン**] メニューの[DHCPv4]サブメニューの [**スコープ**] を選択して、[DHCP スコープの一覧/追加] ページを開きます。スケジュールを作成するには、「スコープの作成 （138 ページ）」を参照してください。
2. [DHCP スコープの一覧/追加] ページの [その他の設定] グループで、[**予約制限**] 属性を有効にします。

   既存のスコープを変更してクライアント予約を指定するには、必要なスコープ名をクリックして [DHCP スコープの編集] ページを開きます。[その他の設定] グループの[**予約制限**] 属性の[有効] をクリックします。

   フラグクライアント予約は、スコープがクライアント予約に制限されていることを示します。

スコープ テンプレートをクライアント予約に制限するには、次の手順を実行します。

1. [**デザイン**] メニューの[DHCPv4]サブメニューの [**スコープ テンプレート**] を選択して、[DHCP スコープ テンプレートの一覧/追加] ページを開きます。スコープスコープ テンプ

レートの作成と適用 （173 ページ）テンプレートを作成するには、「」を参照してください。

2. [DHCP スコープ テンプレートの一覧/追加] ページの [その他の設定] で [**予約制限**] 属性を有効にします。

   既存のスコープテンプレートを変更してクライアント予約を指定するには、必要なスコープ テンプレート名をクリックして[DHCP スコープ テンプレートの編集]ページを開きます。[その他の設定] グループの[**予約制限**] 属性の[有効] をクリックします。

プレフィックスをクライアント予約に制限するには、次の手順を実行します。

1. [**デザイン**] メニューの[DHCPv6]サブメニューの下にある[**プレフィックス**]を選択して、[DHCP v6 プレフィックスの一覧/追加] ページを開きます。
2. [プレフィックス]ウィンドウの[**プレフィックスの追加**]アイコンをクリックし、プレフィックス名とアドレスを入力して、[IPv6 プレフィックスの追加] をクリックします。
3. [プレフィックス]ペインのプレフィックス名をクリックして、[DHCPv6 プレフィックスの編集] ページを開きます。[親以外の設定] グループの[**予約制限**] 属性を有効にします。

注 予約制限属性が有効になっているプレフィックスは、ライセンスが必要なアクティブリースの合計にはカウントされません。クライアント予約を受信するクライアントは、そのアクティブなリース数をカウントしますが、これは、リースが実際にクライアントによって保持されている場合にのみ発生します。

プレフィックス テンプレートをクライアント予約に制限するには、次の手順を実行します。

1. プレフィックスをクライアント予約に制限するには、[**デザイン**] メニューの[DHCPv6]サブメニューの [**プレフィックス テンプレート**]を選択して、[DHCP v6 プレフィックス テンプレートの一覧/追加] ページを開きます。
2. [プレフィックス テンプレート] ウィンドウの [**プレフィックス テンプレートの追加**] 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。
3. プレフィックス テンプレート名を入力し、[**接頭辞テンプレートを追加**]ボタンをクリックします。
4. [**予約に制限**]属性を有効にする] をクリックします。

   既存のプレフィックステンプレートを変更してクライアント予約を指定するには、クライアント予約に制限するプレフィックス テンプレート名をクリックします。
   restrict-to-reservations 属性に対して [有効（enabled）] をクリックします。

クライアント予約とリース予約の違い

クライアントの予約には、リース予約に関して次のような大きな違いがあります。

- 任意のアドレスに対してクライアント予約が1つだけであることを確認するための検証**はありません**。同じアドレスまたはプレフィックスを指定するクライアントが2つある場合は、どちらのクライアント要求が最初に到着しても、そのリースが許可されます。
- クライアント予約は、クライアントが DHCP 構成を完了した後にのみ、実際に存在します。リース予約は、クライアントトランザクションが発生しない場合でも知られているため、DHCP サービスをまったく提供しないクライアントにも使用できます。

  Cisco Prime Network Registrar は以下をサポートします。

  - 特定の IP アドレスのリース予約を作成する。
  - ケーブル ソース検証がケーブル モデム終端システム(CMTS)で正しく動作する IP アドレスに対して正しいケーブル モデムの MAC アドレスを設定します。

  これは、Cisco Prime Network レジストラー DHCP サーバーが DHCP クライアント トランザクションの前にリース予約を認識し、それらのアドレスに対する CMTS からの leasequery 要求に正しく応答するためです。これに対して、クライアント予約は DHCP サーバーに DHCP クライアント パケットが到着する前に DHCP サーバーに認識されません。クライアント登録のためにクライアント予約として構成された IP アドレスの leasequery は、IP アドレスがクライアント予約であることを (一般に) 認識しません。

  したがって、DHCP サーバーが正の応答を返すはずの leasequery は、クライアントがリースを要求していない場合でも、適切なケーブル モデム MAC アドレスを含む肯定的な結果を返す場合でも、クライアント予約では動作しません。

リース予約の作成

クライアントが常に同じリースを取得するようにするには、リース予約を作成します。リース予約の管理は、ローカル クラスターで dhcp-admin ロールを持つ管理者、または地域クラスターの dhcp 管理サブロールを持つ中央 cfg-admin ロールを持つ管理者のみが使用できます。

サーバーから DHCPv4 および DHCPv6 予約を照会することができます。

（注） すべてのリース予約は、ライセンスされた IP アドレスの数と比較されるアクティブなリースの合計にカウントされます。

DHCPv4 予約

DHCP 編集モードが同期モードの場合、予約変更は自動的に DHCP サーバーに転送され、直ちに有効になります。

編集モードがステージングされると、ローカルクラスタの予約リストに対して行った変更は、親スコープを変更して、サーバーの再ロードが必要であることを示します。地域の予約リストに変更を行うと、親サブネットが変更されます。

ローカルの基本 Web UI

リース予約Designを表示するには、メニューからScopesDHCPv4サブメニューを選択して[DHCP スコープの一覧表示/追加] ページを開き、[**予約**] タブをクリックします。

このページで引当を作成するには、リース用に予約する IP アドレスを入力し、[ルックアップ キー] フィールドにルックアップ キーを入力します。ルックアップ キー エントリに応じて、MAC アドレス(デフォルト)または文字列またはバイナリ ラジオボタンをクリックします。Add Reservation をクリックします。リース IP アドレス、ルックアップ キー、スコープの詳細は、[DHCP 予約の一覧/追加] ページに表示されます。

ローカル アドバンスド Web UI

DHCPv4 スコープのリース予約を表示するには、Designメニューからサブメニューの Scopes下をDHCPv4選択して[DHCP スコープの一覧/追加]ページを開きます。基本 Web UI に関する手順を実行します。

詳細モードでは、スコープに依存しない予約を作成するメカニズムも提供されます。DHCPv4 スコープの予約を直接構成するには、次の手順を実行します。

ステップ 1 メニューからDesignReservationsDHCPv4サブメニューの下で選択し、[DHCP 予約の一覧表示/追加] ページを開きます。

ステップ 2 [予約] ウィンドウの[DHCP **予約の追加**] アイコンをクリックし、リース用に予約する IP アドレスを入力します。

ステップ 3 ルックアップ キー エントリに応じて、MAC アドレス(デフォルト)または文字列またはバイナリ ラジオ ボタンをクリックします。Save をクリックします。

ヒント フィルタを使用して、表示されるリストのサイズを小さくすることができます。これを行うには、[フィルタタイプ（Filter Type）] ドロップダウン リストからフィルタ タイプを選択します。フィルターの値は、フィルターの種類の選択として設定されます。[フィルタの設定（Set Filters）] をクリックします。フィルタ タイプを「None」に設定するには、[フィルタのクリア（Clear Filter）] をクリックします。リースの IP アドレス、ルックアップ キー、およびスコープの詳細は、[DHCP 予約の一覧と追加（List/Add DHCP Reservations）] ページに表示されます。

（**注**） 複数の DHCP サーバーは、DHCP フェールオーバー パートナーでない限り、同じサブネット上に IP アドレスを配布しないでください。フェールオーバーを使用する場合、クライアント予約は各サーバーで同一である必要があります。存在しない場合、リース予約が存在するクライアントは、異なるサーバーから異なる IP アドレスのオファーを受け取ることができます。フェールオーバー同期機能は、パートナーの構成が一貫していることを確認するのに役立ちます。

CLI コマンド

予約コマンドを使用すると、Cisco プライムネットワーク レジストラーの DHCPv4 予約のグローバル リストにアクセスできます。

使用して新しいアドレスを作成します,**予約**[vpn-name/]アドレス**作成**{macaddr |検索キー}[-mac |-**ブロブ**|-**文字列**][属性=値..]

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 create 00:d0:ba:d3:bd:3b
```

使用してアドレスを削除する**予約**[vpn-name/]アドレス**削除**

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 delete
```

を使用して属性を取得する、**予約**[vpn-name/]アドレス**取得**属性

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 get value
```

使用して属性を設定する、**予約**[vpn-name/]アドレス**セット**属性=値

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 prefix=cm_prefix
```

使用して属性を設定解除する、**予約**[vpn-name/]アドレス**の設定解除**属性

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 unset value
```

を使用してアドレスを表示**する予約**[vpn-name/]アドレス**ショー**

次に例を示します。

```
nrcmd> reservation white/192.168.1.110 show
```

予約**リスト**[VPN名/]アドレスを使用して予約を表示する|-**マック**|-**キー**]。このコマンドは、ソート順を変更するために -key が指定されていない限り、予約をアドレス順に表示します。

次に例を示します。

```
nrcmd> reservation list white/192.168.1.110
```

予約の簡単な詳細を表示するには、**予約リストブリーフ**[-macaddr=mac-addr] [-lookup-key=ルックアップキー [-mac |-**ブロブ**|-**文字列**][-vpn=VPN 名][-**カウントのみ**]

次に例を示します。

```
nrcmd> reservation listbrief -lookup-key=d4:6a:a8:d3:e2:ea -mac
```

DHCPv6 リース予約

予約は、非一時アドレスとデリゲートされたプレフィックスにのみ適用されます。これらは、構成内のプレフィックスに関連付けられており、常に、構成済みのプレフィックスオブジェクトの下のアドレス (またはプレフィックス) に対して使用する必要があります。

予約は、別のプレフィックスのオブジェクト範囲内になっていない場合、プレフィックスのオブジェクト範囲の外側に置くことができます。ただし、新しいプレフィックスオブジェクトを追加する場合は、この影響を受けます。プレフィックスの新しい範囲に含まれている予約が存在する場合、プレフィックスは追加されません。これにより、EX_CONFLICTステータスになります。詳細は、リース予約の作成 （248 ページ）を参照してください。

（注） DHCPv4 予約の操作は、アドレスが v4 アドレスではなく v6 アドレスであることを除いて、DHCPv6 予約に似ています。また、DHCPv6 クライアントの主な ID は、MAC アドレスではなく、クライアント DUID です。DHCPv6 予約には、アドレスと委任されたプレフィックスが含まれます。

v6予約リストで行った変更は、親プレフィックスを変更して、サーバーの再ロードが必要であることを示します。地域サーバーでは、DHCP編集モードが同期モードで、親プレフィックスがローカル クラスタに割り当てられている場合、変更は自動的にローカル クラスタに転送されます。これらの変更を有効にするには、サーバーの再ロードが必要です。

注意 複数の DHCP サーバーが同じプレフィックスに IP アドレスを配布する場合、予約は同一である必要があります。存在しない場合、予約が存在するクライアントは、異なるサーバーから異なる IP アドレスのオファーを受け取ることができます。

リース予約は、IP アドレスとルックアップ キーを組み合わせます。検索キーには、文字列値またはバイナリ BLOB を指定できます。

（注） サーバーが再ロードされるときに、既存のリースに競合する (または含まれている) 短いプレフィックスまたは長いプレフィックスを持つ新しいプレフィックス委任予約が追加された場合、予約は既存のリースの読み込みができなくなります。

ローカル アドバンスド Web UI

DHCPv6 プレフィックスの予約を表示するには、次の手順を実行します。

ステップ 1 DHCPv6 リース予約を表示するには、[**設計**] メニューのPrefixesサブメニューDHCPv6の下で [DHCPv6 プレフィックスの一覧/ 追加] ページを開きます。

ステップ 2 [プレフィックス] ペインでプレフィックスを選択し、[**予約**] タブをクリックします。

ローカル アドバンスド Web UI

DHCPv6 プレフィックスの予約を直接設定するには、次の手順を実行します。

拡張モードでは、有効な親プレフィックスが指定されていない場合、CCM サーバーは自動的に適切な親プレフィックスを設定します。

ステップ 1 メニューからDesignReservationsDHCPv6サブメニューの下で選択し、DHCP v6 予約のリスト/追加ページを開きます。

ステップ 2 予約を作成するには、[予約] ウィンドウの[DHCP v6 **予約の追加**] アイコンをクリックし、リース用に予約する IP アドレスを入力し、[検索キー] フィールドにルックアップ キーを入力します。

ステップ 3 [検索キー] フィールドにバイナリ値を入力した場合は、[文字列] ラジオ ボタンをクリックするか、[バイナリ] ラジオ ボタンをクリックします。

ステップ 4 Add Reservation をクリックします。

ステップ 5 [予約] ウィンドウで、[**フィルターの種類**] ドロップダウン リストからフィルターの種類を選択します。[フィルタ値] フィールドに値を入力します。[フィルタの設定（Set Filters）] をクリックします。[フィルタの種類] を [なし] に設定するには、[**フィルタのクリア**] をクリックします。リース IP アドレス、ルックアップ キー、およびプレフィックスの詳細が[DHCP v6 予約の一覧/追加]ページに表示されます。

CLI コマンド

reservation6コマンドを使用すると、Cisco プライムネットワーク レジストラーの DHCPv6 予約のグローバル リストにアクセスできます。

グローバル リストの各予約に一致するプレフィックスが存在する必要があります。

を使用して新しいアドレスを作成します,**予約**6 [vpn-name/]アドレス**作成**ルックアップキーを作成する [-blob |-**文字列**][属性=値..]

次に例を示します。

```
nrcmd> reservation6 white/2001:db8::1 create 00:03:00:01:01:02:03:04:05:06
```

使用してアドレスを削除する,**予約**6 [vpn-name/]アドレス削除

次に例を示します。

```
nrcmd> reservation6 white/2001:DB8::1 delete
```

使用して属性を取得します,**予約**6 [vpn-name/]アドレス**取得**属性

次に例を示します。

```
nrcmd> reservation6 white/2001:DB8::1 get value
```

使用して属性を設定する、**予約**6 [vpn-name/]アドレス**セット**属性=値

次に例を示します。

```
nrcmd> reservation6 white/2001:DB8::1 set prefix=cm_prefix
```

使用して属性を設定解除します,**予約**6 [vpn-name/]アドレス**の設定なし**の属性

次に例を示します。

```
nrcmd> reservation6 white/2001:DB8::1 unset value
```

使用してアドレスを表示する予約6 [vpn-name/]アドレス**ショー**

次に例を示します。

```
nrcmd> reservation6 white/2001:DB8::1 show
```

予約を使用して予約を表示する予約6**リスト**[[VPN名/]アドレス|-**キー**]。このコマンドは、ソート順を変更するために -key が指定されていない限り、予約をアドレス順に表示します。

次に例を示します。

```
nrcmd> reservation6 list white/2001:DB8::1
```

使用して予約の簡単な詳細を表示します,**予約6リストブリーフ**[-**検索キー**=ルックアップキー [-blob |-**文字列**][-vpn=VPN 名][-**カウントのみ**]

次に例を示します。

```
nrcmd> reservation6 listbrief -lookup-key=def -string -vpn=vpn1
```

リースと予約プロパティの詳細設定

高度なリースと予約のプロパティを設定することができます。

- 現在リースされている IP アドレスの予約-現在リース済みのアドレスの予約 （254 ページ）
- リースの予約解除 -を参照してください。リースの予約解除 （255 ページ）
- MAC 以外のアドレスへのリースの延長:を参照してください。MAC 以外のアドレスへの予約の拡張 （256 ページ）
- リースの可用性の強制-「」を参照してください。リースを強制的に使用可能にする （259 ページ）
- リースの更新の抑制- 「」を参照リース更新の抑制 （259 ページ）
- 利用不可とマークされたリースの処理 -「」を参照使用不可としてマークされているリースの処理 （263 ページ）
- 利用できないリースのタイムアウトの設定 -を参照してください。使用不可リースのタイムアウトの設定 （264 ページ）

現在リース済みのアドレスの予約

1 台目のクライアントにリースがある場合でも、別のクライアントに対して再使用している間に、そのクライアントの予約を削除できます。

ローカル アドバンスド Web UI

既存のリースを予約するには、次の手順を実行します。

ステップ 1 メニューのDesignScopesサブメニューの下DHCPv4でを選択し、スコープの名前を選択して [DHCP スコープの編集] ページを開きます。

ステップ 2 [リース] タブをクリックします。

ステップ 3 リースの IP アドレスをクリックします。

ステップ 4 IP アドレスがリースされていない場合 (使用可能な状態)、予約のルックアップ キーまたは MAC アドレスを入力します。

ステップ 5 Make Reservation をクリックします。[DHCP スコープの編集] ページで、リースが予約済みとして表示されます。

ステップ 6 [保存（Save）] をクリックしてスコープを保存します。

ステップ 7 予約を削除するには、[DHCPRemove Reservationスコープの編集] ページをクリックし、スコープを変更します。リースは予約済みとして表示されなくなります。

既存のリース予約の例

このCLIコマンドの例では、既存のリースから予約を作成します。これは、dhcp-editモードが同期に設定され、予約がサーバーに動的に追加されることを前提としています。

```
nrcmd> reservation 192.168.1.110 create 1,6,00:d0:ba:d3:bd:3b
nrcmd> lease 192.168.1.110 activate
```

クライアント 1,6,00:d0:ba:ba:d3:bd:3b は DHCPDISCOVER を行い、192.168.96.110 のオファーを受け取ります。クライアントは DHCPREQUEST を実行し、同じ IP アドレスに対する ACK メッセージを取得します。

時間が経過すると、クライアント 1,6,00:d0:ba:d3:bd:3b は、サーバーが確認する更新されるいくつかの DHCPREQUEST を実行します。次に、クライアントリースの有効期限が切れる前の時点で、予約を終了します。

```
nrcmd> lease 192.168.1.110 deactivate
nrcmd> reservation 192.168.1.110 delete
```

その後、その IP アドレスが最初のクライアントにリースされている場合でも、その IP アドレスに対して別のクライアントの予約を追加します。

```
nrcmd> reservation 192.168.1.110 create 1,6,02:01:02:01:02:01
nrcmd> lease 192.168.1.110 activate
```

このアクションにより、あるクライアントにリースされているが、別のクライアント用に予約された IP アドレスが作成されます。新しいクライアント (1,6,02:02:02:01:02:02:01) が元のクライアント (1,6,00:d0:d0:d3:bd:3b) の前に DHCPDISCOVER を実行した場合、新しいクライアントは 192.168.96.110 を取得しませんが、動的プールからランダムな IP アドレスを取得します。

元のクライアント (1,6,00:d0:ba:d3:bd:3b) が次の DHCPREQUEST/RENEW を 192.168.96.110 のリースに送信すると、NAK メッセージが表示されます。一般に、非確認メッセージを受信すると、クライアントは直ちに DHCPDISCOVER を送信します。DHCPDISCOVER を受信すると、サーバーは 192.168.96.110 の残りのリース時間をキャンセルします。

次に、サーバーはクライアントに 1,6,00:d0:ba:d3:bd:3b 適切なリースを提供します(192.168.96.110 以外の予約、動的リース (使用可能な場合) 、または何も (動的リースが利用できない場合) 。新しいクライアント (1,6,02:01:02:02:02:02:01) が受信したランダム IP アドレスを更新しようとすると、サーバーは予約済みアドレスを指定するため、NAK を送信します。新しいクライアントが DHCPDISCOVER を実行すると、192.168.96.110 予約アドレスが取得されます。

また、リースの可用性を強制することもできます(「」リースを強制的に使用可能にする （259 ページ）を参照)。ただし、これを行っても、元のクライアント (1,6,00:d0:d0:d3:bd:3b) が 192.168.96.110 を使用するのを停止しません。また、新しいクライアント (1,6,02:01:02:01:01:02:02:01) が 192.168.96.110 を取得するのを妨げるわけではありません。つまり、クライアントの予約は、予約が行われる IP アドレスのリース状態 (および実際のリース クライアント) とは無関係です。

したがって、あるクライアントに対して予約を行うと、別のクライアントがリースをすぐに失うわけではありませんが、クライアントは次回 DHCP サーバーに接続する際に NAK 応答を受信します(秒または数日)。また、IP アドレスを予約したクライアントは、他のクライアントが既に IP アドレスを持っている場合、そのアドレスを取得しません。代わりに、次の手順を実行するまで、別の IP アドレスを取得します。

- 受信するはずのIPアドレスは無料です。
- クライアントは更新として DHCPREQUEST を送信し、NAK 応答を受信します。
- クライアントが DHCP ディスカバリを送信します。

リースの予約解除

リース予約はいつでも削除できます。ただし、リースがまだアクティブな場合、クライアントは、有効期限が切れるまでリースを使用し続けます。別のクライアントのリースを予約しようとすると、警告が表示されます。

リージョンから最後の予約を削除すると、予約を選択して変更をローカル クラスターにプッシュすることはできません。親サブネットをプッシュして、予約リストを同期させて、予約のローカル コピーを削除する必要があります。

地域のDHCPv6予約にはプッシュ機能はありません。予約を再同期するには、常に親プレフィックスをプッシュする必要があります。地域削除アクションを同期する場合は、この方法が推奨されます。

ローカル アドバンスド Web UI

リースの予約を解除するには、[**デザイン**] メニューのReservations[DHCPv4]サブメニューの下で [DHCP 予約の一覧/追加] ページを開き、削除する予約を選択した後に [**予約の削除**] アイコン (左ペイン) をクリックします。これにより、予約は確認なしで直ちに削除されます。

CLI コマンド

リースの予約を解除するには、[reservation vpn/]ipaddrdeleteまたはscope name removeReservation {ipaddr |マカドル|検索キー}[–mac |–blob |–string]. それでも、次の対応を試してください。

- nrcmd 内部データベースから予約がなくなっていることを確認します。
- 予約を含むスコープでフェールオーバーを使用する場合:

1. 両方reservationのサーバーで [vpn/]ipaddr delete、scope または name removeReservationを使用します。
2. バックアップ サーバーで、ステージング dhcp 編集モードの場合は、 lease [vpn/]ipaddrdelete-reservationを使用します。
3. メイン サーバーで同じコマンドを使用します。

lease ipaddrdelete-reservationを発行した場合のみサーバー内部メモリに影響するため、この操作の結果を保存して、サーバーの再ロード後も保存します。

MAC 以外のアドレスへの予約の拡張

場合によっては、着信クライアント パケットの MAC アドレス以外のアドレスに基づいてリース予約を作成する必要があります。スイッチ ポートに接続されている DHCP クライアント デバイスは、MAC アドレスに関係なく、同じ IP アドレスを取得する必要があります。この方法は、工場出荷時のデバイスを同一のデバイス(異なる MAC アドレス)で置き換えるが、同じ IP アドレスを維持する場合に役立ちます。

クライアント ID の上書き

Relay エージェント情報オプション (82) からスイッチの MAC アドレスとポートを抽出し、そこからクライアント ID を作成するクライアント クラスオーバーライド クライアント ID属性で式を設定できます。着信パケットのクライアント ID に関係なく、IP アドレスを割り当てる ID は、同じスイッチ ポートを経由して着信するデバイスと同じです。属性に使用する式は、オプション 82 形式によって異なります。DHCP サーバーは、クライアント クラスにパケットを割り当てると式を計算します。オーバーライドクライアント ID値は、その後のクライアントの ID になります。

（注） [v6-] オーバーライド クライアント ID式を使用する場合、クライアント ID による leasequery 要求は、クライアントのリースに関する情報を正しく取得するために、オーバーライドクライアント ID属性を指定する必要があります。

ただし、ポリシーでuse-client-id-for-予約属性を有効にすると、サーバーはその要求のクライアント ID をnn:nn nn...:という形式の文字列に変換します。nn:nnをクリックし、その文字列を使用して予約を検索します。

クライアントまたはクライアントクラスの環境への追加属性は、名前と値のペアとして指定された DHCP 拡張環境ディクショナリ (を参照拡張ポイントの使用 （433 ページ）) に属性値を送信する機能も提供します。クライアントまたはクライアントクラスのどちらでも、環境への追加のディクショナリ属性を構成できます。クライアントとクライアントクラスの両方でこの属性を構成する場合は、クライアントクラスで構成する名前と値のペアとは異なる名前を持つようにする必要があります。同じ環境辞書に入れられます (特定の名前に対して 1 つの値しか持てありません)。一般的に、この属性はクライアントまたはクライアントクラスでのみ構成し、両方で構成しないことをお勧めします。

ローカル アドバンスド Web UI

[DHCPクライアント クラスの編集] ページでオーバーライド クライアント ID 属性をDesign確認できます (メニューから、Client ClassesDHCP Settingsサブメニューの下でクライアント クラスの名前を選択します)。

また、DHCP サーバーのクライアント クラスルックアップ ID を設定して、すべてのパケットを特定のクライアント クラスに入れ、そこでオーバーライド クライアント ID式を設定する必要があります。メニューからOperateManage Servers、サブメニューの下Serversで選択し、ローカル DHCP サーバーのリンクをクリックして、ローカル DHCP サーバーの編集 ページを開きます。クライアント クラス属性に、クライアントクラス検索 ID式を入力します。

予約にクライアント ID を使用するには、[DHCP ポリシーの追加] ページの [クライアントID の予約] 属性を有効にするようにDesignポリシーを構成します (メニューのPoliciesDHCP Settings []メニューの []をクリックしDesign、[DHCP ポリシーの編集]ページをクリックAddPoliciesします)。 Policies DHCP Settings

CLI コマンド

オーバーライドクライアント ID属性を設定するための構文はclient-class 、name set override-client-id="式"です。クライアント クラス検索 ID属性を設定するための構文は式dhcp set client-class-lookup-id="です"。use クライアント ID-for-予約属性を設定するための構文はpolicy name enable use-client-id-for-reservationsです。

予約の上書きの例

次の例は、予約のクライアント ID をオーバーライドする方法を示しています。

ステップ 1 予約のスコープを作成します。

a) サブネット アドレスを入力します。

b) 動的予約が必要な場合は、IP アドレス範囲を追加します。

ステップ 2 スコープの予約を追加します。

a) ルックアップ キーの値を含めます。

b) ルックアップ キーの種類をバイナリとして指定します。

ステップ 3 目的のポリシーを作成し、use-client-id-予約属性を有効にします。

ステップ 4 目的のクライアント クラスを作成します。

a) 前の手順で作成したポリシーを指定します。

b) パケットの内容に基づいて、目的のクライアント ID を持つ BLOB 値を返すオーバーライド クライアントID属性の式を含めます。

ステップ 5 MAC アドレスを持つクライアントのリースを取得します。このクライアントはオーバーライド ID を取得します。

IPv6 リースの再設定

DHCPv6 リースの場合、RECONFIGURE メッセージをクライアントに送信して、サーバーに新しい構成パラメータまたは更新された構成パラメータがあることをクライアントに通知できます。適切な認証によって承認された場合、クライアントはサーバーと、更新、再バインド、または情報要求の応答トランザクションを開始して、新しいデータを取得できるようにします。

DHCPv6 ポリシーの再構成を有効にする方法の詳細DHCPv6 ポリシーの設定 （198 ページ）については、を参照してください。

ローカル アドバンスド Web UI

[プレフィックスの DHCP リースの一覧/追加Reconfigure] ページには、各リースのボタンが含まれるため、その特定のリースに対して再構成要求を開始できます。

CLI コマンド

再設定をサポートするために、Cisco Primeネットワーク レジストラーlease6には、コマンドの次の構文が含まれています。

```
lease6 [vpn-name/]ipaddr reconfigure [renew | rebind | information-request] [-unicast | -via-relay]
```

オプションは、クライアントがリコンフィグレーション メッセージに対して更新パケット、再バインドパケット、または情報要求パケットで応答するかどうか、およびサーバーがユニキャストするかリレー エージェントを通過するかを決定します。およびlease6 showコマンドは、これらの関連属性の値も表示しますlease6。 list

- クライアント再構成キー- クライアントへのメッセージの再構成のためにサーバーが生成する 128 ビットキー。
- クライアント再構成キー生成時間: サーバーがクライアント再構成キーを生成した時刻。

ポリシーコマンドには、関連する 2 つの属性設定が含まれています。

- 再構成—(1)、許可しない(2)、または(3)サポートを再設定する必要があるかどうか。プリセット値は許可 (1) です。

- リレー経由で再構成— リレー エージェント上での再構成を許可するかどうか。プリセット値は false で、それによって再設定通知はサーバーからのユニキャストによって行われます。

リースを強制的に使用可能にする

現在のリースを強制的に使用可能にすることができます。ユーザーがリースを解放するか、または自分でリースを解放するように要求してから、そのユーザーの可用性を強制する必要があります。リースの可用性を強制する場合、サーバーの再ロードは必要ありません。

（注） リースが強制的に使用可能になった後、クライアントはDHCPサーバーに接続するまでリースを使用し続けます。

ローカル アドバンスド Web UI

リースの可用性を強制するには、次の手順を実行します。

ステップ 1 Design メニューで、DHCPv4 サブメニューから Scopes を選択し、[DHCPスコープの一覧/追加（List/Add DHCP Scopes）] ページを開きます。

ステップ 2 リースがあるスコープの [リース（Lease）] タブをクリックします。

ステップ 3 [DHCPスコープの編集（Edit DHCP Scope）] ページでリースの IP アドレスをクリックします。

ステップ 4 [DHCPスコープの編集（Edit DHCP Scope）] ページで Force Available をクリックします。リースは、フラグ列に空の値を表示します。

CLI コマンド

リースの可用性を強制するには、 lease vpn/]ipaddrforce-availableを使用します。scope名前 clearUnavailableを使用して、スコープ内のすべての **"利用不可"** リースを強制的に **"利用可能"** 状態に変更します。

リース更新の抑制

通常、Cisco Primeネットワーク レジストラー DHCP サーバーは、クライアントとそのリースIPアドレスとの関連付けを保持します。DHCPプロトコルは、この関連付けを明示的に推奨しており、通常は望ましい機能です。ただし、ISP などの一部の顧客では、長期間のリース関連付けを持つクライアントは、IPアドレスを定期的に変更する必要があるため、望ましくない場合があります。Cisco Prime Network レジストラーには、DHCP クライアントがリースの更新または再起動を試みたときに、リースアソシエーションを強制的に変更できるようにする機能が含まれています。

サーバーはクライアントにリースの変更を強制することはできませんが、DHCPRENEW 要求またはDHCPDISCOVER要求に基づいてクライアントに強制的に変更を強制することができます。Cisco Prime Network レジストラーには、クライアントに IP アドレスの変更を強制するために使用するインタラクションを選択できる設定オプションがあります。

- Inhibiting all —lease**クライアント**がリースされたアドレスを使用している間、リースのrenewals延長を定期的に試みます。更新を行うたびに、サーバーはリースを拒否して、クライアントが IP アドレスの使用を停止することを強制できます。クライアントは、リースが終了すると終了するアクティブな接続を持っている可能性があるため、DHCP対話のこの時点での更新の禁止はユーザーに表示される可能性があります。
- Inhibiting renewals - at DHCP クライアントが再起動すると、有効期限のない有効なリースバインディングが記録されたか、有効なrebootリースが存在しない可能性があります。リースがない場合は、サーバーが最後に保持したリースを許可しないようにすることができます。クライアントに有効なリースがある場合、サーバーはそれを拒否し、クライアントは新しいリースを取得することを余儀なくされます。いずれの場合も、アクティブな接続はリースされたアドレスを使用しないため、禁止が目に見える影響を与えません。
- —更新Effectの禁止よりも予約がon**優先**reservationsされます。クライアントに予約がある場合、更新禁止が構成されているかどうかにかかわらず、予約済み IP アドレスを引き続き使用できます。
- —更新禁止テストの後、クライアントクラスのテストが行われます。 Effect on client-classes クライアントが更新禁止によって IP アドレスの変更を強制された場合、クライアント・クラスの処理は、サーバーがクライアントに提供するアドレスに影響を与える可能性があります。

スコープまたはクライアントごとに、システム全体を設定できるポリシーのリース更新禁止を有効または無効にできます。禁止 all-re-news属性により、サーバーはすべての更新要求を拒否し、クライアントが DHCP サーバーに接続するたびに新しい IP アドレスを取得することを強制します。再起動時の更新を禁止属性は、クライアントがリースを更新することを許可しますが、サーバーは再起動するたびに新しいアドレスを取得するように強制します。これは、ディスカバーおよびINIT-REBOOT 操作にのみ適用されます。ディスカバーが含まれているのは、再起動時に INIT-REBOOT を使用する DHCP クライアントが少ないためです (ほとんどのクライアントはディスカバリを行うだけです)。

次の条件下では、更新が禁止されません。

- フェールオーバーを使用する場合、および開始状態がMCLTより短い時間から経過した時間。デフォルトの MCLT は 60 m です。
- フェールオーバーを使用する場合、フェールオーバーの状態は通常またはPARTNER-DOWNではありません。
- リースが AVAILABLE で、クライアント作成時間が更新禁止時間よりも短い時間である場合。更新禁止 -最大時間のデフォルト値は 60s です。
- リースが提供またはリースされ、要求がディスカバーまたはリクエスト選択であり、状態の開始時刻が更新禁止時間よりも短い時間である場合。renewal-inhibition-max-time のデフォルト値は、60 s です。

DHCP サーバーは、拒否する必要があるクライアント メッセージ (更新要求など) と再送信を表すメッセージを区別する必要があります。サーバーはメッセージを処理するときに、パケッ

トが到着した時刻を記録します。また、クライアントにリースバインドを行った時刻と、そのバインドに関するクライアントからのメッセージを最後に処理した時刻も記録します。次に、パケットの到着時刻をリース バインディング時間 (開始状態時間) と比較し、バインディングの開始時刻から一定の時間間隔内にクライアントからのパケットを処理します。既定では、この時間間隔は 1 分です。

ローカル アドバンスド Web UI

リースの更新を禁止するには、[DHCP ポリシーの編集] ページでポリシーをDesign作成しPolicies(メニューから[DHCP **設定**]サブメニューの下で選択し、ポリシーの名前を選択して、すべての更新を禁止するか、再起動時に更新を禁止する)を有効にします。(両方の属性は、無効にプリセットされています)。次に、ポリシーを変更し、[**保存**]をクリックして変更を保存します。

サーバー間でのリースの移動

サーバーの構成が、推奨される制限を超えるほど大きくなり、新しいDHCPサーバーにリースを移動する必要がある場合があります。リースを新しいサーバーに移動するか、既存のサーバーに移動するかによって、このタスクを実行する方法はさまざまです。これらの方法のどちらを使用する場合も、特別な考慮事項と慎重な実行が必要です。多くの場合、新しいサーバーは、構成全体と状態データベースを移動することによって最も簡単に実行できます。リースを別のサーバーに移動するには、leaseadmin ユーティリティを使用します。このユーティリティを使用すると、すべてのリースまたは選択したリースセットをエクスポートしたり、エクスポートしたリースセットをインポートしたりすることもできます。

⚠

注意 leaseadminユーティリティはローカルクラスタ (エクスポートまたはインポート) でのみ使用する必要があり、dhcpサーバーはleaseadminユーティリティを実行する前に停止する必要があります。

リースをあるサーバーから別のサーバーに移動できるようにleaseadminユーティリティがCisco Prime Network レジストラーに追加されました。このユーティリティは、DHCP サーバーと同じマシン上で実行する必要があり、データベースファイルの読み取りおよび変更を行うには、スーパーユーザー/ルート権限が必要です。このユーティリティでは、リース状態データベースに直接アクセスする必要があります。ただし、DHCPサーバーを停止しても、停止したサーバーはリース状態データベースを開いたままにしているので、十分ではありません。データベースがまだ使用中のときにユーティリティが実行されると、leaseadminユーティリティは"リース状態データベースへの排他的アクセスを取得できませんでした"というエラーを報告します。デフォルトの場所は次のとおりです。

/opt/nwreg2/local/usrbin

コマンド プロンプトで上記の場所に移動し、次の構文を使用してユーティリティを実行します。

./leaseadmin <options>

次の表では、leaseadmin ユーティリティの修飾オプションについて説明します。

表 *26 :* リース管理者コマンド オプション

オプション	説明
リースをエクスポートするには	
–e filename	ファイルにエクスポート
–x	未加工の出力形式を送信します (インポートに必要)。
–t{カレント|歴史|詳細|すべて|v6リース|v6履歴}	エクスポートするレコードの種類を指定します。有効な値は次のとおりです。 現在の 、履歴、詳細、すべて、 v6leases、 v6history
–s subnet | prefix	サブネットまたはプレフィックスにエクスポートするリース レコードを制限します。
リースをインポートするには	
–i filename	ファイルからのインポート。nオプションと共に使用する場合は、VPN を指定します。
–o	i(インポート) オプションと共に使用すると、既存のデータが上書きされます。
–c	レコードを圧縮します。
リースまたはサーバー DHCP 一意識別子 (DUID) を削除するには	
–d 住所|サブネット|プレフィックス	削除するアドレス、サブネット、またはプレフィックスを指定します。

オプション	説明
–dサーバー・デュード	サーバー DUID 情報をデータベースから削除することを指定します。 （注） サーバー・デュードを指定すると、自動生成された DHCPv6 サーバー DUID が存在する場合は、その DUID が削除されます。 リースデータベースが別フェールオーバーに関連する問題のトラブルシューティング時に避けるべき事項（109 ページ）のローカル クラスタにコピーされる場合は、推奨しませんが、コピーされたデータベースに存在する可能性のあるサーバー duid は、この操作を使用してサーバーの duid を削除することが重要です。 サーバー・デュードが削除されると、DHCP サーバーが始動すると、新しいサーバー・デュードが生成されます。これにより、クライアントが DHCPv6 再バインド要求の送信を開始するまで、古いサーバー・デュードをサーバー ID オプションとして指定したすべての DHCPv6 更新要求がドロップされます。 10.x 以降の場合、サーバーは一度削除されると、ローカル クラスターの UUID を使用し、クラスターの構成から利用できるようにリース データベースに格納されません。
[全般（General）] オプション	
–n vpn	e (エクスポート)、-i (インポート)、または -d (削除) オプションと共に使用する場合は、VPN を指定します。すべての VPN を含めるには、「すべて」を指定します。
–h path	データベースへの既定のパスをオーバーライドします。
–v	データベース バージョンを表示します。
–z{文字}=レベル	デバッグ出力レベルを設定します。

使用不可としてマークされているリースの処理

効果的なリースメンテナンスの側面の1つは、スコープ内の利用できないリースの数を決定することです。この数は予想よりも大きくなる場合があります。利用できないリースは、深刻な問題を示している可能性があります。リースが利用できない原因として、次のことが考えられます。

- 現在アクティブ***The DHCP server is configured for a ping before an offer, and the ICMP echo message is returned successfully***なクライアントがその IP アドレスを使用しているので、DHCP サーバーは使用不可能とマークします。 サーバーがアドレスを使用しないようにす

るには、クライアントにアドレスを提供する前に ping を無効にします。アドレス提供前のホストへの ping 実行 （238 ページ）を参照してください。

- —クライアントはローカル LAN セグメントの IP アドレスに対してアドレス解決(ARP)要求を行い、別のクライアントがそのアドレスに応答します。 The server receives a DHCPDECLINE message from a client to which it leased what it considered to be a good IP address クライアントは、DHCPDECLINE パケットを使用してサーバーにアドレスを返し、別の DHCPDISCOVERパケットを送信して新しいアドレスを取得します。サーバーは、クライアントから返されるアドレスを使用できないものとしてマークします。サーバーが DHCPDECLINE メッセージに応答しないようにするには、スコープ属性 (無視拒否) を設定します。

- DHCPOFFER メッセージに続くすべての DHCPREQUEST メッセージがブロードキャストされるため、サーバーは他の DHCP サーバーに送信されたメッセージを見ることができます。 The server receives “other server” requests from the client サーバーは、パケット内の server-idオプションの値によってメッセージがメッセージに送信されることを認識しています。Cisco Prime Network レジストラーサーバーが、自身の IP アドレスがserver-idオプションに表示されないという点で、別のサーバーに向けられたメッセージを認識する場合、メッセージ内のアドレスはサーバーが制御するアドレスであり、2 台のサーバーが同時にアドレスを管理しようとしていると考えています。次に、ローカルアドレスを利用不可としてマークします。この動作は、DHCP フェールオーバー構成では適用されません。2 つのサーバーが、同じ IP アドレスの一部またはすべてを使用して構成されているか、または(まれに) DHCP クライアントがパケットに誤ったserver-idオプション値を配置したかのどちらかです。

  クライアントが(実際に他のサーバーに送信されるパケットではなく)不正なサーバー IDオプションを送信していると考える理由がある場合、Cisco Prime Network レジストラーには、この動作を無効にするサーバー属性を持つことができます。

- —非常にまれで、サーバーの起動時に、サーバーがリースの設定中に、内部キャッシュのリフレッシュ中にディスクからリースデータを読み取る場合にのみ発生します。 Inconsistent lease data リース状態はリース済みとして表示されますが、リースにクライアント IDオプション値が設定されていない可能性があるなど、そのリース用のクライアントを構築するための不完全なデータが存在します。サーバーはデータに不整合があると見なし、IP アドレスを利用不可とマークします。リースを強制的に利用可能にする(CLI で lease ipaddr force-availableコマンドを使用するなど) この問題を解決する必要があります。

使用不可リースのタイムアウトの設定

で使用不可としてマークされているリースの処理 （263ページ）説明したように、リースが使用不能になった時点では、すべての利用不可能なリースは構成された時間だけその状態のままになり、その後も再び利用可能になります。ポリシー属性(利用不可タイムアウト)は、この時間を制御します。system_default_policyポリシーでは、既定でこの値を 1 日に設定します。

このタイムアウト機能を持たない Cisco Prime Network レジストラーの旧リリースからのアップグレードを処理するために、サーバー レベルで特別なアップグレードタイムアウト属性、アップグレード不可タイムアウト(1 日に事前設定されている)が含まれます。アップグレード不可

タイムアウト値は、Cisco Prime Network レジストラーのアップグレード前に使用不可能に設定されたリースに与えられるタイムアウトです。この設定は、実行中のサーバーのみに影響し、データベースの書き換えは行いません。サーバーが再ロードせずに1日稼働している場合、前回のリロード時に存在していたすべての利用不可能なリースはタイムアウトになります。サーバーが1日未満でリロードすると、次のリロードでプロセス全体が再開されます。このプロセスは、アップグレード前に使用不可能に設定されたリースに対してのみ行われます。アップグレード後に利用不可になったリースは、前述のように、ポリシーから利用不可タイムアウト値を受け取ります。

リースの照会

Cisco Prime Network レジストラーは、シスコのルータと連携して、プロビジョニング機能を強化できます。この機能は、CISCO プライムネットワーク レジストラーが準拠する DHCP リースクエリ仕様(RFC 4388)で説明されています。Cisco uBR アクセス コンセントレータ リレー エージェントの実装の一部は、DHCP リース要求および応答から情報を収集して収集することです。この情報は、次の用途に使用されます。

- 加入者ケーブル モデムとクライアント MAC アドレスをサーバーが割り当てた IP アドレスに関連付けます。
- アップストリーム データグラムの送信元 IP アドレスを確認します。
- DOCSIS ベースライン プライバシー プロトコルを通じてユニキャスト ダウンストリーム トラフィックを暗号化します。
- uBR とサブスクライバ ホストに負担をかける可能性があり、悪意のあるクライアントが侵害する可能性がある、ダウンストリーム アドレス解決プロトコル (ARP) 要求のブロードキャストを避けてください。

uBR デバイスは、グリーニングを通じてすべての DHCP 状態情報をキャプチャするわけではありません。uBR デバイスは、ユニキャスト メッセージ (特に更新およびリリース) から収集できません。また、このデータは uBR のリブートまたは置換の間で保持されません。したがって、uBR デバイスの DHCP 状態情報の唯一の信頼できるソースは、DHCP サーバー自体です。

このため、DHCP サーバーは、DHCPINFORM メッセージに似たメッセージをサポートします。アクセス コンセントレータおよびリレー エージェントは、DHCP サーバーから、DHCPv4 アドレスおよび DHCPv6 アドレスに対してクライアントロケーション データを直接取得できます。

関連項目

リースクエリの実装 （266 ページ）

DHCPv4 の 事前 RFC リースクエリ （266 ページ）

DHCPv4 の RFC 4388 リースクエリ （267 ページ）

DHCPv6 のリースクエリ （268 ページ）

リースクエリの統計 （269 ページ）

リースクエリの例 （271 ページ）

リースクエリの実装

Cisco プライムネットワーク レジストラーは、次の 3 つのリースクエリ実装を提供します。

- DHCPv4 以前の RFC 4388 用のシスコ独自仕様DHCPv4 の 事前 RFC リースクエリ （266 ページ）
- RFC 4388 に準拠する DHCPv4 - 「」を参照してください。DHCPv4 の RFC 4388 リースクエリ （267 ページ）
- DHCPv6：「DHCPv6 のリースクエリ （268 ページ）」を参照

DHCPv4 のシスコ独自の実装と最新の RFC 準拠の実装は、わずかな方法でしか異なっており、共存します。DHCP サーバーは、同じポートで Leasequery 要求を受け入れ、両方の実装に指定されたデータを返します。DHCPv6 の実装は、RFC 5007 および RFC 5460 に準拠しています。

DHCP サーバーは、DHCPv4 および DHCPv6 のリースクエリ応答にリース予約データを含めることができます。Cisco Prime Network レジストラーは、予約済み DHCPv4 のデフォルトリース時間 (31536000 秒) を返し、応答で DHCPv6 リースのリースの有効期間を返します。IP アドレスが実際にリースされている場合、Cisco Primeネットワーク レジストラーは残りのリース時間を返します。

リースクエリは、すべての実装で有効にするように事前設定されています。それを無効にするには、エキスパートモード属性を無効にします。

DHCPv4 の 事前 RFC リースクエリ

リースクエリ メッセージには、通常、要求フィールドとオプションが含まれます。例として、リレー エージェントの再起動または交換後に、リレー エージェントがパブリック ブロードバンド アクセス ネットワークにダウンストリームのデータグラムを転送する要求を受信したとします。リレーエージェントはダウンストリーム ロケーションデータを持たなくなったため、リレー エージェントのゲートウェイ IP アドレス (giaddr) と、ターゲット クライアントの MAC アドレスまたはdhcp クライアント識別子(オプション 61) を含む DHCP サーバーに LEASEQUERY メッセージを送信します。DHCP サーバーは、クライアントを検出すると、クライアントの IP アドレスを leasequery への応答のクライアント アドレス (ciaddr) フィールドに返します。サーバーがクライアント アドレスを見つけられない場合は、DHCPNACK を返します。

DHCPv4 の事前 RFC 実装では、リクエスタは IP アドレス、クライアント ID オプション (61)、または MAC アドレスを問い合わせることができ、DHCPACK (返されたデータを含む) または DHCPNACK メッセージをサーバーから受信するか、サーバーがパケットをドロップします。要求に複数のクエリタイプが含まれている場合、DHCP サーバーは最初に見つかるクエリタイプに応答します。リクエスタからのgiaddr値は、検索されたciaddrから独立しており、単にサーバーからの応答の戻り IP アドレスです。次の 3 つのクエリの種類があります。

- IPaddress ciaddr —要求パケットは ciaddr フィールドの IP アドレスを含(みます。) DHCP サーバーは、そのアドレスを使用するために、最新のクライアントのデータを返します。ciaddr値を含むパケットは、MAC アドレス フィールド (htype、 hlen、および chaddr) また

はdhcp クライアント識別子オプションの値に関わらず、IP アドレスによる要求である必要があります。IPアドレスによるクエリは最も効率的な方法であり、最も広く使用されている方法であり、他の 2 つの方法は DHCP サーバーに負荷をかける可能性があります。

- : 要求パケットにはdhcp クライアント識別子オプション値が含まれます。 dhcp-client-identifier option (61) DHCP サーバーは、最後にアクセスされたクライアントの IP アドレス データを含む DHCPACK パケットを返します。要求が MAC アドレスを省略した場合、サーバーは、要求されたクライアント ID のすべての IP アドレスとデータをcisco-leased-ipオプション(関連付けられたIPとも呼ばれます)に返します。要求にMACアドレスが含まれる場合、サーバーはDHCP クライアント識別子と MAC アドレスを IP アドレスのクライアント データと照合し、そのデータをciaddrフィールドまたはcisco-leased-ip(関連 IPとも呼ばれる)オプションに返します。
- : 要求パケットには、ハードウェア タイプ (htype) 、アドレス長 (hlen) 、およびクライアント ハードウェア アドレス (chaddr) フィールド、および空の ciaddr フィールドに MAC アドレスが含まれます。 MAC address サーバーは、応答パケットのcisco リース IP(関連付けられた IPとも呼ばれます)オプションの MAC アドレスのすべての IP アドレスと最新のリース データを返します。

RFC 前実装の DHCPメッセージ タイプオプション (53) の DHCPLEASEQUERY メッセージ番号は 13 です。この種類のメッセージをサポートしていないサーバーは、パケットをドロップする可能性があります。DHCPACK メッセージ応答には、htype、 hlen、およびchaddrフィールドのリース所有者の物理アドレスが常に含まれます。要求にciaddrが含まれている場合、返されるデータは常にciaddrに基づいており、クライアント ID または MAC アドレスはベースにしません。

リクエスターは、アドレスに関する特定のオプションを要求するパラメーター要求リスト・オプション (55) を含めることができます。応答には、dhcp-lease-timeオプション (51) と、クライアントが送信したRelay-agent-infoオプション (82) の元の内容が含まれることがよくあります。サーバーがクライアントの有効なリースを検出しない場合、サーバーはオプション 51 を返さないため、リクエスタは有効なリースがあるかどうかを判断する必要があります。

サーバーからの DHCPACK には、次のリースクエリ オプションを含めることもできます。

- **シスコリース-*ip*** (161)- クライアントに関連付けられたすべての IP アドレスのデータ。関連付けられた IPオプション (および後で名前が変更された) とも呼ばれます。
- ***cisco* クライアントが要求したホスト名** (162):ホスト名オプション(12)またはクライアントFQDNオプション(81)でクライアントが要求したホスト名。要求されたホスト名は、RFC 4388 の実装で削除されました。
- ***cisco* クライアント-最後のトランザクション時間** (163):DHCP サーバーがクライアントに接続した最新の時間。

DHCPv4 の RFC 4388 リースクエリ

リースクエリは、2006年2月にDHCPv4の公式RFC 4388になりました。Cisco プライムネットワーク レジストラーは、RFC 4388 実装を前DHCPv4 の 事前 RFC リースクエリ （266 ページ）の RFC の実装と共に提供します (を参照) と、それらの間に競合はありません。ただし、RFC 4388 の実装には、いくつかの顕著な変更が含まれています。

- DHCP メッセージ タイプオプション (53) に含まれる DHCPLEASEQUERY メッセージ タイプは、メッセージ ID を 10 に変更し (ID 13 は DHCPLEASEACTIVE メッセージに与えられました)、応答メッセージは DHCPACK および DHCPNACK からより具体的に変更されました。
  - クエリの場合は 10 です。
  - 割り当てられていないアドレスの応答に対する DHCPLEASEUNASSIGNED (11)
  - 不明なアドレスの応答に対しては、DHCPLEASEUNKNOWN 不明 (12)
  - アクティブ・アドレスの応答に対する DHCPLEASEACTIVE (13)
- 応答オプション名と ID が変更され、cisco クライアントが要求した host-nameオプションがドロップされ、応答オプションが 2 つしかないようになっていました。
  - **クライアント最終トランザクション時間** (91):DHCP サーバーがクライアントに接続した最新の時間。
  - **関連付け*-ip*** (92)—クライアントに関連付けられているすべての IP アドレスのデータ。
- クライアント ID または MAC アドレスによる照会の場合、要求にはdhcp クライアント識別子オプション (61) または MAC アドレスのみを含めることができます。パケットに両方が含まれている場合、サーバーはそれをドロップします。

DHCPv6 のリースクエリ

Cisco プライム ネットワーク レジストラーは、RFC 5007 (UDP) と RFC 5460 (TCP、バルク) DHCPv6 の両方のリースクエリ機能

（注） RFC 5460 (TCP、一括) リースクエリ サポートを使用するには、IPv6 用のDHCP リスナーの設定 （290 ページ）DHCP リスナを作成する必要があります (を参照)。

DHCPv6 リースクエリのメッセージタイプは次のとおりです。

- LEASEQUERY (14)
- LEASEQUERY_REPLY (15)
- LEASEQUERY_DATA (17)
- LEASEQUERY_DONE (16)
- 240)

クエリは次の方法で行うことができます。

- QUERY_BY_ADDRESS (1)
- QUERY_BY_CLIENTID (2)
- QUERY_BY_RELAY_ID(3)
- QUERY_BY_LINK_ADDRESS(4)
- QUERY_BY_REMOTE_ID(5)

DHCPv6 LEASEQUERY_REPLYメッセージには、以下のオプションを 1 つ以上含めることができます。

- ***lq-query*** (44)—クエリが実行されています。要求でのみ使用されるオプションには、クエリの種類、リンク アドレス (0::0)、およびクエリに必要なデータを提供するオプションが含まれます。
- **クライアント データ**(45)- 単一のリンク上の単一のクライアントのデータをカプセル化します。クライアントデータには、これらのオプションまたはその他の要求されたオプションをいくつでも含めることができます。
- ***clt-time*** (46)—クライアント データ オプションにカプセル化されたクライアントの最後のトランザクション時間 (45);は、サーバーがクライアントと最後に通信した時間 (秒単位) を示します。
- ***lq-relay-data*** (47)—クライアントが最後にサーバーと通信したときに使用されるリレー エージェント データ。フィールドはピア アドレスとリレー メッセージです。このオプションには、さらにオプションを含めることができます。
- ***lq-client-link*** (48)—クライアントがバインディングを持つリンク。リンク アドレスが省略され、クライアントが複数のリンク上にあることが判明した場合に、クライアント クエリに対する応答で使用されます。
- ***option_lq_base_time***—バインド情報を送信した時点での DHCPv6 サーバーの現在の絶対時刻を指定します。

DHCPv6 LEASEQUERY_REQUESTメッセージには、以下のオプションを 1 つ以上含めることができます。

- ***option_lq_start_time***-指定した時間以降に更新されたバインド。このオプションは、オフライン期間中に発生したバインディング更新のリストに使用されます。
- ***option_lq_end_time***-指定された期間中に更新されたバインド。

DHCPv6 は、オプション要求オプション (oro) を使用して、リースクエリ応答のオプションのリストを要求します。

（注） クライアント ID による leasequery 要求では、[v6-override-client-id] 式を使用してクライアントのリースに関する情報を正しく取得する場合に、オーバーライドクライアントID属性を指定する必要があります。

リースクエリの統計

リース クエリは、Web UI の[DHCP サーバーの統計情報] ページ (の「統計の表示」Cisco プライムネットワーク レジストラー 11.0 管理ガイドセクションを参照)、および CLI でdhcp getStats 統計属性を提供します。リースクエリの統計は次のとおりです。

- lease-queries:指定された時間間隔で受信した RFC 4388 メッセージ ID 10 (または RFC 以前のメッセージ ID 13) DHCPv4 リースクエリ パケットの数。
- lease-queries-active: RFC 4388 DHCPLEASEACTIVE パケットの数。

- lease-queries-unassigned: RFC 4388 DHCPLEASEUN 割り当てパケットの数。
- lease-queries-unknown: RFC 4388 DHCPLEASEUNKNOWN パケットの数。
- leasequeries-受信した DHCPv6 リースクエリ パケットの数。
- leasequery-replies- 成功する場合と成功しない場合がある DHCPv6 リースクエリ パケットに対する応答の数。
- tcp-current-connections- DHCPv6 アクティブクエリおよびバルク リースクエリの DHCP サーバーへの現在開いている TCP 接続の数。
- tcp-total-connections- この時間間隔で DHCPv6 アクティブクエリおよびバルク リースクエリの DHCP サーバーに対して開かれた TCP 接続の数。
- bulk-leasequeries- この時間bulk-leasequeries間隔ですべての TCP 接続で受信した LEASEQUERY パケットの数。
- bulk-leasequery-replies- この時間間隔ですべての TCP 接続を介して送信された LEASEQUERY-REPLY パケットの数。
- bulk-leasequery-data- この時間間隔ですべての TCP 接続を介して送信された LEASEQUERY-DATA パケットの数。
- bulk-leasequery-done- この時間間隔ですべての TCP 接続を介して送信された LEASEQUERY-DONE パケットの数。
- tcp-lq-status-unspec-fail- この時間間隔で TCP を介して送信されるステータス コード UnspecFail(1) を持つ LEASEQUERY-REPLY パケットの数。
- tcp-lq-status-unknown-query: この時間間隔で TCP を介して送信される状態コードが不明なリースクエリ-応答パケットの数です。
- tcp-lq-status-malformed-query- この時間間隔で TCP を介して送信された、状態コードが異常である LEASEQUERY-REPLY パケットの数です。
- tcp-lq-status-not-configured- この時間間隔で TCP を介して送信される状態コードが未構成(9) の LEASEQUERY-REPLY パケットの数。
- tcp-lq-status-not-allowed-この時間間隔で TCP 経由で送信されるステータス コードが NotAllowed(10) の LEASEQUERY-REPLY パケットの数。
- tcp-lq-status-query-terminated: この時間間隔で TCP を介して送信された状態コードが[11]であるリースクエリ-応答/リースクエリ-DONEパケットの数。
- tcp-connections-dropped- DHCPv6 リクエスターによって TCP 接続がクローズ (またはリセット) されたために、この時間間隔で終了した TCP 要求の数。これは、通常の接続のクローズまたはサーバーの再ロードを除外します。
- **アクティブ リースクエリ**—この時間間隔内にすべての TCP 接続を介して受信される ACTIVELEASEQUERY パケットの数。
- **アクティブリースクエリ応答**-アクティブなリースクエリのこの時間間隔内にすべての TCP 接続を介して送信される LEASEQUERY-REPLY パケットの数。
- **アクティブ リース クエリ データ**-アクティブなリース クエリに対して、この時間間隔内にすべての TCP 接続を介して送信される LEASEQUERY-DATA パケットの数。
- **アクティブリースクエリ完了**-アクティブなリースクエリに対して、すべての TCP 接続を介して送信される LEASEQUERY-DONE パケットの数。
- tcp-lq **状況データ欠落**-この時間間隔で TCP を介して送信される状態コード DataMissing(240) を持つ LEASEQUERY-REPLY パケットの数。

- tcp-lq **状況キャッチアップ-完了**- この時間間隔で TCP を介して送信される状態コードが CatchUpComplete(241) の LEASEQUERY-DATA パケットの数。

リースクエリの例

次の例は、リンク アドレスがないクライアント ID による DHCPv6 UDP クエリのパケット トレースを示していますが、複数のリンクにアドレスが含まれています。出力の最初の部分はクエリ メッセージを示し、2 番目の部分は応答データを示します。lq-queryオプションは、照会のタイプを識別します。要求のオプション要求オプション(oro)を使用して要求されたオプションのリストと、応答のlq-client-linksオプションで返される 2 つのアドレスを確認します。

例: UDP **リース クエリのパケット トレース**

```
+- Start of LEASEQUERY (14) message (113 bytes)
| transaction-id 22
| lq-query (44) option (37 bytes)
| (query-type 2, link-address ::)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| oro (6) option (2 bytes)
| 47
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| vendor-class (16) option (14 bytes)
| (enterprise-id 1760,
| ((00:08:41:49:43:20:45:63:68:6f)))
| vendor-class (16) option (14 bytes)
| (enterprise-id 1760,
| ((00:08:41:49:43:20:45:63:68:6f)))
+- End of LEASEQUERY message
+- Start of LEASEQUERY-REPLY (15) message (72 bytes)
| transaction-id 22
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| lq-client-links (48) option (32 bytes)
| 2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e,2001:4f8:ff00:0:e400:f92:1bfd:60fa
+- End of LEASEQUERY-REPLY message
```

次の例は、クライアント ID による DHCPv6 TCP クエリのパケット トレースを示しています。出力の最初の部分は要求メッセージを示し、2 番目の部分は最初のクライアントのバインディング データを含む応答メッセージを示し、最後の部分はクエリが正常に終了したことを示します。返されるクライアントが複数ある場合、3 番目の部分は 2 番目の部分に続きます。

（注） リースクエリ-応答メッセージにバインディング データがない場合、パケットには LEASEQUERY-DONE メッセージは存在しません。

例: TCP **リース クエリの例のパケット トレース**

```
+- Start of LEASEQUERY (14) message (59 bytes)
| transaction-id 2
| lq-query (44) option (37 bytes)
| (query-type 2, link-address ::)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| oro (6) option (2 bytes)
| 47
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
+- End of LEASEQUERY message

+- Start of LEASEQUERY-REPLY (15) message (162 bytes)
| transaction-id 2
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| client-data (45) option (122 bytes)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| clt-time (46) option (4 bytes)
| 5m54s
| iaaddr (5) option (24 bytes)
| (address 2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e,
| preferred-lifetime 6d23h54m6s,
| valid-lifetime 1w6d23h54m6s)
| lq-relay-data (47) option (68 bytes)
| peer-address fcc0:a803::214:4fff:fec1:226a
| +- Start of RELAY-FORW (12) message (52 bytes)
| | hop-count 0,
| | link-address 2001:4f8:ffff::,
| | peer-address fe80::302:3ff:fe04:506
| | vendor-class (16) option (14 bytes)
| | (enterprise-id 1760,
| | ((00:08:41:49:43:20:45:63:68:6f)))
| +- End of RELAY-FORW message
+- End of LEASEQUERY-REPLY message
+- Start of LEASEQUERY-DATA (17) message (130 bytes)
| transaction-id 2
| client-data (45) option (122 bytes)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| clt-time (46) option (4 bytes)
| 5m33s
| iaaddr (5) option (24 bytes)
| (address 2001:4f8:ff00:0:e400:f92:1bfd:60fa,
| preferred-lifetime 6d23h54m27s,
| valid-lifetime 1w6d23h54m27s)
| lq-relay-data (47) option (68 bytes)
| peer-address fcc0:a803::214:4fff:fec1:226a
| +- Start of RELAY-FORW (12) message (52 bytes)
| | hop-count 0,
| | link-address 2001:4f8:ff00::,
| | peer-address fe80::302:3ff:fe04:506
| | vendor-class (16) option (14 bytes)
| | (enterprise-id 1760,
| | ((00:08:41:49:43:20:45:63:68:6f)))
| +- End of RELAY-FORW message
+- End of LEASEQUERY-DATA message

+- Start of LEASEQUERY-DONE (16) message (4 bytes)
```

```
| transaction-id 2
+- End of LEASEQUERY-DONE message
```

TCP バルク リースクエリと UDP リースクエリの違い

TCP バルク リースクエリと UDP リースクエリの違いは次のとおりです。

- UDP リースクエリは、IPv6 アドレスによるクエリとクライアント識別子によるクエリをサポートしています。ただし、TCP 一括リースクエリは 5 つのクエリタイプをすべてサポートします。つまり、IPv6アドレスによるクエリ、クライアント識別子によるクエリ、リレー識別子によるクエリ、リンク アドレスによるクエリ、およびリモート ID によるクエリです。
- UDP Leasequery では、サーバーが複数のリンク上のリレー エージェントのバインディングを検出した場合、DHCP サーバーは応答メッセージにOPTION_CLIENT_LINKオプションを送信します。リレー エージェントは、返された各リンク アドレスを使用して LEASEQUERY メッセージを再送信し、すべてのクライアントのバインディングを取得する必要があります。TCP一括リースクエリでは、サーバーは異なるリンク上のクライアントの複数のバインディングを返します。ただし、OPTION_CLIENT_LINKは、一括リースクエリの応答ではサポートされていません。

アドレス レポートとリース レポートの実行

IP アドレスとリースに関する次のレポートを実行できます。

- **アドレスの使用法**-「」を参照してください。アドレス使用状況レポートの実行 （273 ページ）
- **リース履歴**-参照IP リース履歴の実行 （274 ページ）
- **現在の使用状況**：「リース使用率レポートの実行 （281 ページ）」を参照
- **リース通知**— 「」を参照してください。リース通知の受信 （281 ページ）

アドレス使用状況レポートの実行

アドレス使用状況レポートには、リースが割り当てられている IP アドレスが表示されます。

ローカル アドバンスド Web UI

IP アドレスのリースを表示するには、[DHCP スコープの編集] ページDesign(メニューの Scopes[DHCPv4]サブメニューの下で選択) をクリックし、スコープの[DHCPリースの一覧] タブを開きます。特定のリースを管理するには、ページで該当する IP アドレスをクリックします。

CLI コマンド

指定したサーバーの IP アドレスの使用状況を表示reportするには、 を使用します。

ヒント　まだ自動化された方法で使用lease-notificationしていない場合は、サーバーのlease-notification available=100%状態のスコープごとの簡潔な概要を試してください。

IP リース履歴の実行

特定のデータベースから IP リース履歴データを抽出して、特定の IP アドレスの過去の割り当て情報を確認することができます。クライアントがリースを発行した時間、クライアントまたはサーバーがリースの期限切れ前にリリースした時間、およびサーバーがリースを更新したかどうか、およびどのくらいの期間をクライアントがリースを発行したかの履歴ビューを取得できます。

Cisco プライムネットワーク レジストラーは、IP 履歴データのクエリを制御するクライアントを提供します。このクライアントを使用すると、次のことができます。

- 特定の時間の間に特定の IP アドレスに関連付けられた MAC アドレスを取得します。
- IP 履歴データベース全体をカンマ区切りファイルとして参照してください。
- リース履歴の属性(リース履歴の詳細レポート)を表示するIP リース履歴の照会 （275 ページ）- を参照してください。

レコードの IP 履歴データベースをトリミングするために、データベースのサイズが限界なく拡大しないようにするには、追加の管理機能を使用する必要があります。

（注）　既存のリースの状態が変更された場合(予約済み IP アドレスとして構成されている場合や、非アクティブ化された場合など)、その変更は地域でのリース履歴の変更として表示されません。詳細コレクションが無効になっている場合、リース履歴の変更は、リースがリース済みからリースされていない状態に遷移するか、別のクライアントに割り当てられている場合にのみ表示されます。

ローカル クラスタでのリース履歴録音の有効化

ローカル クラスタ DHCP サーバーのリース履歴記録を明示的に有効にする必要があります。DHCP サーバーは、IP 履歴記録エラーを通常の DHCP ログ ファイルに記録します。

ローカルクラスタでリース履歴が有効になっている場合、サーバーのパフォーマンスとリース状態データベースのサイズに影響します。リースが終了(有効期限が切れたり解放されたり)するたびに、リース用の履歴レコードが作成されます。クライアントが長期間にわたって更新するリースでは、履歴レコードは作成されません。各リース履歴レコードのサイズは多くの要因に依存しますが、1 レコードあたり約 1 KB の見積もりが適しています。リースが終了するレートとリース履歴が保持される期間によっては、リース履歴レコードの数が多く作成され、かなりのディスク領域が必要になる可能性があります。これは、アクティブなリースに必要なスペースよりも多くの注文が大きくなる可能性があります。

ローカル アドバンスド Web UI

リース履歴の記録を有効にするには、次の手順を実行します。

ステップ 1 メニューからDeployDHCPServer[DHCP]サブメニューの下で選択し、[DHCPサーバーの管理]ページを開きます。

ステップ 2 [DHCPLocal DHCP Serverサーバー] ペインで をクリックします。

ステップ 3 [ローカル DHCP サーバーの編集] ページで、リース履歴属性を探します。

- Lease History (ip-history)：v4 のみ（DHCPv4）、v6 のみ（DHCPv6）、またはその両方のリース履歴データベースを有効または無効にします。
- ip-history-max-age：収集するリース履歴の最大経過期間。リース履歴が v4 のみに設定されている場合、v6 のみ、または両方の DHCP サーバーが定期的にリース履歴レコードを調べ、この経過時間のしきい値より古いリース履歴バインドを持つレコードを削除します。

ステップ 4 Save をクリックします。

ステップ 5 サーバーをリロードします。

CLI コマンド

リース履歴の記録を有効にするには、 を使用dhcp set ip-history=<value> (v4-only, v6-only, both, or disable)して IP アドレスの IP (リース) 履歴の記録を明示的に有効にする必要があります。

IP リース履歴の照会

リースを取得したら、その履歴を照会できます。IP リース履歴は、ローカル または地域のクラスターから照会できます。DHCP サーバーを含むローカルクラスターを地域クラスターの一部としてセットアップし、地域クラスターからのリース履歴データのポーリングを有効にします (の「リース履歴収集の有効化」セクションCisco プライムネットワーク レジストラー 11.0 管理ガイドを参照)。

地域クラスター Web UI のクラスターのポーリング基準は、「 」の「ポーリング使用率およびリース履歴データ」セクションで説明されている属性を使用して調整できます。 Cisco プライムネットワーク レジストラー 11.0 管理ガイド

また、リース履歴データのクエリの選択基準も、以下のセクションで説明します。

ローカルおよびリージョンの高度な Web UI

IPv4 リース履歴を照会するには、次の手順を実行します。

ステップ 1 メニューからOperate[**レポート**]DHCPv4 Lease Historyサブメニューの下で[DHCPリース履歴検索]ページを開きます。

（注） ローカルの詳細 Web UI の [検索（Search ）] ボタンを使用して、[DHCP リース検索（DHCP Lease Search）] ページに移動できます。このボタンを使って、リース履歴の検索ページとアクティブリースの検索ページを切り替えられます。

ステップ 2 ドロップダウン リストから [フィルター] 属性と [タイプ] を選択し、[値] フィールドで選択したフィルタータイプの値を入力します。

ステップ 3 Search をクリックし、リースの一覧を表示します。

ローカルおよびリージョンの高度な Web UI

IPv6 リース履歴を照会するには、次の手順を実行します。

ステップ 1 メニューからOperate[**レポート**]DHCPv6 Lease Historyサブメニューの下で[DHCP v6 リース履歴検索]ページを開きます。

（注） ローカルの詳細 Web UI の [検索（Search ）] ボタンを使用して、[DHCP リース検索（DHCP v6 Lease Search）]ページに移動できます。このボタンを使って、リース履歴の検索ページとアクティブリースの検索ページを切り替えられます。

ステップ 2 ドロップダウン リストから [フィルター] 属性と [タイプ] を選択し、[値] フィールドで選択したフィルタータイプの値を入力します。

ステップ 3 Search をクリックし、リースの一覧を表示します。

ローカルおよびリージョンの高度な Web UI

（注） 地域サーバーは、最新のポーリングと同じ最新のリース履歴のバージョンのみを検索します。最新のデータの場合、最新のリース履歴データを取得するために、地域の明示的なリース履歴ポーリングを実行する必要があります。

iphist ユーティリティの使用

ユーティリティを使用して、ローカルおよび地域クラスタの IP 履歴データベースを照会し、結果を標準出力またはファイルにiphist送ることができます。デフォルトの場所は次のとおりです。

/opt/nwreg2/local/usrbin

コマンド プロンプトで上記の場所に移動し、次の構文を使用してユーティリティを実行します。

iphist[オプション]{イパドル|all} [開始日 |start [終了日|end]]

IPアドレスは単一のアドレスまたはキーワードallであり、開始日は現地時間またはデータベースの最も早startい日付のキーワードで、終了日はデータベースの最後の日付のローカル時刻ま

たはキーワードendです。ただし、ローカル時間を指定する–lオプションを使用しない限り、出力は既定でグリニッジ標準時 (GMT) に設定されます。

コマンド オプションの完全な一覧が下の表に表示されます。

表 *27 : iphist* コマンド オプション

オプション	説明
–N username	管理者ユーザー名。省略すると、ユーザー名の入力を求められます。
–P password	管理者パスワード。省略した場合は、パスワードを入力するように求められます。
–C cluster [:port]	宛先サーバーとオプションの SCP ポート。
–6	出力 DHCPv6 リース
–a	リース属性の可視性 3 を表示します。
–f 形式	出力行の形式。デフォルトの形式は次のとおりです。"address,client-mac-addr,binding-start-time,binding-end-time"
–t	タイトル行として印刷形式を指定します。
–n namespace	アドレスの名前空間を指定します。
–o file	出力をファイルに送信します。
–l	デフォルトの UTC/GMT ではなく、現地時間で出力を表示します。
–i	指定した IPv6 アドレスを含むデリゲートされたプレフィックスの出力–を表示します (6 のみ)。
–s{自己|パートナー}	リースを自己またはパートナーに制限します。
–v	出力バージョンを表示します。
–zデバッグ引数	デバッグ出力レベルを設定します。

日付では次の構文を使用できます (スペース文字を含める場合は引用符が必要です)。

- 月/日/年@時間:分:秒(例えば、8/28/2007@10:01:15)、時間オプション
- 月/日/年時:分:秒(例えば、"8/28/2007 10:01:15")、時間オプション
- 月の日の時間:最小:秒年(例えば、8月 28 10:01:15 2007)、秒オプションで
- キーワードstart、 end、 now 、または (現在の時刻の場合)

日付フィルターは、その間にアクティブだったリースに出力を制限することを目的としています。つまり、開始日より前に終了しない限り、指定した開始日より前に開始できます。また、指定した終了日以降は開始できません。たとえば、次のコマンドを呼び出します。

```
# ./iphist -N user -P password all "Aug 28 00:00 2008" "Dec 31 23:59:59 2008"
```

次のリースの場合。

リース 1	Begin	2008 年 1 月 1 日	終了（End）	2008 年 6 月 30 日
リース 2	Begin	2008 年 3 月 10 日	終了（End）	2008年9月01日
リース 3	Begin	2008年6月01日	終了（End）	2008年9月30日
リース 4	Begin	2009 年 1 月 1 日	終了（End）	2009 年 3 月 10 日

リース 2 とリース 3 は、どちらもクエリの指定された開始日の後に終了するため、リース 2 とリース 3 のみを返します。他の 2 つは、指定された開始日より前に終了するか、クエリの指定された終了日より後に開始されるため、範囲外です。

各行の値は、DHCP サーバーが格納する特定のリース オブジェクトによって異なります。format コマンドを使用して、含めるiphist–f値を指定できます。

format引数は、出力行のテンプレートを提供する名前をコンマで区切った引用符で囲まれたリース属性名のリストです。デフォルトの出力はipaddress、クライアント-mac-addr、バインディング開始時、バインディング終了時です。

次に例を示します。

```
# ./iphist -f "address,client-mac-addr,binding-start-time,binding-end-time" all
```

出力は、オペレーティングシステムに適した改行シーケンスで終了する行のシーケンスです（UNIX では \n）。各行には、単一のリース レコードにデータが含まれます。行の形式は、通常、引用符で囲まれたコンマ区切り値です。引用符の内側にリテラルの円記号 (\) または引用符 (") を使用するには、前に 1 つのバックスラッシュ (\) を付けます。属性の新しい行は\nとして印刷されます。

次の表は、出力に含めることができる一般的なリース オブジェクト属性の一部を示しています。また、コマンドのヘルプもlease参照してください。完全なリストを取得するには、iphist -aを使用します。

表 *28 : IP* 履歴クエリの出力属性

リース属性	説明
address	リースの IP アドレス。
binding-start-time	リース バインドの開始時刻。
binding-end-time	リース バインドの終了時刻。
client-binary-client-id	クライアントの MAC アドレスのバイナリ形式。
client-dns-name	DHCP サーバーによって認識されるクライアントの最新の DNS 名。

リース属性	説明
client-domain-name	クライアントが存在するドメイン。
client-flags	クライアント フラグの数。
client-host-name	クライアントが要求したホスト名。
client-id	クライアントが要求したクライアント ID またはクライアント用に合成されたクライアント ID。
client-last-transaction-time	クライアントがサーバーに最後に接続した日時です。
client-mac-addr	クライアントが DHCP サーバーに提示した MAC アドレス。
client-os-type	リースされたクライアントのオペレーティングシステム。
expiration	リースが期限切れになった日付と時刻。
フラグ（Flags）	予約済みまたは非アクティブ化。
lease-renewal-time	クライアントがリース更新を発行する予定の時間を最小限に抑えます。
lease-rebinding-time	クライアントが再バインド要求を発行する予定の最小時間。
relay-agent-circuit-id	回線 IDサブオプション (1) の内容。
relay-agent-option	最新のクライアント対話からのオプションの内容。
relay-agent-remote-id	リモート IDサブオプションの内容 (2)。
relay-agent-server-id-override	サーバー IDオーバーライド・サブオプションの IP アドレス。
relay-agent-subnet-selection	サブネット選択サブオプションの IP アドレス。
relay-agent-vpn-id	vpn-idサブオプションの内容。
start-time-of-state	リースの状態が変更された日時です。
state	使用可能な、期限切れ、リース、提供、または使用不可のいずれか。
vendor-class-id	クライアントが要求したベンダー クラス ID。
vpn-id	VPN の識別子 (存在する場合)。

リース履歴データのトリミング

リージョン クラスターで IP 履歴トリミングを有効にした場合、IP 履歴データベースは自動的にトリミングされ、ディスク領域を再利用できます。各履歴レコードには有効期限があります。DHCP サーバー自体、および履歴データの DHCP サーバーをポーリングする CCM 地域サーバーには、トリミングが必要です。

CCM サーバーは、一定の期間を経過したリース履歴データを一定の間隔でトリミングする、地域クラスターでバックグラウンドトリミングを実行します。トリミング間隔はデフォルトで 24 時間に設定され、年齢 (トリミング前にどのくらいさかのぼるか) は 24 週に設定されます。ローカル クラスタの DHCP サーバーは、毎日自動トリミングを実行し(現地時間の午前 3 時)、デフォルトで 4 週間のデータを格納します。

リージョン Web UI

リース履歴データをトリミングするには、中央の構成管理者である必要があります。

ステップ 1 Operate メニューの [サーバー（Servers）] サブメニューの下から Manage Servers を選択し、[サーバーの管理（Manage Server）] ページを開きます。

ステップ 2 [サーバーの管理（Manage Server）] ペインで Local CCM Server をクリックします。

ステップ 3 [ローカルCCMサーバーの編集（Edit Local CCM Server）] ページの [リース履歴の設定（Lease History Settings）] で、次の属性を設定します（入力した値を持つ s、m、h、d、w、m、または y サフィックスを使用できます）。

- lease-hist-trim-interval：古いリース履歴データを自動的にトリミングする頻度（デフォルトは毎日）。0 に設定すると、自動的にリースがトリミングされません。境界値は 0 ～ 1 年です。
- lease-hist-trim-age：lease-hist-trim-interval が 0 に設定されていない場合に古いリース履歴データを自動的にトリミングするのに遡る期間（デフォルトは 24 週間）。境界値は 1 日から 1 年です。

ステップ 4 即時トリミングを強制するには、ページの下部にある [トリム/コンパクト入力（Trim/Compact Inputs）] セクション（圧縮はDHCP使用率データでのみ使用可能）を見つけます。トリム/コンパクト年齢を希望の値に設定します。この期間は、リース履歴データをトリミングするのにどのくらいの時間が経過します。この値に対する境界はありません。ただし、非常に小さい値(1m など)を設定すると、最新のデータをトリミングまたは圧縮しますが、これは望ましくない場合があります。実際、ゼロに設定すると、収集されたデータがすべて失われます。値を大きくし過ぎる (10y など) に設定すると、データのトリミングや圧縮が行えなくなる可能性があります。

ステップ 5 すぐにトリミングする場合は、 をTrim All Lease Historyクリックします。

IP-history-max-age属性を設定することで、DHCP サーバー自体が実行するトリミングを調整できます。ip-historyが設定されている場合、DHCP サーバーは、リース バインディングの変更に応じて、時間の経過と同時にデータベース レコードを蓄積します。このパラメーターは、データベースに保持される履歴レコードの経過時間の制限を設定します。サーバーは定期的にリース履歴レコードを調べ、このパラメーター

に基づいて経過時間のしきい値を設定し、しきい値より前に終了したバインディングを表すレコードを削除します。プリセット値は 4 週間です。

リース使用率レポートの実行

リース使用率レポートには、アドレスブロック、サブネット、およびスコープの現在の使用率が表示されます。両方のユーザーインターフェイスについては使用率履歴レポートの生成 （132 ページ）、「」を参照してください。

ローカル アドバンスド Web UI

アドレス・スペース機能のページから、アドレス・ブロック、サブネット、およびスコープの現在の使用率を表示します。

CLI コマンド

リース使用率レポートを表示するには、reportを使用します。

リース通知の受信

CLI は、使用可能な IP アドレスの数が特定のしきい値以下の場合に通知を送信する機能を提供します。このlease-notificationコマンドは、使用可能なリースの数が特定のしきい値に達した場合または下回った場合に通知が発生するタイミングを、使用可能な属性を使用して指定します。レポートをユーザーに電子メールで送信できます。対話的にコマンドを使用できますが、主に UNIXcron タスクなどの自動化された手順で使用します。

次の例では、リース通知を examplescope の空きアドレスが 10% に落ちたときの設定を行います。特定の Windows メール ホストで、受信者のビリー、ジョー、および Jane にレポートを送信します。

```
nrcmd> lease-notification available=10% scopes=examplescope recipients=billy,joe,jane
mail-host=mailhost
```

出力は、説明ヘッダー、空きアドレスの数がしきい値以下の各スコープの行を含むテーブル、および要求されたスコープとクラスターに関連する可能性のある警告で構成されます。

Cisco プライムネットワーク レジストラーでは、特に指定しない限り、デフォルトでデフォルト クラスタと .nrconfig ファイルが使用されます。コマンドの構文については、コマンドのヘルプをlease-notification参照してください。

関連項目

リース通知を自動的に実行する （282 ページ）

リース通知用の設定ファイルの指定 （282 ページ）

リース通知を自動的に実行する

cron(1)コマンドを実行するコマンドをcrontab(1)に指定することで、**定期的にリース通知**を実行することができます。

crontabに指定したこの例では、月曜日から金曜日までの 00:15 および 12:15 (午前 0 時と正午の 15 分後) に**リース通知**を実行します (これは単一のコマンド ラインを含みます)。

```
15 0,12 * * 1-5 . .profile; /opt/nwreg2/local/usrbin/nrcmd lease-notification
available=10\% config=/home/jsmith/.nrconfig addresses=192.32.1.0-192.32.128.0
recipients=jsmith,jdoe@example.com >/dev/null 2>&1
```

UNIX のcrontab -eコマンドを実行して、**クロンタブ**編集を実行できます。ed(1)を使用する場合を除き、コマンドを実行する前にEDITOR環境変数を設定します。詳細については、crontab(1)のマニュアルページを参照してください。

crontabコマンド行でCLIコマンドの絶対パスを指定する必要があることに注意してください。どのnrcmdコマンドを使用して、ご使用の環境の完全なパスを判別できます。

また、crontabを使用してリース通知**コマンド**を実行すると、nrcmdコマンドは、CNR_CLUSTER、CNR_NAME、およびCNR_PASSWORDのユーザー環境変数を無視します。他のビューアは実行中のコマンドを表示できるため、セキュリティ上の理由から、コマンドラインの-Pオプションを使用してパスワードを指定しないでください。

crontab -eを実行しているユーザーのホーム ディレクトリ内の .profile またはその他のファイルからnrcmdコマンドを実行するクラスターのクラスター名、ユーザー、およびパスワードの情報を指定します。次に例を示します。

```
CNR_CLUSTER=host1
export CNR_CLUSTER
CMR_NAME=admin1
export CNR_NAME
CNR_PASSWORD=passwd1
export CNR_PASSWORD
```

。crontabエントリの .profile 指定は、ファイルを明示的に読み取ります。最初のドット (.) は、ファイルを読み取るシェルコマンドで、少なくとも 1 つのスペース文字を使用してそれに従う必要があります。nrcmdが実行されている場所とは異なるクラスター (またはクラスター) で通知する場合は、次の情報を指定します。

- クラスタを使用して構成ファイルをチェックインしますリース通知用の設定ファイルの指定 （282 ページ）(を参照)。
- このセクションの冒頭にあるサンプル**の** crontab項目のように完全に指定されたパス。

chmod go-rwx config-file UNIX コマンドを使用してアクセス権を変更することにより、他のユーザーが作成した .profile および構成ファイルの内容を調べたり変更したりできないようにすることができます。

リース通知用の設定ファイルの指定

構成ファイルを省略する場合はlease-notification、現在のディレクトリ、ホームディレクトリ、最後に /var/nwreg2/{local | regional}/conf ディレクトリで既定の .nrconfig ファイルを探します。

Cisco プライムネットワーク レジストラーは、最初に検出されたファイルを使用します。ファイルの各行は、文字 # (コメント)、角かっこで囲まれたセクション ヘッダー、またはパラメーターと値のペアまたはその継続で始まる必要があります。Cisco プライムネットワーク レジストラーは、各行から先頭のスペース文字を取り除き、空白行を無視します。

動的リース通知

DHCPv4 および DHCPv6 動的リース通知機能により、外部クライアント アプリケーションは DHCP サーバーの IP アドレス バインディング アクティビティに関する更新を受信できます。この機能を使用すると、特定のリースアクティビティが発生したときに、リースアクティビティを使用して外部データベースを更新したり、合法的傍受などのアクションをトリガしたりできます。

（注） 動的リース通知は、現在のリース状態情報のみを提供します。すべてのリース状態の変更が報告されることを保証するものではありません。DHCP サーバーへの接続がダウンまたは輻輳状態の場合など、特定の条件下でリース状態の変更が失われます。

動的リース通知機能は、追加機能をサポートするために DHCP サーバーを拡張し、サンプルクライアント (Java で書かれている) を含み、リース状態情報を MySQL データベースに格納して機能を示します。

動的リース通知の使用

動的リース通知を使用するには:

1. ローカル クラスターに dhcp リスナ オブジェクトを作成する必要があります。dhcp リスナオブジェクトは、サーバーが着信 TCP 接続をリッスンするポートと、これらの接続のその他DHCP リスナーの設定 （290 ページ）の属性を指定します (を参照)。DHCP リスナー オブジェクトを作成した後、DHCP サーバーを再ロードする必要があります。
2. 動的リース通知クライアントは、DHCP サーバーとの TCP 接続を確立し、次のいずれかの要求を行う必要があります。
   - 一括リースクエリ- この要求は、特定の時点以降に状態が変化した DHCP サーバー内のすべてのリースの現在の状態を取得するために行われます。時間が指定されていない (または、時刻にゼロが指定されている) とき、すべてのリースの現在の状態が送信されます。これは、DHCP サーバーが 1 つの要求に応答してクライアントにすべてのリースを配信する点が異なる点を除いて、UDP ベースの DHCPv4 リースクエリ (RFC 4388) と DHCPv6 リースクエリ (RFC 5460) に似ています。通常、バルク リースクエリは、外部データベースを初期化するために使用されます。また、アクティブなリースクエリが何らかの中断を起こした後、そのデータベースを最新の状態にする場合にも使用されます。
   - アクティブリースクエリ: この要求は、DHCP サーバーが行うすべての今後の重要なリース変更に対するリース状態情報を取得するために行われます。DHCP サーバーが

重要なリース状態情報をデータベースに書き込む場合、リース状態情報はTCP接続を介して送信されます。

- アクティブ リース クエリ (キャッチアップ付き) - この要求は、将来のリース状態の変更と、最近変更されたリースの最新データを取得するために行われます。動的リース通知クライアントは、動的リース通知クライアントやDHCP サーバーの再起動時など、接続損失の短い期間に失われた最近変更されたリースの最新データを取得できます。キャッチアップを伴うアクティブなリースクエリは、リースの現在の状態のみをフェッチします。これは、見逃した可能性のあるすべての中間リース状態変更に関するデータをフェッチしません。

DHCPサーバーは、リースクエリ メッセージのストリームで、リース状態情報を動的リース通知クライアントに送信します。バルク リースクエリの場合、DHCPサーバーが処理する時間が与えるとすぐにリース状態情報が送信されます。アクティブなリースクエリの場合、リース状態の変更が発生すると、リース状態情報が送信されます。動的リース通知クライアントは、これらのメッセージを処理して、データベースの更新などの適切なアクションを実行できます。

（注） DHCPサーバーは複数の動的リース通知クライアントをサポートしていますが、複数のクライアントがDHCPサーバーのリースパフォーマンスに影響を与える可能性があるので、クライアント数を最小限に抑えることをお勧めします。

フェールオーバー構成では、DHCPクライアントと対話するアクティブ フェールオーバー パートナーのみが、動的リース通知クライアントに対して、アクティブな leasequery 要求を使用して動的リース通知の更新を送信します。したがって、完全な情報を受信するには、動的リース通知クライアントが両方のフェールオーバー パートナーに接続する必要があります。

サーバーは、dhcp リスナのleasequery-send-all属性に基づいて、アクティブなリースクエリ通知のキューにリースが登録されているかどうかを判断します。この属性が有効になっている場合、DHCPサーバーは常にアクティブなリースクエリ クライアントに通知を送信します。この属性が無効または未設定の場合、DHCP サーバーは、アクティブな leasequery クライアントで正確な状態を維持するために必要な通知のみを送信します。

また、エクステンションを使用してリースクエリ通知を制御することもできます。拡張機能は、アクティブリースクエリ制御要求および応答データディクショナリ項目を使用して、アクティブなleasequery通知用にリースがキューに入拡張ポイントの使用 （433 ページ）れられていないかどうかを決定できます。

リース通知クライアントの例

Cisco プライム ネットワーク レジストラーは、スタンドアロンのサンプル Java クライアントを提供します。スタンドアロンのサンプル Java クライアントは、1 つ以上の DHCP サーバーからリース状態データを収集し、最新のリース データで SQL データベースを更新します。サンプルの Java クライアントは、両方のフェールオーバー パートナーからのリース状態の更新を受け入れ、最新のリース状態情報が SQL データベースに含まれることを確認するように設計されています(更新が正しい順序で受信された場合でも)。サンプルJavaクライアントを使用す

る場合、バルクおよびアクティブなリースクエリ プロトコルの詳細を知る必要はありません。サンプル Java クライアントソースが提供されています。したがって、サンプルの Java クライアントがニーズを満たさない場合は、独自の実装ではなく、変更することをお勧めします。

サンプル Java クライアントは、すべてのリースの状態を取得するために初めてサーバーに接続するときに、バルク・リース照会を実行します。サンプル Java クライアントがサーバーと通信したことがある場合、キャッチアップを使用してアクティブなリースクエリを試行します。サンプル Java クライアントは、キャッチアップを伴うアクティブなリースクエリが、クライアントがしばらくダウンしていたか、DHCPサーバーが再ロードされた場合など、キャッチアップ データが使用できないという場合にのみ、バルク リースクエリを実行します。

サンプル Java クライアントは、複数の VPN および複数のサーバーを持つ構成をサポートしています。ただし、サンプルの Java クライアントでは、これらのサーバー間のリースは VPN および IP アドレスに関して一意であると想定しています。2 つのサーバーが VPN またはグローバル名前空間で同じ IP アドレスをリースしている場合、SQL データベースには 2 つのリースのうちの 1 つだけのレコードが含まれます。これは、フェールオーバー ペアではなく、2 つの独立した DHCP サーバーに適用されます。また、SQL データベースを最新の状態に保つために、フェールオーバー ペアの両方のフェールオーバー パートナーと通信するようにサンプルの Java クライアントを構成する必要があります。

（注） サンプル Java クライアントは、インストール・パス/例/dhcp/cnrnotify.jar で入手できます。cnrnotify-readme.txt という名前のテキスト ファイルも、そのディレクトリに用意されており、最初に読み取る必要があります。

例/dhcp/cnrnotify.jar は、次の zip ファイルを含む zip ファイルです。

- サンプル Java クライアントソースコードと Javadoc ドキュメント。
- たとえば、lnc.properties ファイルと lnc6.properties ファイルを指定します。(使用可能なプロパティの詳細については、-listpropsオプションを指定してクライアントを実行します。
- Cisco Prime ネットワーク レジストラー実装のための一括およびアクティブなリースクエリ インターネット ドラフト。
- Cisco Prime Network レジストラー専用リース情報に使用されるメッセージ値、オプションコード、ベンダー固有のデータを詳しいドキュメント。インターネット割り当て番号機関(IANA)は、バルクおよびアクティブリースクエリインターネットドラフトで使用されるメッセージおよびオプションコードにまだ値を割り当てていないため、Cisco Prime Network レジストラーで使用される値について説明します。

これらの項目を抽出するには、Winzip などの zip ツールを使用して cnrnotify.jar ファイルを開きます。(cnrnotify-readme.txt ファイルを参照してください)。Javadoc を抽出するには、次の使用をお勧めします。

```
jar xvf cnrnotify.jar docs_notify
```

上記のコマンドは、ドキュメントを抽出するために使用されます。

DHCPv4 サブサブ オプション コード

次の表は、DHCPv4 リースクエリの要求時に使用されるサブサブ オプション コードの一覧です。これらのコードは cnrnotify-プロトコル-numbers.txt ファイルに存在し、cnrnotify.jar zip ファイルで使用できます。

表 *29 : DHCPv4* サブ-サブ オプション コード

サブサブ オプション コード	オプション名	オプション タイプ
1	oro	サブサブ オプション番号の 1 バイト以上
2	状態	バイト
3	data-source	バイト
4	start-time-of-state	基準時間からの過去の期間
5	ベースタイム	絶対時間(1970年からの秒)
8	クライアント クラス名	文字列 (ゼロ終了なし)
9	パートナー-最終トランザクション時間	base-time からの経過時間
10 0xa	client-creation-time	base-time からの経過時間
11 0xb	制限 ID	制限 ID を含む blob
12 0xc	バインディング開始時刻	base-time からの経過時間
13 0xd	バインディング終了時刻	基準時からの将来/過去の期間を表す負/正の値
14 0xe	fwd-dns-config-name	文字列（0 で終了しない）
15 0xf	レブ・DNS-コンフィグ名	文字列（0 で終了しない）
16 0x10	client-override-client-id	クライアントのクライアント ID を含む blob
17 0x11	ユーザー定義データ	文字列（0 で終了しない）
18 0x12	scope-name	文字列（0 で終了しない）
19 0x13	フェールオーバー状態シリアル番号	4 バイト整数, ネットワークの順序
20 0x14	予約キー	blob、タイプバイトで始まる: • 0x2e、46: ゼロ終了なしの文字列 • 0x7、7:ブロブ

サブサブ オプション コード	オプション名	オプション タイプ
21 0x15	クライアント-prl	クライアントのパラメーター要求リスト、DHCPv4 オプション コードの BLOB

DHCPv6 サブサブ オプション コード

次の表は、DHCPv6 リースクエリの要求時に使用されるサブサブ オプション コードの一覧です。これらのコードは cnrnotify-protocol6-numbers.txt ファイルにも存在し、cnrnotify.jar zip ファイルで使用できます。

表 *30 : DHCPv4* サブ-サブ オプション コード

サブサブ オプション コード	オプション名	オプション タイプ
1	oro	サブサブ オプション番号の 1 バイト以上
2	状態	バイト
3	data-source	バイト
4	start-time-of-state	基準時間からの過去の期間
5	ベースタイム	絶対時間(1970年からの秒)
8	クライアント クラス名	文字列 (ゼロ終了なし)
9	パートナー-最終トランザクション時間	base-time からの経過時間
10 0xa	client-creation-time	base-time からの経過時間
12 0xc	バインディング開始時刻	base-time からの経過時間
13 0xd	バインディング終了時刻	基準時からの将来/過去の期間を表す負/正の値
14 0xe	fwd-dns-config-name	文字列（0 で終了しない）
15 0xf	レブ・DNS-コンフィグ名	文字列（0 で終了しない）
16 0x10	検索キー	クライアントのクライアント ID を含む blob
17 0x11	ユーザー定義データ	文字列（0 で終了しない）
18 0x12	prefix-name	文字列（0 で終了しない）
19 0x13	フェールオーバー状態シリアル番号	4 バイト整数, ネットワークの順序

サブサブ オプション コード	オプション名	オプション タイプ
20 0x14	予約キー	blob、タイプバイトで始まる: • 0x2e、46: ゼロ終了なしの文字列 • 0x7、7:ブロブ
21 0x15	フェールオーバー パートナーの有効期間	base-time からの未来/過去の経過時間を表す負または正の値
22 0x16	フェールオーバー - 次のパートナーの有効期間	base-time からの未来/過去の経過時間を表す負または正の値
23 0x17	フェールオーバーの有効期限	base-time からの未来/過去の経過時間を表す負または正の値
24 0x18	クライアントオロ	クライアントの ORO、 DHCPv6 の BLOB 2 バイト オプション コード

サンプル Java クライアントの要件

サンプル Java クライアントの要件は次のとおりです。

- JDK 1.8
- JDK 1.8 の java.sql パッケージ。
- JDBC ドライバーと互換性のあるデータベースのインストール。データベースには、事前定義された列セットを含む特定のテーブルが存在する必要があります。

ヒント　テーブルが存在しない場合は、-c オプションを指定してクライアントを実行します。テーブルが作成されます。

MySQL の要件は次のとおりです。

- MySQL サーバーの最新バージョン。
- MySQL の JDBC コネクタ。
- サンプル Java クライアントの状況とエラーをログに記録するための log4j パッケージ。

（注）　MySQL-8.0.22 データベース、mysql-connector-java-8.0.21.jar、および log4j-1.2.17.jar を使用することを推奨します。

抽出され、lnc.properties ファイルが構成されたら、サンプルの Java クライアントを次の方法で実行できます。

ステップ 1 同じディレクトリに 3 つの .jar ファイル（cnrnortify.jar、mysql-connector-java-8.0.21.jar、log4j-1.2.17.jar）を配置してください。

ステップ 2 同じディレクトリ内の lnc.properties/lnc6.properties ファイルを抽出します。

DHCPv4 クライアントの場合:

```
jar xvf cnrnotify.jar com/cisco/cnr/notify/lnc.properties
```

DHCPv6 クライアントの場合:

```
jar xvf cnrnotify.jar com/cisco/cnr/notify/lnc6.properties
```

ステップ 3 lnc.properties/lnc6.properties ファイルを構成します。

ステップ 4 Java実行可能ディレクトリが現在のパスにある場合、サンプル・クライアントは次の方法で実行されます。

DHCPv4 の場合:

```
java -cp .:cnrnotify.jar:mysql-connector-java-8.0.21.jar:log4j-1.2.17.jar
com/cisco/cnr/notify/LeaseNotificationClient
```

DHCPv6 の場合:

```
java -cp .:cnrnotify.jar:mysql-connector-java-8.0.21.jar:log4j-1.2.17.jar
com/cisco/cnr/notify/LeaseNotificationClient6
```

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

Web UI は、構成属性を表示および管理し、関連サーバーの情報を表示します。リース クエリに関する統計情報は、[DHCP サーバーの統計情報] ページで確認できます。

ステップ 1 Deploy メニューで、[DHCP] サブメニューから DHCP Server を選択し、[DHCPサーバーの管理（Manage DHCP Server）] ページを開きます。

ステップ 2 [**統計情報**] タブをクリックして、[DHCP サーバーの統計情報] ページを開きます。

このページに、サーバー統計の詳細情報が表示されます。

CLI コマンド

既存のdhcp getRelatedServersコマンドは、DHCP リスナーとアクティブな接続に関する情報を提供するために拡張されます。

```
nrcmd> dhcp getrelatedservers
```

（注） このコマンドは、ローカル クラスターでのみ使用できます。

DHCP リスナーの設定

DHCP リスナ構成を使用して、TCP 接続を介して DHCP サーバーに対するアクティブおよびバルク リースクエリを有効にするようにオブジェクトを構成できます。DHCP サーバーが複数の TCP ポートでの接続のリッスンをサポートするか、サーバーが受信接続を受け入れるアドレスを制限する必要がない場合は、単一のオブジェクトで十分です。

ローカル アドバンスド Web UI

ステップ 1 メニューからDeploy 、Listenersサブメニューの下DHCPでを選択して、[DHCP TCP リスナーの一覧/追加] ページを開きます。

ステップ 2 [リスナー（Listeners）]ペインの[リスナーの追加（Add Listeners）]アイコンをクリックし、[名前（Name）] フィールドに名前を入力して、[TCPリスナーの追加（Add TCP Listener）] をクリックします。

ステップ 3 サーバーが接続を受け入れるインターフェイスを制限するために、アドレス/ip6address フィールドに IP アドレスを入力します。これは通常、指定されていません。IPv6 リスナーを設定する場合は、ip6address を入力します。アドレスと ip6 アドレスの両方が指定されていない場合は、IPv4 アドレス 0.0.0.0 が使用されます。

TCP 接続が受け入れられるアドレスを制限するには、アドレス (IPv4 の場合) または ip6address (IPv6 の場合) 属性を入力します。どちらの属性にも値が入力されていない場合、ホストの IPv4 アドレスへの IPv4 接続は受け入れられます。IPv6 経由の接続を指定するには、ip6address 属性に値を入力する必要があります (0::0ホストの IPv6 アドレスへの接続を受け入れる場合に使用できます。両方の属性ではなく、両方の属性にのみ値を入力できます。

（注） DHCP サーバーに対して IPv4 と IPv6 の両方のリスナーを指定することはできません。

ステップ 4 デフォルト値が適切でない場合は、ポートフィールドにポートの値を入力します。デフォルトのポートは、DHCPv4 のサーバーポートと DHCPv6 のサーバーポートです。

ステップ 5 enable 属性に対しては、[真（true）] または [偽（false）] ラジオボタンをクリックします。デフォルト値は true です。

ステップ 6 デフォルト値の 10 が適切でない場合は、max-connections の値を入力します。

ステップ 7 デフォルト値の 120 が適切でない場合は、leasequery-backlog-time の値を入力します。

ステップ 8 leasequery-send-all 属性に対しては、[真（true）] または [偽（false）] ラジオボタンをクリックします。デフォルト値は false です。

ステップ 9 [保存（Save）] をクリックします。

CLI コマンド

DHCP リスナ コマンドを次の表に示します。

表 *31 : DHCP* リスナ コマンド

操作	コマンド
作成（Create）	dhcp-listener name create [attribute=value]
削除（Delete）	dhcp-listener name delete
リスト（List）	dhcp-listener list
名前を一覧表示する	dhcp-listener listnames
表示（Show）	dhcp-listener name show
設定	dhcp-listener name set attribute=value [attribute=value ...]
get	dhcp-listener name get attribute
設定解除	dhcp-listener name unset attribute
有効	dhcp-listener name enable attribute
[無効（Disable）]	dhcp-listener name disable attribute

リース履歴データベース圧縮ユーティリティ

cnr_leasehist_compressユーティリティは、地域クラスタ(DHCPv4)リース履歴データベースを圧縮するために、Cisco Prime Network レジストラーに追加されました。このユーティリティは、データベース内のデータを直接圧縮するのではなく、既存のデータを、可能な限りコンパクトに最適化された新しいデータベースにコピーします。このユーティリティは、シスコ Web サイトの Cisco Primeネットワーク レジストラーダウンロード セクションからダウンロードできます。

注意 cnr_leasehist_compressユーティリティは、地域のクラスターリース履歴データベースでのみ使用し、特に DHCPRELEASE パケットのためにデータベースが大幅に増加したと思われる場合に使用します。

コピー操作中に、このユーティリティを使用して次の操作を行うことができます。

- 一定の時間間隔より古いレコードをトリミングする - 通常は–t、このオプションを使用します。このオプションで指定する間隔は、ネットワークレジストラー時間間隔形式を使用します。たとえば、3030d日または1y1 年間です。
- 同じリースとクライアントに属するレコードのマージ: このcnr_leasehist_compress ユーティリティを使用して、IP アドレスのリースを解放した後にリースを解放したクライアントに属するレコードをマージします。通常、–m オプションを使用します。このオプションで

指定する間隔は、ネットワークレジストラー時間間隔形式を使用します。たとえば、120120s 秒または2m2 分間です。

レコードのマージ中に、このユーティリティは、突然終了したリース履歴レコードや、バインドの終了時刻が正しくない(後続のリース操作によって発生した可能性がある)を修正します。レコードをマージするこのオプションは、サーバーに追加の負荷を生じさせる特定のルーター構成によって作成される膨大な数のレコードにも対応します。

ユーティリティを実行するcnr_leasehist_compress 前に、次の手順を実行します。

- ネットワークレジストラー地域クラスターを停止します。アクティブな地域クラスタデータベースでは動作しません。
- 既存のリース履歴データを単独で圧縮するために使用できることに注意してください。リージョン クラスターが将来のリース履歴レコードを収集する方法は変更されません。チャット クライアントが疑われる場合は、DHCP サーバーが DHCPRELEASE メッセージを処理しないことを確認します。このような場合は、ユーティリティを定期的に実行する必要があります。
- サービス プロバイダであり、一部のデバイスで DHCPDISCOVER、DHCPOFFER、DHCPREQUEST、DHCPACK のシーケンスを繰り返し生成するなどの既知の問題が発生し、30 以降に発生する可能性があるため、サービス プロバイダであり、ネットワーク内の地域リース履歴が増加していると疑われる場合に使用できます。メッセージを送信します。すべての DHCPRELEASE メッセージをドロップするか、または設定されたしきい値を超えるクライアントに属するメッセージをドロップするかを選択できます。
- 新しいデータベースは最適な方法で書き込まれます。新しいデータベースは、最初はかなりの速度で拡張できますが、追加のリース履歴レコードが収集された後、通常の状態に戻ります。

Cnr_leasehist_compress の実行に関する全般的なコメント

注意 この手順のすべての手順に慎重に従ってください。いずれかの手順を省略すると、リース履歴データが失われる可能性があります。各タスクに関連するリース履歴データベースをメモします。リース履歴レコードの数とレコードのトリミングまたはマージにかかる時間によっては、このユーティリティの実行に数時間または数日かかる場合があります。実行が完了する前にサーバーが再起動した場合は、実行中にユーティリティを中断できます。後で再開できます。ただし、前の実行で使用したのと同じオプションを指定する必要があります。

インストールパスは、Cisco Prime Network Registrar をインストールするパスです。

次の表に、このユーティリティの限定オプションをcnr_leasehist_compress 示します。

表 *32 : cnr_leasehist_compress* オプション

オプション	説明
–a	一時アクティブ データベース内のすべてのリース履歴レコードを、新しいデータベースのリース履歴レコードに追加します。
–c 制限	クライアントに対してマージされたレコード数が指定数を超えた場合にレポートを生成します。–fこのオプションを使用すると、これらのレコードはログ・ファイルに転送されます。
–C	書き込み時にリース・レコードを圧縮する (詳細については、CCM のリース・ヒスト圧縮属性を参照してください。
–d path	圧縮されたリース履歴レコードを含む新しい転送先データベースへのパスを指定します。
アトリスト–e	除外されたマージ属性リストを上書きします。
–f file	ほとんどのリース履歴レコードの警告をログ ファイルにリダイレクトします。
–g	dbtxn-seq 属性と dbtxn-generation 属性を使用して、宛先データベースに書き込まれているすべてのリース履歴レコードに割り当てられた番号順に新しいシーケンスを生成します。
–i ipaddr	特定の IP アドレスのレコードをログ ファイルに転送します。
–l 制限	データベースが事前に設定された 20 ファイルの制限に達した後に、ログ ファイルをパージします。
–mタイムイント	特定binding-start-timeのリースのが、以前のリースのbinding-end-time期間内にある場合にリース レコードをマージします。このオプションの推奨値は、120s です。
–n	隣接するレコードをマージせずに比較します。
–p	詳細なリース履歴レコードを削除します。このオプションは、詳細リース履歴を有効にしている場合にのみ使用できます。 （注） シスコ プライム ネットワーク レジストラーは、詳細なリース履歴をサポートしなくなりました。ただし、詳細なリース履歴をサポートしているバージョンからのアップグレードの場合、このオプションは保持されます。
–q records	ユーティリティの実行中に生成される定期的な進行状況レポートの間隔を設定します。デフォルト値は 100000 です。次に例を示します。 `+00:00:18 Read 100000 records (0 bad); trimmed 6717; merged 73370; 19912 written (19.91%)`

オプション	説明
–r records	ソース データベースから読み取られるレコードの数を制限します。
–s path	データを新しいデータベースにコピーするソース データベースを指定します。
–t age	特定の時間間隔より古いレコードをトリミングするための値を指定します。このオプションには、11y年または30d30 日間など、標準のネットワークレジストラー時間間隔を使用します。
–v	バージョンを出力して終了します。
–w records	転送先データベースに書き込まれるレコードの数を制限します。
–y"ライン attr"	リース履歴レコードのダンプの幅を変更します。このオプションは推奨されません。ただし、132 列132 30の出力には値を使用できます。
–z{文字}=レベル	標準のネットワーク レジストラーデバッグ トレース構文を使用して指定されたデータベースをデバッグします。

圧縮の実行

cnr_leasehist_compress ユーティリティを実行するには、次の手順を実行します。

ステップ 1 LD_LIBRARY_PATHにインストールパス/lib を追加して、ユーティリティにネットワーク レジストラー IPライブラリへのアクセスを提供します。

```
$ bash
# export LD_LIBRARY_PATH=install-path/lib:$LD_LIBRARY_PATH
```

ステップ 2 ネットワークレジストラー地域クラスターを停止します。

```
# systemctl stop nwregregional
```

ステップ 3 元のインストールパス/data/leasehist ディレクトリの名前をインストールパス/data/oldleasehist に変更します。/leasehist ディレクトリは元のデータベースになります。

```
# mv install-path/data/leasehist
# install-path/data/oldleasehist
```

ステップ 4 新しいリースディレクトリを作成します。

```
# mkdir install-path/data/leasehist
```

ステップ 5 ユーティリティ cnr_leasehist_compress を 実行して、地域のクラスターがアクティビティを再開できるようにします。

```
# install-path/bin/cnr_leasehist_compress
> -r 0
> -s install-path/data/oldleasehist
> -d install-path/data/leasehist
> -p
```

注意 これらのコマンドを実行しても、元のデータベースは圧縮されません。–rこの0オプションは、一時的なアクティブ・データベースを作成するようにユーティリティーに指示するので、非常に重要です。ユーティリティが元のデータベースを圧縮している間、地域クラスタはアクティブなままです。

ステップ 6 ネットワークレジストラーの地域クラスターを再起動します。

```
# systemctl start nwregregional
```

ただし、この時点では、元のデータベースからリース履歴データを取得することはできません。リージョンクラスターは、新しいリース履歴データを収集し、一時的にアクティブなデータベースに転送します。次に、このユーティリティは、新しいリース履歴データを新しいデータベースにマージします。

ステップ 7 インストールパス/data/newleasehist という新しいディレクトリを作成します。この /newleasehist ディレクトリが新しいリース履歴データベースになります。

```
# mkdir install-path/data/newleasehist
```

ヒント 地域クラスターが新しいデータベースにデータを取り込んだ後、必要に応じてこの新しいディレクトリを別のパーティションに作成し、最終的な場所にコピーできます。

ステップ 8 ユーティリティを cnr_leasehist_compress 実行して、元のデータベースを新しいデータベースにトリミング、マージ、および圧縮します。

```
# install-path/bin/cnr_leasehist_compress
> -s install-path/data/oldleasehist
> -d install-path/data/newleasehist
> -t trim-time-interval
> -m merge-time-interval
> -f /tmp/cnr-compress.log
```

元のデータベースに詳細なリース履歴レコードが含まれている場合は、–pこのオプションを使用して、これらのレコードを新規データベースに転送しないことがユーティリティーに許可されることを確認する必要があります。それ以外の場合、ユーティリティは実行されません。

（注） シスコ プライム ネットワーク レジストラーは、詳細なリース履歴をサポートしなくなりました。ただし、詳細なリース履歴をサポートしているバージョンからのアップグレードの場合、このオプションは保持されます。

ステップ 9 ユーティリティが元のデータベース全体を処理した後、新しいデータベースに新しいリース履歴レコードを追加するには、次のタスクを実行します。

（注） 次の手順を完了するまで、地域クラスターを再起動しないでください。次の手順でシステムが再起動する場合は、この手順を繰り返します。

a) Network Registrar のリージョン クラスタを停止します。

```
# systemctl stop nwregregional
```

b) このユーティリティcnr_leasehist_compress を実行して、新しいリース履歴レコードを新しいデータベースに追加します。

```
# install-path/bin/cnr_leasehist_compress
> -a
> -s install-path/data/leasehist
```

```
> -d install-path/data/newleasehist
> -m merge-time-interval
> -f /tmp/cnr-append.log
```

注意 この–aオプションは、ユーティリティが一時アクティブ・データベースのリース履歴レコードを新規データベースのリース履歴レコードに追加する必要があることを示すため、重要です。元のデータベースに使用したのと同じマージ時間間隔値を使用することをお勧めします。

c) ユーティリティが新しく収集したリース履歴レコードを追加するタスクを完了したら、一時アクティブデータベースディレクトリの名前をインストールパス/data/leasehist から install-path/data/tmpleasehist に変更します。

```
# mv install-path/data/leasehist
# install-path/data/tmpleasehist
```

d) 新しいデータベース ディレクトリの名前を変更します, インストール パス /data/newleasehist,インストール パス/data/leasehist として:

```
# mv install-path/data/newleasehist
# install-path/data/leasehist
```

ステップ 10 Network Registrar のリージョン クラスタを起動します。

```
# systemctl start nwregregional
```

ステップ 11 Network Registrarの Web UI を使用して、リージョン リースの履歴データを確認します。

ステップ 12 インストールパス/data/oldleasehist、および一時的なアクティブなデータベースをインストールパス/data/tmpleasehist にアーカイブします。データベースをアーカイブするときに、すべてのサブディレクトリとファイルを必ず含めます。

ステップ 13 元のデータベースと一時的なアクティブなデータベースを削除します。

```
# rm -rf install-path/data/oldleasehist
# rm -rf install-path/data/tmpleasehist
```

柔軟なリース時間

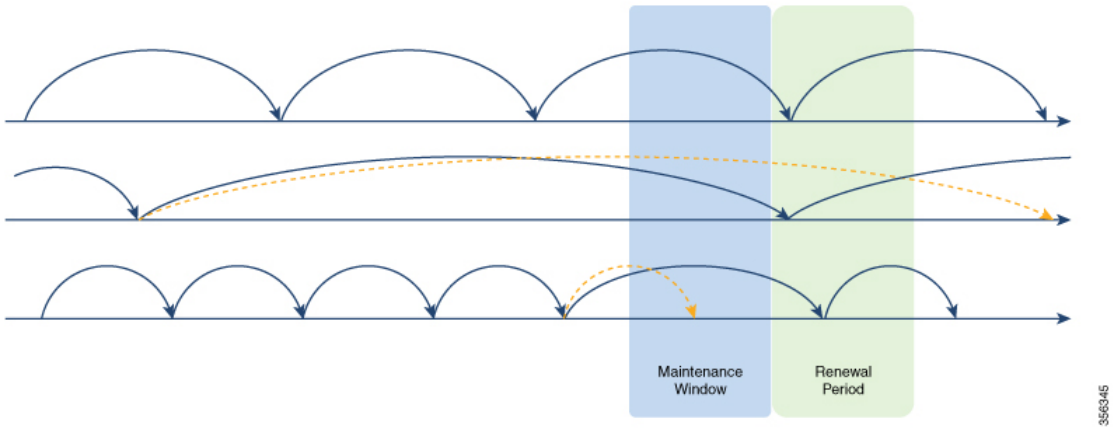
特定のネットワークセグメントの番号を変更したり、構成の変更を迅速に有効にしたりする必要があるため、ネットワークの再構成が必要になる場合があります。通常、これは、変更の前にクライアントのリース時間を短縮し、変更を適用し、リース時間を元の値に戻すことによって行われます。つまり、更新時間を比較的狭いウィンドウ (メンテナンスウィンドウ) に圧縮し、サーバーの負荷を均等に戻す必要があります。これらの手順は手動で、エラーが発生しやすいものです。Cisco Prime Network レジストラーは、メンテナンス期間の前、中、および後の DHCP サーバーの更新負荷を軽減するために、このプロセスを自動化するのに役立ちます。

ネットワークの再設定のスケジューリング

Cisco Prime Network レジストラーでは、メンテナンス期間をスケジュールして、エラーが発生しやすく、リース時間のリセットを忘れないようにすることができます。必要なメンテナンス期間の開始時刻、終了時刻、および更新期間を設定できます。また、メンテナンス期間をサーバー全体に適用するか、特定のスコープ、リンク、またはプレフィックスにのみ適用するかを指定できます。サーバーは、この期間中にサーバーがシャットダウンする可能性があるため、保守期間の開始時刻と終了時刻の間にクライアントがDHCPサーバーに接続しようとするのを避けるために、リース、更新 (T1)、および再バインド (T2) の時間を調整しようとします。メンテナンス期間中、DHCPサーバーは最小リース時間を使用し、メンテナンス期間の後はリース時間を元の状態に戻しますが、更新は広がったままにします。再構成中および再構成後の更新は、サーバー負荷の急増を最小限に抑えるために適切に分散されます。最終的には、メンテナンス ウィンドウの構成を削除するか、新しい構成に置き換えることができます。サーバーは、過去に発生した保守を無視します。

図 13 : Maintenance Window （297 ページ）は、異なるリース時間を持つ 3 つのクライアントを示し、メンテナンスウィンドウと対話します。最初の(上)のケースでは、更新期間中にクライアントが入ってくるので、変更はありません。2 番目 (中央) の場合、サーバーは更新期間中にクライアントが更新されるように時間を短縮します。3 番目 (下位) の場合、サーバーはメンテナンス期間中にクライアントの更新を回避するために時間を増やします (サーバーに到達できない可能性があるため)。

図 *13 : Maintenance Window*



1 つのメンテナンスウィンドウを作成、編集、および削除できます。DHCP サーバーは、構成されている場合はメンテナンス ウィンドウを読み込みます。現在の時刻が終了時刻に更新期間を加算した場合 (すべてのクライアントが更新された構成を持つ必要があるメンテナンス期間の終了後の時間間隔) は、メンテナンスウィンドウをロードするために無視されます。また、リース更新の配布が有効になっていない場合も読み込まれません (を参照リース更新の配布（299ページ）)。メンテナンス期間が適用されるスコープ、リンク、またはプレフィックスの場合、サーバーは次のようにクライアントに送信されるリース時間または更新時間を変更します。

- メンテナンス期間の終了前にクライアントに与えられたリース時間は、メンテナンス期間の終了時刻に更新期間を加えた時間を超えないことを示します。
- メンテナンス期間の終了前にクライアントに与えられた更新時間は、メンテナンス期間の終了時刻に更新期間を 1/2 を加えた時間を超えないことをお知りください。
- メンテナンス期間の開始時刻と終了時刻の間に終了するクライアントに与えられたリース時間は、メンテナンス期間の終了後と終了時刻の前の時間間隔に時間を加えた後の間隔を 1/2 に加えた間隔の間のどこかで期限切れに調整されます。
- メンテナンス期間の開始時刻から終了時刻の間に発生するクライアントに与えられた更新時間は、メンテナンス期間の終了から更新期間の 1/2 までの間に更新をトリガーするように調整されます。

（注） フェールオーバー時間の制限は引き続き適用され、メンテナンス期間が原因で変更されません。これらの制限により、サーバーがリース、更新(T1)、および一部のクライアントの再バインド (T2) の時間を最適化できなくなる可能性があります。

メンテナンス期間オブジェクトの追加

メンテナンス ウィンドウ オブジェクトを追加するには、次の手順を実行します。

ローカル アドバンスド Web UI

ステップ 1 [**展開**] メニューの[DHCP]サブメニューの [**メンテナンス ウィンドウ**] を選択します。[メンテナンス ウィンドウの一覧/追加] ページが開きます。

ステップ 2 左側のペインで [**メンテナンス ウィンドウの追加**] アイコンをクリックし、次のフィールドに詳細を入力します。

- **名前**：DHCP メンテナンス ウィンドウのオブジェクトの名前。
- **開始日**— メンテナンス期間が開始される日時。これは、DHCP サーバーが停止すると予想される場合です。
- [**終了日**]- メンテナンスウィンドウが終了する日時。これは、DHCP サーバーが再び利用可能になると予想される場合 (構成の変更が行われた後) です。
- [**更新期間**]- 影響を受けるすべてのクライアントが新しく構成された情報を受け取るためにサーバーに接続する必要があるメンテナンス期間の終了後の期間。

ステップ 3 [**メンテナンス ウィンドウの追加**] をクリックします。

ステップ 4 メンテナンスウィンドウを特定のスコープ、プレフィックス、またはリンクに適用する場合は、次の操作を行います。

- [DHCPスコープの一覧]ページで[一覧] を有効に設定したメンテナンス属性を持つスコープが、[スコープ] 領域の下に表示されます。メンテナンス期間を特定のスコープに適用するには、[**スコープの構成**]

オプションの横にある**無効な**ラジオ ボタンをクリックし、[スコープ] 領域から必要なスコープを選択または追加します。**有効**にされたラジオ ボタンをクリックすると、構成内のすべてのスコープが現在のメンテナンス ウィンドウに参加します。

- [リスト/追加 DHCP v6 プレフィックス] または [リスト/追加 DHCP v6 リンク] ページでメンテナンス属性を持つプレフィックス/リンクが有効に設定されているリンクまたはプレフィックス領域の下に一覧表示されます。メンテナンス ウィンドウを特定のプレフィックスまたはリンクに適用するには、[**プレフィックス/リンクの設定**] オプションの横にある**無効な**ラジオ ボタンをクリックし、[リンク] 領域または[プレフィックス]領域から必要なリンクまたはプレフィックスをそれぞれ選択または追加します。**有効**なラジオ ボタンをクリックすると、構成内のすべてのプレフィックスとリンクが現在のメンテナンス ウィンドウに含まれます。

ステップ 5 [保存（Save）] をクリックします。

メンテナンスウィンドウオブジェクトの詳細は、メンテナンスウィンドウ編集ページで編集できます。メンテナンス ウィンドウ オブジェクトを削除するには、左側のウィンドウでメンテナンス ウィンドウ オブジェクトの名前を選択し、左側のウィンドウで[**選択したメンテナンス ウィンドウの削除**] アイコンをクリックして、削除を確認します。

（注） メンテナンス ウィンドウ オブジェクトを削除すると、スコープ、プレフィックス、およびリンクのメンテナンス属性もすべてクリアされます。

CLI コマンド

保守ウィンドウ オブジェクトを作成するには、dhcp-**メンテナンス ウィンドウ**名 create [属性=value ..] コマンドを使用します。保守ウィンドウ オブジェクトを削除するには、dhcp-**メンテナンス ウィンドウ**名削除**コマンドを**使用します。

dhcp**メンテナンス ウィンドウクリア**を使用するメンテナンス [dhcpv4 |dhcpv6]をクリックして、すべてのスコープまたはすべてのプレフィックス/リンクのメンテナンス フラグをクリアします。dhcpv4を指定すると、スコープのみがクリアされます。dhcpv6を指定すると、プレフィックス/リンクのみがクリアされます。どちらも指定しない場合、すべてクリアされます。

すべてのメンテナンス ウィンドウ コマンドの完全な一覧については、/docs ディレクトリの CLIGuide.html ファイルのdhcp-maintenance-windowコマンドを参照するか、CLI の**ヘルプ** dhcp-**メンテナンス ウィンドウ**を使用してください。

リース更新の配布

DHCPサーバーは、リース更新の負荷が可能な限り均等に分散されるようにクライアントの更新を調整し、更新トラフィックの急増を回避します。更新トラフィックの急増は、多数のクライアントが一度に戻るメンテナンスウィンドウ、ネットワーク (または停電) の後に発生する可能性があります。そのようなスパイクを回避するために、この機能はデフォルトでは有効になっています。

サーバーは、バケット間隔内に更新するクライアント数を保持します。サーバーがクライアントの更新時間 (リース時間の 50%) を決定すると、そのバケットの値が標準 (クライアント数/(最新の更新時間/バケット間隔) を超えているかどうかを確認します。標準を超えると、サーバーは更新時間の 20～120% のランダムな値を選択し、そのバケットを標準と照らしてチェックします。このプロセスは、基準を下回るバケットが見つかるまで、または満たされていないバケットの時間が使用されるまで、限られた回数だけ繰り返されます。

（注） バケットが 10 更新/秒未満の場合、サーバーはその負荷を簡単に処理できるため、サーバーはカウントを調整しません。

図 *14 :* リース更新の配布の例

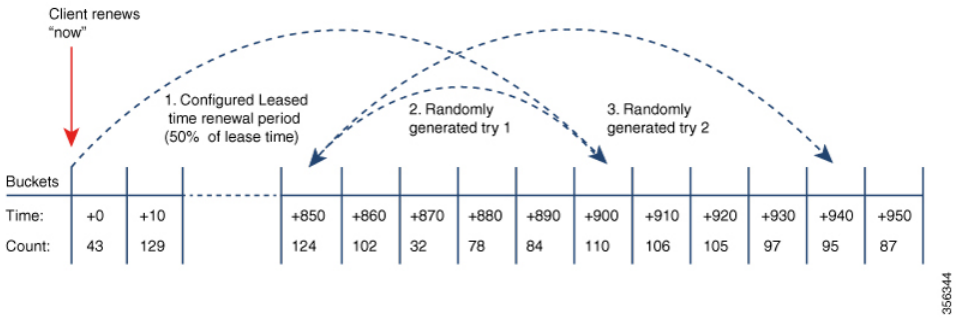


図 14 : リース更新の配布の例 （300 ページ）更新の配布機能の例を示します。この例では、クライアントの通常の更新時間 (リース時間の 50% が 1800 秒 = 900 秒) のバケットが、そのバケットの期間中に更新される予定クライアントのしきい値を超えた場合、サーバーは更新時間を調整します。ここでは、サーバーはランダムな代替更新時間(元の更新の20%から120%の間)を選択します。ただし、最初の試みもしきい値を超えているため、セカンダリ試行が試行され、更新時間 (944) がしきい値を下回るバケット内にあることが検出されます。クライアントには、その更新時間 (944 秒) が与えられます。

DHCPv4 の場合、この機能が有効になっている場合、サーバーは強制的に dhcp 更新時間オプション (58) とdhcp 再バインド時間オプション (59) を送信します。DHCPv6 の場合、サーバーは常に IA_NA および IA_PD オプションの T1/T2 フィールドを設定するので、その処理に影響はありません。

更新の配布機能の制御

更新の配布機能を制御するには、次の手順を実行します。

ローカル アドバンスド Web UI

ステップ 1 [操作（Operate）] メニューの [サーバ（Servers）] サブメニューの下の [サーバーの管理（Manage Servers）] をクリックし、[ローカルDHCPサーバ（Local DHCP Server）] リンクをクリックして [ローカルDHCPサーバの編集（Edit Local DHCP Server）] ページを開きます。

ステップ 2 [分散更新（Distributed Renewals）] セクションで、次の属性を設定します。

- distribute-renewals：DHCP サーバーが更新時間を調整してサーバーの更新負荷を平滑化できるかどうかを制御します。

  （注） 設定されているポリシーの dhcp-lease-time オプション（51）または優先存続期間が 180 日より長く設定されている場合、サーバーはこの機能を有効にしません。

- distribute-renewals-max-renewal-time：サーバの更新負荷を円滑に調整するために、サーバーが更新を調整する際に使用する最大更新時間を制御します。この属性が設定されていない（または0）場合、サーバーは dhcp-lease-time オプションの 50%（51）またはすべての名前付きポリシーと組み込みポリシーの優先存続期間に基づいてこれを決定します。

- distribute-renewals-bucket-interval：サーバーの負荷を円滑にするために使用されるバケットの時間間隔を制御します。この属性が設定されていない場合、バケット数が 100,000 を超えない限り、サーバは10秒を使用します。この場合、サーバーは時間間隔を使用してバケットを最大100,000に制限します。

ステップ 3 [保存（Save）] をクリックします。

CLI コマンド

配布更新機能を無効にするには、dhcp を使用してdhcp disable distribute-renewals にします。配布更新機能を有効にするには、dhcp を**使用して配布更新を有効にします。また、**dhcp setコマンドを使用して、配布更新-最大更新時間および分散更新-バケット間隔の値を変更することもできます。

DHCP 更新レポートの表示

ローカル Web UI の [DHCP 更新レポート] ページには、DHCP サーバー上で予想される更新の負荷がグラフィカルに表示されます。これは、特定の時間間隔(バケット)で将来更新される予定のクライアントの数を示します。

Web UIのダッシュボードから更新データを確認することもできます。詳細については、DHCP 更新データ （503 ページ）を参照してください。

DHCP 更新レポートを表示するには、次の手順を実行します。

ローカルWeb UI

ステップ 1 [操作（Operate）] メニューで、[サーバー（Servers）] サブメニューから [サーバーの管理（Manage Servers）] を選択し、[ローカルDHCPサーバー（Local DHCP Server）] リンクをクリックして [ローカルDHCPサーバーの編集（Edit Local DHCP Server）] ページを開きます。

ステップ 2 [DHCP**更新レポート**] タブをクリックします。

ステップ 3 [バケット数]フィールドに、希望するバケット**数を入力します**。更新データが報告されるバケットの数を指定します。バケットは、その時間間隔中に更新する予定のクライアントを表します。

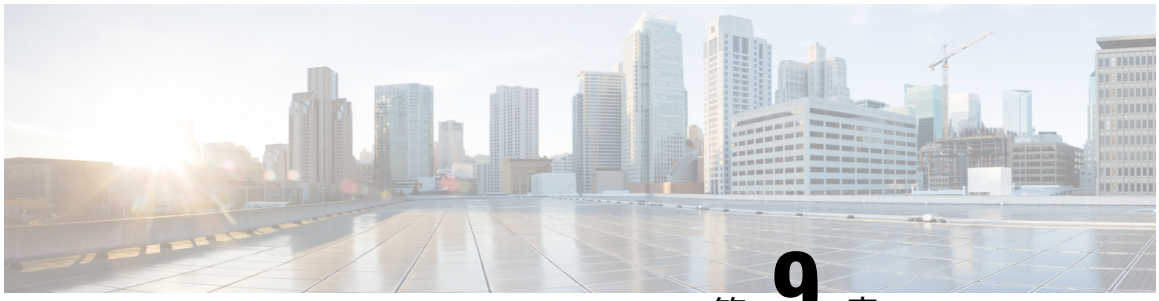
ステップ 4 [表示（Show）] をクリックします。

DHCP 更新データはグラフ形式で表示され、Y 軸に沿って特定の間隔で更新するクライアント数と X 軸に沿って日付/時刻のスタンプを更新します。

CLI コマンド

配布更新機能に関連する情報を報告するには、dhcp getRenewalData [max-buckets] を使用します。既定では、時間の経過に伴う予想されるクライアント更新数は、20 個のバケットに最も多く表示されますが、この値は希望する数を指定することでオーバーライドできます。

これは、設定に関するいくつかの情報と、各更新バケット内のクライアント数の(文字セル)グラフも表示します。



第 9 章

DNS 更新の管理

DNS 更新プロトコル (RFC 2136) は、DNS と DHCP を統合します。後者の 2 つのプロトコルは相互補完します。つまり、DHCP は、IP アドレス割り当てを集中化および自動化し、ダイナミック DNS 更新は、割り当てられたアドレスとホスト名の間のアソシエーションを自動的に記録します。DHCP を DNS 更新を使用する場合、ホストが IP ネットワークに接続するときに、必ずそのホストのネットワーク アクセスを自動的に設定します。固有の DNS ホスト名を使用してホストを検索し、ホストにリーチできます。たとえば、モバイルホストは、ユーザーや管理者の介入なしで、自由に移動できるようになります。

この章では、Cisco Primeネットワーク レジストラーサーバーで DNS アップデートを使用する方法と、Windows クライアント システムとの特別な関連性について説明します。

DNS 更新のプロセス

DNS 更新を構成するには、次の操作を行う必要があります。

1. 前方ゾーンまたは逆ゾーン、またはその両方に対して DNS 更新構成を作成します。DNS 更新設定の作成 （317 ページ）を参照してください。

2. 次の 2 つの方法のいずれかで、この DNS 更新の構成を使用します。
   - 名前付き、埋め込み、または既定の DHCP ポリシーで DNS 更新の構成を指定します。DHCP ポリシーの設定と適用 （204 ページ）を参照してください。
   - CISCO Prime Network レジストラー DHCP サーバーまたはフェールオーバー ペアと DNS サーバーまたは高可用性(HA)ペア間の単一 DNS アップデート 関係を自動設定する DNS アップデート マップを定義します。DNS 更新マップで更新の構成を指定します。DNS 更新マップの作成 （326 ページ）を参照してください。
3. 必要に応じて、DNS 更新のアクセス制御リスト (ACL) またはトランザクション署名 (TSIG) を定義します。アクセス コントロール リストとトランザクション セキュリティの設定 （308 ページ）を参照してください。
4. 必要に応じて、これらの ACL または TSIG に基づいて 1 つ以上の DNS 更新ポリシーを作成し、ゾーンに適用します。DNS 更新ポリシーの設定 （320 ページ）を参照してください。
5. 必要に応じて、DHCPv4 の TXT RR から DHCID RR に移行するように DNS 更新を構成します。DHCPv4 の DHCID RR への移行 （329 ページ）を参照してください。
6. 必要に応じて、Windows クライアントの DNS 更新構成を調整します。たとえば、デュアル ゾーン更新の場合などです。Windows クライアントの DNS 更新の構成 （331 ページ）を参照してください。
7. ホスト名を提供するか、Cisco Prime ネットワーク レジストラーがそれらを生成するように要求するように DHCP クライアントを設定します。
8. 必要に応じて、編集モードに基づいて DHCP サーバーと DNS サーバーを再ロードします。

特殊な DNS 更新に関する考慮事項

DNS 更新を構成する際には、次の 2 つの問題を考慮してください。

- セキュリティ上の理由から、Cisco Prime Network レジストラー DNS 更新プロセスでは、管理者が DNS データベースに手動で入力した名前は変更または削除されません。
- 大規模な展開で DNS 更新を有効にし、HA DNS を使用していない場合 (「高可用性 DNS ペアの展開」のCisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイド章を参照) は、プライマリ DNS サーバーと DHCP サーバーを複数のクラスターに分割します。DNS 更新は、サーバーに追加の負荷を生成します。

DHCPv6 の DNS 更新プログラム

Cisco プライムネットワーク レジストラーは現在、IPv4 および IPv6 経由の DHCPv6 DNS アップデートをサポートしています。DHCPv6 の場合、DNS 更新は非一時的なステートフル アドレスと委任されたプレフィックスに適用されます。

非一時ステートフル アドレスの DNS 更新

DHCPv6 の DNS 更新には、リース用の AAAA および PTR RR のマッピングが含まれます。Cisco Prime Network レジストラーでは、サーバーまたはエクステンションを使用した完全修飾ドメイン名と DHCPv6 クライアントFQDNオプション(39)がサポートされます。

Cisco Primeネットワーク レジストラーは RFC 4701、4703、および 4704 に準拠しているため、DHCID リソース レコード(RR)をサポートします。すべての RFC-4703 準拠のアップデータは、DHCID R を生成し、クライアント識別子 (DUID) と FQDN (RFC 4701 に従う) のハッシュであるデータを生成できます。ただし、更新ポリシー ルールで AAAA および DHCID の R を使用できます。

DHCPv6 の DNS 更新処理は、DHCPv4 の場合と似ていますが、1 つの FQDN が複数のリースを持つことができる点を除いて、1 つのクライアントに対して複数の AAAA および PTR R が発生します。複数の AAAA R は、同じ名前または異なる名前にすることができます。ただし、PTR の R は、リース アドレスに基づいて常に異なる名前で指定されます。RFC-4703 準拠のアップデータは、複数のクライアント間の競合を回避するために DHCID RR を使用します。

（注） DNS サーバーがダウンしていて、DHCP サーバーが DNS 更新を完了して DHCPv6 リースに追加された R を削除できない場合、リースは引き続き AVAILABLE 状態で存在します。同じクライアントのみがリースを再利用します。

委任されたプレフィックスの DNS 更新

委任されたプレフィックスの DNS 更新を有効にして、委任されたプレフィックス リースの AAAA および PTR マッピングを更新できます。ただし、この場合、委任されたプレフィックスの 0 アドレスの DNS のみが更新されます。たとえば、2001:db8:3333:3333:/64 のプレフィックスが委任されている場合、2001:db8:3333:3333::0 の PTR および/または AAAA のみが委任されます。は DNS で更新されます。この機能は、委任されたプレフィックスに対して DNS 委任を行う手段を提供しません。

委任されたプレフィックスの更新は、DNS 更新構成でプレフィックス委任更新属性が有効になっている場合にのみ有効になります。この属性はデフォルトでは無効になっています。委任されたプレフィックスの更新は、アドレス更新とは異なるゾーンに発生する可能性が高いため、新しいDNS更新構成を作成して、対応するプレフィックスに関連付ける必要があります。

標準の名前生成規則が適用されるため、ヒントを含む FQDN オプションを含むクライアントは、結果の名前に影響を与える可能性があります(構成で許可されている場合)。クライアントは、FQDNオプションを要求した場合、プレフィックスの委任の更新に使用される名前を返されることはありません。

（注） この機能を使用する場合は、両方のフェールオーバー パートナーがこの機能をサポートするバージョンを実行していることを確認する必要があります。それ以外の場合、更新はアップグレードされたサーバーによってサービスを提供された場合にのみ実行されます。したがって、両方のパートナーがアップグレードされるまで、この機能を有効にしないでください。

関連項目

DHCPv6 のアップグレードに関する考慮事項 （306 ページ）
DHCPv4 と DHCPv6 での合成名の生成 （306 ページ）
DNS 更新のための逆引きゾーンの決定 （307 ページ）
Client FQDN の使用 （308 ページ）

DHCPv6 のアップグレードに関する考慮事項

Cisco Prime Network レジストラーの前に設定された、DHCPv6 処理用の DNSDHCPv6 ポリシー階層 （202 ページ）更新オブジェクトを参照するポリシーを使用する場合(を参照)、サーバーは、指定された DNS サーバーに対する DNS 更新のキューイングを開始します。これは、DNS 更新が DHCPv6 リースに対して自動的に (および予期せず) 開始する可能性があることを意味します。

注意 Cisco Prime Network レジストラーまたはその他の DNS サーバーの以前のバージョンを使用する場合、最近の DHCID RR 標準の変更により、ゾーン転送および DNS 更新の相互運用性の問題が発生する可能性があります。DHCPv6 DNS 更新をサポートするために、DNS サーバーをアップグレードする必要がある場合があります。

DHCPv4 と DHCPv6 での合成名の生成

クライアントがホスト名を指定しない場合、DHCPv4 および DHCPv6 には合成名生成プログラムが含まれます。DNS 更新構成の v6 合成名前生成属性を使用すると、次の内容に基づいて生成された名前を合成名のステムに追加できます。

- クライアント DHCP 一意識別子 (DUID) 値 (プリセット値) のハッシュ。
- 未加工のクライアント DUID 値 (区切り記号のない 16 進数のストリング)。
- CableLabs ケーブルラボ-17オプションdevice-idサブオプション値(区切り文字のない 16 進数文字列、または見つからない場合はクライアント DUID のハッシュ)。
- CableLabs ケーブルラボ-17オプションcm-mac-addressサブオプション値(区切り記号のない 16 進数の文字列として、または見つからない場合はクライアント DUID のハッシュ)。

注意 ドメインがインターネットからアクセス可能な場合、一部の生成方法によってプライバシーの問題が発生する可能性があります。

DNS 更新構成のv4 合成名前生成属性では、次の内容に基づいて生成された名前を合成名のステムに追加できます。

- address:クライアントの v4 アドレスを識別します。
- **クライアント** ID:要求で DHCPv4 クライアントによって指定されたクライアント ID または DUID (オプション 61)。
- hashed-client-id—SHA-256 ハッシュの右部分 64 ビットで形成された 13 文字のベース 32 でエンコードされた文字列である、ハッシュ化されたクライアント ID に、前方ゾーン名が付加されます。

合成DNS 更新設定の作成 （317 ページ）名の生成を使用して DNS 更新構成を作成する方法については、「」を参照してください。

CLI では、この設定の例を次に示します。

```
nrcmd> dhcp-dns-update example-update-config set v6-synthetic-name-generator=hashed-duid

nrcmd> dhcp-dns-update example-update-config set v4-synthetic-name-generator=client-id
```

DNS 更新のための逆引きゾーンの決定

DNS 更新構成では、指定された逆ゾーン プレフィックス長属性のプレフィックス長の値を使用して、ip6.arpa ドメインの逆ゾーンを生成します。ip6.arpa ドメインを使用して合成できるため、完全なリバースゾーンを指定する必要はありません。逆引き DNS 更新の構成に対してDNS 更新設定の作成 （317 ページ）この属性を設定します(「」を参照してください)。逆引きゾーンプレフィックス長に関する規則を次に示します。

- ip6.arpa ゾーンは 4 ビット境界上にあるため、値には 4 の倍数を使用します。4 の倍数でない場合、値は 4 の次の倍数に切り上げられます。
- 最大値は 124 で、128 を指定すると、ホスト名が含まれる可能性のないゾーン名が作成されます。
- 値 0 はゾーン名に使用されるビットが一切使用されないので、ip6.arpa が使用されます。
- DNS 更新構成から値を省略すると、サーバーはプレフィックスの値を使用するか、最後の手段としてプレフィックスのアドレス値から取得されるプレフィックス長を使用します (「」をプレフィックスとリンクの設定 （161 ページ）参照)。

逆ゾーン名を合成するには、DHCP サーバーに対して、ゾーンの逆引きのシンセを有効にしておく必要があります。したがって、逆ゾーン名が DHCPv6 に対して合成される順序は次のようになります。

1. 逆引き DNS 更新の構成で完全な逆ゾーン名を使用します。
2. 逆引き DNS 更新構成では、逆ゾーン プレフィックス長からの ip6.arpa ゾーンに基づいて設定します。

3. プレフィックス定義の逆ゾーンプレフィックス長から ip6.arpa ゾーンに基づいて設定します。
4. プレフィックス定義のアドレスのプレフィックス長から ip6.arpa ゾーンに基づいて設定します。

CLI では、リバース ゾーン プレフィックス長を設定する例を次に示します。

```
nrcmd> dhcp-dns-update example-update-config set reverse-zone-prefix-length=32
```

Web UI でプレフィックスの逆引きゾーンを作成するには、プレフィックスの一覧/追加ページ Create Reverse Zoneに各プレフィックスのボタンが含まれています。（プレフィックスの作成と編集 （161 ページ） を参照）。

CLI では、プレフィックスprefixのリバース–rangeゾーンを作成する名前createReverseZone[ ] コマンドも提供します (アドレスまたは範囲の値から)。prefix 名前deleteReverseZone[ ]–rangeを使用して、逆引きゾーンを削除します。

逆ゾーンを直接構成するときにサブネットまたはプレフィックスの値を入力して、DHCPv4 サブネットまたは DHCPv6 プレフィックスからリバース ゾーンを作成することもできます。詳細については、以下の「プライマリ リバース ゾーンの構成」を Cisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイド参照してください。

Client FQDN の使用

既存の DHCP サーバーの使用クライアント fqdn属性は、要求の DHCPv6 クライアント FQDN オプションにサーバーが注意を払うかどうかを制御します。クライアントに複数の名前が存在する場合に、サーバーが返す名前を決定するために使用する規則は、次の優先順位です。

1. クライアントを使用するサーバー FQDN は、(DNS 内に存在すると見なされない場合でも) リースに使用されている場合に、FQDN を要求しました。
2. DNS 内に最も長い有効期間を持つ FQDN が有効であると見なされます。
3. DNS 内にまだない有効期間が最長の FQDN。

アクセスコントロールリストとトランザクションセキュリティの設定

ACL は権限リストですが、トランザクション・シグニチャー (TSIG) は認証メカニズムです。

- ACL を使用すると、サーバーはパケットに定義された要求またはアクションを許可または禁止できます。
- TSIG は、DNS メッセージが信頼された送信元から送信され、改ざんされないようにします。

セキュリティで保護する DNS クエリ、更新、またはゾーン転送ごとに、アクセス許可を制御する ACL を設定する必要があります。TSIG 処理は、TSIG 情報を含むメッセージに対してのみ実行されます。この情報を含まない、またはこの情報が取り除かれるメッセージは、認証プロセスをバイパスします。

完全に安全なソリューションの場合、メッセージは同じ認証キーによって承認される必要があります。たとえば、DHCP サーバーが DNS アップデートに TSIG を使用するように設定されており、更新するゾーンの ACL に同じ TSIG キーが含まれている場合、TSIG 情報を含まないパケットは認証ステップに失敗します。これにより、更新トランザクションがセキュリティで保護され、ゾーンの変更を行う前にメッセージが認証され、承認されます。

ACL と TSIG は、サーバーまたはゾーンの DNS 更新ポリシーを設定する役割をDNS 更新ポリシーの設定 （320 ページ）果たします。

関連項目

DNS キャッシュ サーバーまたはゾーンでの ACL の割り当て （309 ページ）

ACL のゾーンの設定 （310 ページ）

トランザクションのセキュリティ （311 ページ）

DNS キャッシュ サーバーまたはゾーンでの ACL の割り当て

DNS キャッシュ サーバーまたはゾーン レベルで ACL を割り当てます。ACL には、次の 1 つ以上の要素を含めることができます。

- IP - ドット区切り 10address進表記法たとえば、192.168.1.2 とします。
- Network - ドット 10 進表記とaddressスラッシュ表記。たとえば、192.168.0.0/24 などです。この例では、そのネットワーク上のホストのみが DNS サーバーを更新できます。
- Another - 事前定義するACL必要があります。埋め込みリレーションシップを削除するまでは、別の ACL に埋め込まれている ACL を削除できません。その ACL へのすべての参照が削除されるまで、ACL を削除しないでください。
- Transaction- Signature値は、キーワードの後にシークレット値が続く形式の値でなければなりません。 (TSIG) key key key スペース文字を格納するには、リスト全体を二重引用符で囲む必要があります。TSIG キーについては、トランザクションのセキュリティ （311 ページ）を参照してください。

各 ACL に一意の名前を割り当てます。ただし、次の ACL 名には特別な意味があり、通常の ACL 名には使用できません。

- any—誰でも特定のアクションを実行できます
- none-誰も特定のアクションを実行できません
- localhost- ローカル・ホスト・アドレスは、特定のアクションを実行できます。
- localnets- ローカル ネットワークは、特定のアクションを実行できます。

次の点に注意してください。

- ACL が設定されていない場合はany、この値が想定されます。
- ACL が設定されている場合、少なくとも 1 つの句でトラフィックを許可する必要があります。

- 否定演算子 (!) は、前のオブジェクトのトラフィックを禁止しますが、明示的に指定しない限り、本質的に他のトラフィックを許可しません。たとえば、IP アドレス 192.168.50.0 のトラフィックのみを禁止するには、を使用!192.168.50.0, anyします。

ローカル アドバンスド Web UI

[**デザイン**] メニューのACLs[**セキュリティ**] サブメニューの下で [リスト/ アクセスコントロールリストの追加]ページを開きます。[ACL]ペインの[ACL **の追加**]アイコンをクリックし、ACL 名と一致リストを入力して、[ACLの追加] をクリックします。key 値ペアは引用符で囲んではいりません。地域レベルでは、レプリカ ACL をプルしたり、ローカル クラスタに ACL をプッシュしたりできます。ACL を再利用することもできます。

CLI コマンド

名前aclと 1 つ以上の ACL 要素を受け取る名前create match-listを使用します。ACL リストはカンマで区切られ、スペース文字がある場合は二重引用符で囲まれます。CLI はプル/プッシュ機能を提供しません。

たとえば、次のコマンドは 3 つの ACL を作成します。1 つ目は値を持つキーで、2 つ目はネットワーク用で、3 つ目は最初の ACL を指します。値の前に感嘆符!() を含めると、その値を否定するので、一連の値で除外することができます。

```
nrcmd> acl sec-acl create "key h-a.h-b.example.com."

nrcmd> acl dyn-update-acl create "!192.168.2.13,192.168.2.0/24"

nrcmd> acl main-acl create sec-acl
```

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| acl**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| acl**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リスト [-レポートのみ | acl -**レポート**]

ACL のゾーンの設定

DNS サーバーまたはゾーンの ACL を構成するには、DNS 更新ポリシーを設定し、ゾーンに対してこの更新ポリシーDNS 更新ポリシーの設定 （320 ページ）を定義します (「」を参照)。

トランザクションのセキュリティ

トランザクション署名 (TSIG) の R を使用すると、DNS サーバーは、受信した各メッセージを、TSIG を含む認証を行います。サーバー間の通信は暗号化されませんが、認証されるため、データの信頼性とパケットの送信元を検証できます。

DNS アップデートに TSIG を使用するように Cisco Prime Network レジストラー DHCP サーバーを設定すると、サーバーはメッセージに TSIG RR を付加します。TSIG レコードの一部は、メッセージ認証コードです。

DNS サーバーは、メッセージを受信すると TSIG レコードを検索します。見つかった場合は、まず、そのキー名が認識されるキーの 1 つであることを確認します。その後、更新プログラムのタイム スタンプが妥当であることを確認します (トラフィックリプレイ攻撃との戦いを支援するため)。最後に、サーバーはパケットで送信されたキー共有シークレットを調べ、独自の認証コードを計算します。結果として計算された認証コードがパケットに含まれる認証コードと一致する場合、内容は本物であると見なされます。

関連項目

TSIG キーの作成 （311 ページ）

キーの生成 （312 ページ）

キーの管理に関する考慮事項 （313 ページ）

サポート TSIG 属性の追加 （314 ページ）

TSIG キーの作成

ローカル アドバンスド Web UI

[**デザイン**] メニューのKeys[**セキュリティ**] サブメニューの下で [暗号化キーの一覧/追加] ページを開きます。

アルゴリズム、セキュリティタイプ、時間スキュー、キー ID、およびシークレットの各値の説明については、表 33 : cnr_keygen ユーティリティのオプション を参照してください。キーの管理に関する考慮事項 （313 ページ） も参照してください。

TSIG キーを編集するには、[暗号化キーの一覧/追加] ページでキー名をクリックし、[暗号化キーの編集] ページを開きます。

地域レベルでは、レプリカ キーをプルしたり、キーをローカル クラスターにプッシュしたりできます。

CLI コマンド

key 名前シークレットを create使用する: キーの名前 (ドメイン名形式、たとえば、hosta-hostb-example.com など) と、共有シークレットの最小値を base-64 でエンコードされた文

字列として指定します (省略可能な time skew 属性の説明については表 33 : cnr_keygen ユーティリティのオプション を参照してください)。CLI の例は次のようになります。

```
nrcmd> key hosta-hostb-example.com.create secret-string
```

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。

- **キー**<名前|**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**]
- **キー**<名前|**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**]
- **キー**名**再利用**クラスタリスト[-**レポートのみ**|-**レポート**]

キーの生成

TSIG キーを生成するには、Cisco Prime Network レジストラー cnr_keygenユーティリティを使用して、追加するか、またはimport keysを使用してインポートすることをお勧めします。

DOScnr_keygenLinux シェルからキー生成ユーティリティを実行します。このユーティリティは、インストールパス/usrbin ディレクトリにあります。

使用例を次に示します。

```
> /opt/nwreg2/local/usrbin/cnr_keygen -n a.b.example.com. -a hmac-md5 -t TSIG -b 16
-s 300

key "a.b.example.com." {
algorithm hmac-md5;
secret "xGVCsFZ0/6e0N97HGF50eg==";
# cnr-time-skew 300;
# cnr-security-type TSIG;
};
```

キー名だけが必要です。オプションを次の表に示します。

表 *33 : cnr_keygen* ユーティリティのオプション

オプション	説明
–a hmac-md5	アルゴリズム。これはオプションです。現在、hmac-md5 のみがサポートされています。
–bシークレットサイズ	シークレットのバイトサイズ。これはオプションです。プリセット値は 16 バイトです。有効な範囲は 1 から 64 バイトです。
–sタイムスキュー	キーの時間スキュー (秒単位)。これは、このキーで署名されたパケットとローカルシステム時刻のタイムスタンプの最大差です。これはオプションです。プリセット値は 5 分です。範囲は 1 秒から 1 時間です。

オプション	説明
–n name	キー名。必須。最大長は 255 バイトです。
–t TSIG	使用されるセキュリティの種類。これはオプションです。現在、TSIG のみがサポートされています。
–h	[ヘルプ（Help）]。これはオプションです。ユーティリティの構文とオプションが表示されます。
–v	[バージョン]。これはオプションです。ユーティリティのバージョンが表示されます。

結果のシークレットは、ランダムな文字列として base64 エンコードされます。

コマンド ラインの最後で右矢印 (>) または二重右矢印 (>>) を使用する場合は、出力をファイルにリダイレクトすることもできます。> は指定されたファイルを書き込むか、または上書きし、>> は既存のファイルに追加します。次に例を示します。

```
> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com > keyfile.txt

> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com >> addtokeyfile.txt
```

その後、CLI を使用してキー ファイルを Cisco Prime Network レジストラーにインポートし、ファイル内のキーを生成できます。キーのインポートでは、インポート ファイルで検出された数だけキーを生成できます。ファイルへのパスは完全修飾パスにする必要があります。次に例を示します。

```
nrcmd> import keys keydir/keyfile.txt
```

キーの管理に関する考慮事項

独自のキーを生成する場合は、base64 エンコード文字列として入力する必要があります (base64 エンコードの詳細については RFC 4648 を参照してください)。これは、許可される文字は base64 のアルファベット文字と、埋め込み文字としての等号 (=) だけであることを意味します。base64 エンコードされていない文字列を入力すると、エラー メッセージが表示されます。

次に、他の推奨事項をいくつか示します。

- バッチ コマンドを使用してキーを追加または変更しないでください。
- 共有シークレットを頻繁に変更する。2ヶ月ごとにお勧めします。Cisco プライムネットワーク レジストラーでは、明示的にはこれを適用しないことに注意してください。
- 共有秘密の長さは、キー付きメッセージ ダイジェスト (HMAC-MD5 が 16 バイト) の長さ以上にする必要があります。Cisco Prime Network レジストラーでは、明示的に強制するものではなく、共有シークレットが有効な base64 でエンコードされた文字列であることを確認するだけですが、RFC 2845 で推奨されているポリシーです。

サポート TSIG 属性の追加

DNS 更新の構成に対して TSIGDNS 更新設定の作成 （317 ページ）サポートを追加するには(を参照) 次の属性を設定します。

- server-key
- backup-server-key

TSIG で GSS-TSIG セキュリティ アルゴリズムを使用するには、以下の属性を有効にします。

- 使用-gss-tsig

GSS-TSIG

RFC 3645 では、汎用セキュリティ サービス (GSS) の安全なキー交換を許可する TSIG の拡張を提案し、すべての GSS クライアントにキーを手動で配布する必要がなくなります。RFC 2743 で規定されている汎用セキュリティ サービス アプリケーション プログラム インターフェイス（GSS API）に基づく TSIG で使用するアルゴリズムを定義します。

GSS-TSIG は、Kerberos セキュリティ メカニズムを利用して、セキュア DDNS 更新とセキュア ゾーン転送を提供します。

クライアントとサーバーは、GSS API 呼び出しを使用して、認証、整合性、および機密性に関する制限された有効期間のセキュリティ コンテキストを確立します。セキュリティ コンテキストを確立するには、ネゴシエーションが完了するまで、クライアントとサーバーの間で不透明なトークンを渡す必要があります。TKEY リソース レコード [RFC 2930] は、クライアントとサーバー間でトークンを転送する手段として使用されます。セキュリティ コンテキストが確立されると、GSS API 呼び出しを使用して署名を生成および検証するために使用されます。これらの署名は、[RFC 2845] で説明されているように、クライアントとサーバーの間で送信される DNS メッセージで交換される TSIG レコードの一部として、クライアントとサーバーによって交換されます。

このプロトコルを使用する前に、クライアントとサーバーは Kerberos サーバーでローカルに認証される必要があります。一般に、初期TGT(チケットを取得するチケット)チケットは、システムログオンを通じてキャッシュで利用可能であるか、kinitのようなユーティリティを使用して取得されます。DHCP/DNS クライアントは、プリンシパル名 (DNS/ホスト名) を使用してサービス チケット用の Kerberos サーバーを要求します。クライアントは、DNS サーバーと安全に対話する際に認証を証明するサービス チケットを提供します。サービス チケットは、同じサービス キーを使用してアプリケーション サーバーのみが暗号化解除できるサービス キーを使用して、Kerberos サーバーによって暗号化されます。

詳細については、DHCP サーバーGSS-TSIG の設定 （346 ページ）と DNS サーバーで必要な構成のを参照してください。

（注） デフォルトでは、Cisco プライムネットワーク レジストラーは HMAC-MD5 ベースのセキュア TSIGアップデートをサポートします。GSSベースのセキュア更新を有効にするには、ユーザーはtsig 処理属性で none オプションを選択して、DNS サーバーで HMAC-MD5 設定をすべて無効にする必要があります。

DHCP サーバーとセカンダリ DNS サーバーの構成

KDC サーバー情報を /etc/krb5.conf で構成します。KDC から最初のチケットを取得するには、kinit ユーティリティを使用します。

（注） サーバー間の通信に Kerberos サーバーを使用する場合は、/etc/krb5.conf の最新の暗号化アルゴリズムを使用することをお勧めします。

DHCP サーバーとセカンダリ DNS サーバーの構成のトラブルシューティング

- 初期資格情報の取得中に発生する可能性のあるクライアント関連のエラー:
- クロックスキューエラー - Kerberos クライアントとサーバーを確認し、ntp と同期しない場合は時間内に同期します。
- KDC に到達できない - AD ホスト名が解決可能であることを確認します。
- kinit - 初期資格情報を取得中に Kerberos データベースにクライアントが見つかりません - ユーザーが AD に存在するかどうかを確認します。
- kinit - 初期資格情報を取得中に領域「DOMAIN.com」のKDCのサーバーを解決できません - REALMがADに存在するかどうかを確認します。
- kinit - 初期資格情報を取得中に事前認証に失敗しました - チケットを取得するために入力されたパスワードがADのユーザーに関連付けられたパスワードと同じかどうかを確認します。

GSS-TSIG 設定の作成

DNS/DHCP は、キー管理用の非永続的テーブルを維持します。

（注） DHCP および DNS サーバーで使用される既定の TKEY 管理値を変更するオプションがあります。GSS- TSIG 設定を作成し、DHCP/DNS サーバー ページで参照を提供する必要があります。

ローカルおよびリージョン Web UI

[設計] メニューから、[セキュリティ] サブメニューの下の[GSS-TSIG]を選択して、[GSS-TSIG 設定の一覧/追加] ページを開きます。左側のGSS-TSIG ペインで[GSS-TSIGの追加]アイコンをクリックします。名前を入力し、[GSS-TSIG 設定の追加] をクリックします。

GSS-TSIG 属性

- tkey-max-exchanges - 無限ループを防ぐために RFC 3645 からの勧告に従って、DNS サーバーは特定のキーをネゴシエートしようとして、TKEY 交換の最大数 (つまり、特定のクライアントから受け取った数の TKEY クエリ) を課すものとします。この属性は、この制限を指定する必要があります。TKEYテーブル レコードは、交換カウントを保持します。キー ネゴシエーション中に交換カウントが tkey-max 交換を超えた場合、DNS サーバーはキー ネゴシエーションを中止します。
- tkey-テーブル-最大サイズ- この属性は TKEY テーブルのサイズを制限します。
- tkey テーブル消去インターバル- TKEY テーブルから期限切れキーを削除する時間間隔。
- tkey-session-time - ユーザーが構成可能なキーの最大有効期間を指定します。キーの有効期間は、最初のキー ネゴシエーション中およびこの属性を使用して取得した Kerberos サーバーの有効期限時間によって制御されます。0に設定すると、この属性は無効になり、キーの有効期間は、指定された有効期限が指定されたKerberosによってのみ制御されます。この属性が値 > 0 で構成されている場合、Kerberos の有効期限の最小値とこの値がキーの最大有効期間として使用されます。

GSS-TSIG 設定を編集するには、[GSS-TSIG 設定の一覧/追加] ページで名前をクリックし、[GSS-TSIG 設定の編集] ページを開きます。

地域レベルでは、GSS-TSIG 設定をローカル クラスターにプルまたはプッシュすることもできます。

CLI コマンド

gss-tsig名の作成[属性=値..] を使用します。GSS-TSIG 設定オブジェクトの名前を指定します。次に例を示します。

```
nrcmd> gss-tsig gss create tkey-max-exchanges=6 tkey-table-max-size=500
tkey-table-purge-interval=90
```

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。

- gss-tsig <名前|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]
- gss-tsig <名前|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]
- gss-tsig 名再利用クラスタリスト[-レポートのみ|-レポート]

DNS 更新設定の作成

DNS 更新の構成では、DNS サーバーまたは HA DNS サーバーのペアに対する DNS 更新用の DHCP サーバー フレームワークを定義します。これは、前方または逆ゾーンの DNS 更新 (またはその両方) を生成するかどうかを決定します。オプションで、トランザクションの TSIG キー、自動生成されたホスト名のスタイルを制御する属性、および更新する特定の前方または逆ゾーンを設定します。一意のサーバーリレーションシップごとに DNS 更新の構成を指定する必要があります。

たとえば、DHCP サーバーからのすべての更新が単一の DNS サーバーに送信される場合、サーバーの既定のポリシーで設定された単一の DNS 更新構成を作成できます。クライアント クラスのクライアントの各グループを対応する転送ゾーンに割り当てるには、より具体的なクライアント クラス ポリシーで、それぞれのクライアントの前方ゾーン名を設定します。

Cisco Prime Network Registrar 11.0 以降、より厳格なルールが DNS アップデート設定で指定する DNS サーバーに適用されます。DNS サーバーを複数のロールで使用するように設定できなくなります。つまり各サーバーは（アドレスに基づいて）スタンドアロン、HA メイン、または HA バックアップとしてのみ動作することができます。HA メインまたは HA バックアップは、単一の HA の関係でのみ存在できます。したがって、DNS サーバーを複数のロールで実行する必要がある場合は、ロールごとに個別の DNS サーバーのアドレスを使用する必要があります。

（注） DNS 更新設定で、複数のロールが DNS サーバーを使用していた場合、DHCP サーバーのリロード時にエラーが報告されます。報告されるエラーは、メッセージ 19696「DNS Update Configuration ‘name1' with server-config-type of server(s)-address conflicts with DNS Update Configuration 'name2' with server-config-type of server(s)-address DNS Update Configuration ‘name1' will be set to disable DNS updates and thus will not configure server(s)」です。

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 [**展開**] メニューのDNSUpdateConfigsDNSUpdatesサブメニューの下で [DNS 更新の一覧/ 追加] ページを開きます。

ステップ 2 [DNS 更新構成] ウィンドウの[DNS **更新構成の追加**] アイコンをクリックして、[DnsUpdateConfig の追加] ダイアログ ボックスを開きます。

ステップ 3 [名前属性] フィールドに、更新設定の名前を入力します。

ステップ 4 Add DnsUpdateConfig をクリックして、DNS 更新設定を追加します。

ステップ 5 更新構成の名前を選択して、[DNS 更新の構成の編集] ページを開きます。

ステップ 6 [更新設定] セクションで、適切な動的 DNS設定をクリックします。

- update-none- 前方ゾーンまたは逆方向ゾーンを更新しません。

- update-all- 前方ゾーンと逆方向のゾーンを更新します (デフォルト値)。
- update-fwd-only- 転送ゾーンのみを更新します。
- update-reverse-only- 逆ゾーンのみを更新します。

ステップ 7 更新設定ブロックの下で、適切なDNS クライアント ID設定をクリックします。

- txt—サーバーは DHCPv4 DNS 更新に TXT RR を使用し、DHCPv6 DNS アップデートには DHCID RR を使用します。
- dhcid—サーバーは DHCPv4 と DHCPv6 の両方の DNS 更新に DHCID RR を使用します。
- **移行から dhcid へ**—サーバーは、DNS サーバーの新しいレコードに対して DHCID RR を使用し、次の DNS 更新が行われたときに既存のエントリを更新して DHCID RR を使用します。
- regress-to-txt—サーバーは、DNS サーバーの新しいエントリに TXT RR を使用し、次の DNS 更新が行われるときに既存のエントリをアップグレードして TXT RR を使用します。

（注） DNS クライアント ID属性は、DHCP サーバー全体の設定の一部としても使用でき、個々の DNS 更新構成の属性が構成されていない場合に考慮されます。

ステップ 8 他の属性を適切に設定します。

- 必要に応じて、合成名を有効にし、合成名ステム値を設定します。

  クライアントがホスト名を提供しない場合は、合成名前-stemを使用して、デフォルトのホスト名のステムを使用するように設定できます。DHCPv4 の場合、合成名属性を有効にして、合成名ステムの値に基づいて DHCP サーバーがクライアントの一意の名前を合成するようにトリガーします。結果の名前は、名前の stem にハイフン付き IP アドレスが付加された名前になります。たとえば、example.comドメインのアドレス 192.168.50.1 に合成名のステムhostを指定し、合成名属性を有効にすると、結果のホスト名はhost-192-168-50-1.example.comされます。合成名のステムのプリセット値はdhcpです。

  合成名ステムは次の必要があります。

  - 末尾のドットを含まない相対名にします。
  - 英数字の値とハイフン–( ) のみを含めます。スペース文字とアンダースコアはハイフンになり、他の文字は削除されます。
  - 先頭または末尾のハイフンを含めずに使用します。
  - DNS ホスト名は、ラベルあたり 63 文字以下、全体で 255 文字以内にしてください。このアルゴリズムは、構成された転送ゾーン名を使用して、ホスト名に使用できる文字の数を判別し、必要に応じて最後のラベルの末尾を切り捨てます。

  DHCPv6 については、DHCPv4 と DHCPv6 での合成名の生成 （306 ページ）を参照してください。

- 転送ゾーンを更新する場合は、転送ゾーン名を転送ゾーンに設定します。ポリシーの転送ゾーン名は、DNS 更新構成の設定よりも優先されることに注意してください。

DHCPv6 の場合、サーバーは、ポリシー階層で前方ゾーン名の値を検索するときに、クライアントおよびクライアント クラスのポリシーを無視します。前方ゾーン名の検索は、プレフィックス埋め込みポリシーで始まります。

- DHCPv4 の場合は、逆ゾーン名を、PTR および TXT レコードで更新する逆 (.addr.arpa) ゾーンに設定します。設定されていない状態で、DHCP サーバーの逆方向ゾーン属性が有効になっている場合、サーバーは、各リースのアドレス、スコープ サブネット番号、および DNS 更新の構成 (またはスコープ)の DNS ホスト バイト属性値に基づいて逆ゾーン名を合成します。

  dns-host-bytes値は、逆ゾーン名のホストとゾーンの部分の間の分割を制御します。この値は、ホスト名に使用するリース IP アドレスからのバイト数を設定します。残りのバイトは、in-addr.arpa ゾーン名に使用されます。値 1 は、ドメインのホスト部分に 1 バイトのみを使用し、残りの 3 バイトをドメイン名から使用する (逆)。値 4 は、アドレスのホスト部分に 4 バイトすべてを使用し、ドメインの in-addr.arpa 部分のみを使用します。設定されていない場合、サーバーはスコープサブネットのサイズに基づいて適切な値を合成するか、逆ゾーン名が定義されている場合は、この名前からホストバイトを計算します。

  one-a-rr-per-dns-name は、名前ごとに 1 つまたは複数の A RR を許可するように、DHCPv4 DNS 更新を制御します。8.2 より前のバージョンの Cisco Prime Network レジストラーでは、サーバーが Mac アドレス ベースの識別子を使用しているため、名前ごとに A (名前とアドレス マッピング エントリ) のみがサポートされました。Cisco Prime Network レジストラー 8.2 でDUID サポートと DHCID RR が導入されると、マルチ接続クライアントには複数の A RR が存在します。

  DHCPv6 の場合は、DNS 更新のための逆引きゾーンの決定 （307 ページ） を参照してください。

- サーバー アドイン/サーバー ipv6addrを、転送ゾーン (逆ゾーンのみ更新する場合は逆ゾーン) のプライマリ DNS サーバーの IPv4/IPv6 アドレスに設定します。

TSIG キーを使用してすべての DNS 更新を処理する場合は、サーバー キーとバックアップ サーバー トランザクションのセキュリティ （311 ページ）キーを設定します (を参照)。

セキュリティで保護されたキー交換の汎用セキュリティ サービス (GSS) メソッドを使用している場合は、use-gss-tsigを true に設定します (を参照)。 GSS-TSIG の設定 （346 ページ）

- HA DNS が構成されている場合は、バックアップ サーバーの追加/バックアップ サーバー ipv6addrをバックアップ DNS サーバーの IPv4/IPv6 アドレスに設定します。
- 必要に応じて、update-dns-for-bootp（事前設定値は有効）を有効または無効にします。

ステップ 9 地域レベルでは、ローカル クラスターに更新の構成をプッシュしたり、[DNS 更新の一覧] ページまたは [DNS 更新の追加] ページでレプリカ データベースからそれらを取得したりすることもできます。

ステップ 10 Save をクリックします。

ステップ 11 ポリシーでこの DNS 更新の構成を指定するにはDHCP ポリシーの設定と適用 （204 ページ）、「」を参照してください。

CLI コマンド

dhcp-dns-update名前createを使用する [属性=値..] 次に例を示します。

```
dhcp-dns-update example-update-config create
```

dynamic-dns属性を適切な値 (更新なし、すべて更新、更新-fwd のみ、または更新 -逆のみ) に設定します。次に例を示します。

```
nrcmd> dhcp-dns-update example-update-config set dynamic-dns=update-all
```

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 |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 |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]
- dhcp-dns-update名はクラスタリストを再利用する [-レポートのみ|-レポート]

関連項目

DNS 更新のプロセス （303 ページ）

特殊な DNS 更新に関する考慮事項 （304 ページ）

DHCPv6 の DNS 更新プログラム （304 ページ）

DNS 更新ポリシーの設定

DNS 更新ポリシーは、更新の承認を RR レベルで管理するためのメカニズムを提供します。更新ポリシーを使用すると、RR の名前と種類だけでなく、ACL に基づくルールに基づいて DNS 更新を許可または拒否できます。ACL については、「」DNS キャッシュ サーバーまたはゾーンでの ACL の割り当て （309 ページ）を参照してください。

関連項目

Cisco プライムネットワーク レジストラーリリースとの互換性 （320 ページ）

ポリシーの作成と編集 （321 ページ）

更新ポリシーのルールの定義と適用 （321 ページ）

Cisco プライムネットワーク レジストラーリリースとの互換性

Cisco Prime Network レジストラーリリースでは、管理者が入力した静的 R を使用しましたが、DNS 更新は変更できませんでした。静的な R と動的な R の区別はなくなりました。ここで、

R を保護または保護解除としてマークできるようになりました (の「リソース レコード セットの保護」セクションを Cisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザー ガイド参照)。管理者が、R を作成または変更することで、R を保護するかどうかを指定できるようになりました。DNS 更新は、指定されたタイプの RR がセット内にまだ存在しない場合でも、保護された RR セットを変更できません。

（注） 以前のリリースでは、A、TXT、PTR、CNAME、および SRV レコードに対してのみ DNS 更新を許可しました。これは、保護されていない名前セット内の SOA レコードおよび NS レコード以外のすべてのレコードを更新できるように変更されました。以前のリリースとの互換性を維持するには、更新ポリシーを使用して RR 更新を制限します。

ポリシーの作成と編集

更新ポリシーの作成には、最初に名前の作成が含まれます。

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 [**デザイン**] メニューのUpdate Policies[**セキュリティ**] サブメニューの下で [DNS 更新ポリシーの一覧/追加] ページを開きます。このオプションは、サーバーが権限のあるサービスで構成されている場合に使用できます。

ステップ 2 [更新ポリシー] ウィンドウの [**更新ポリシーの追加**] アイコンをクリックして、[DNS 更新ポリシーの追加] ダイアログ ボックスを開きます。

ステップ 3 更新ポリシーの名前を入力します。

ステップ 4 [DNS更新ポリシーの追加（Add DNS Update Policy）] をクリックします。

ステップ 5 更新ポリシーのルールの定義と適用 （321 ページ）に進みます。

CLI コマンド

update-policy name create を使用します。次に例を示します。

```
nrcmd> update-policy policy1 create
```

更新ポリシーのルールの定義と適用

DNS 更新ポリシーは、ACL に基づいて特定の R の更新を許可または拒否するルールを定義する場合にのみ有効です。ルールが満たされない場合、デフォルトの (最後の暗黙的な) ルールは"deny any wildcard * *"、すべての更新を拒否する ( )。

関連項目

名前付き更新ポリシーのルールの定義 （322 ページ）

ゾーンへの更新ポリシーの適用 （325 ページ）

名前付き更新ポリシーのルールの定義

名前付き更新ポリシーのルールを定義するには、一連の Grant ステートメントと Deny ステートメントが必要です。

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 ポリシーの作成と編集 （321 ページ） の説明に従い更新ポリシーを作成するか、編集します。

ステップ 2 [DNS 更新ポリシーの一覧/追加] ページまたは [DNS 更新ポリシーの編集] ページで、次の手順を実行します。

a) [インデックス] フィールドにオプションの値を入力します。
b) [**許可**]を有効にしてルールを許可するか、[**拒否**] を有効にしてルールを拒否します。
c) [ACL リスト]フィールドにアクセス制御リストを入力します。
d) [**キーワード**] ドロップダウン リストからキーワードを選択します。
e) [値] フィールドにキーワードに基づいて値を入力します。これは、RR またはサブドメイン名、またはキーワードがwildcard使用されている場合は、ワイルドカードを含めることができます (下の表を参照してください)。

ネットワークが IPv4 から IPv6 アドレスへの移行を行うため、多くのネットワーク デバイスは IPv4 アドレスと IPv6 アドレスの両方を使用します。これらのデバイスは、同じホスト上の複数のインターフェイスを使用している場合や、異なるネットワークを使用している場合や、異なる DHCP バージョンを使用している場合があります。これらのデバイスは、DHCP サーバーに関して一貫して識別する必要があり、それに応じて DHCP サーバーは DNS サーバーを更新します。

Cisco プライム ネットワーク レジストラー 8.1 以前、DHCPv4 は TXT R を使用し、DHCPv6 は DHCID R を使用して DNS を更新します。クライアントが要求した名前の競合を避けるために、デュアルスタック クライアントは単一の前方 FQDN を使用できません。これらの競合は、主にクライアントが要求した名前に適用され、生成される名前には適用されません。これらの競合を避けるために、DHCPv4 と DHCPv6 の名前に異なるゾーンが使用されました。

Cisco プライムネットワーク レジストラー 8.2 以降では、DHCPv4 は TXT RR または DHCID RR を使用し、DHCPv6 は DNS アップデートに DHCID RR を使用します。DHCP サーバー全体の設定属性 dnsクライアント IDの既定値は txt であり、属性は個々の DNS 更新構成オブジェクトに対して構成されていません。DNS 更新は、次のいずれかの方法で設定できます。

- DHCPv4 の TXT RR と DHCPv6 の DHCID:この構成を有効にするには、dns クライアント ID を txt に設定します。サーバーは、DHCPv4 DNS 更新で TXT RR を使用し、DHCPv6 DNS 更新には DHCID RR を使用します。 この設定は、DHCPv4 で TXT RR の使用のみをサポートする Cisco Prime Network Registrar 8.1 以前のバージョンで下位互換性を得るために使用されます。この設定は、Cisco Prime Network レジストラー 8.1 以前のクラスタがゾーンに対する DNS 更新に関与している場合に使用する必要があります。

- DHCPv4 と DHCPv6 の両方の DHCID RR:この構成を有効にするには、dns クライアント ID を dhcid に設定します。サーバーは、DHCPv4 および DHCPv6 DNS 更新の両方に DHCID RR を使用しま

す。この設定は、デュアル スタック クライアントをサポートするために使用する必要があり、この構成をサポートするゾーンに対して DNS 更新を行うすべての DHCP サーバーが DHCID RR を使用するように構成されている場合にのみ使用できます。

- DHCID RR への移行:この構成を有効にするには、dns クライアント IDを dhcid への移行に設定します。強制DNS 更新属性を true に設定します。サーバーをリロードします。アップグレードする必要があるゾーンについては、dns クライアント ID属性を dhcid に設定し、サーバーで最長のリース時間が設定された後で、force-dns-update属性を以前の値に復元します。

（注） すべての DHCPv4 リソース・レコードが DHCID RR に更新されるまで、dhcid への移行属性を設定する必要があります。詳細については、DHCPv4 の DHCID RR への移行 （329 ページ）を参照してください。

- TXT RR への後退:この設定を有効にするには、dns クライアント IDをリグレッションから txt に設定します。強制DNS 更新属性を true に設定します。サーバーをリロードします。アップグレードする必要があるゾーンについては、dns クライアント ID属性を txt に設定し、サーバーで最長のリース時間が設定された後で、force-dns-update属性を以前の値に復元します。

表 *34* : 更新ポリシー ルールのワイルドカード値

ワイルドカード	説明
*	0 個以上の文字と一致します。たとえば、パターンexample*は、例で始まるすべての文字列に一example–致します。
?	1 つの文字のみと一致します。たとえば、パターンexample?.comは とexample1.comexample2.com一致しますexample.comが、 は一致しません。
/[/]	(エスケープされた) 角かっこ内の任意の文字に一致します。たとえば、/[abc/]などです。各角かっこはスラッシュ ( )/を使用してエスケープする必要があります。文字は範囲内に含めることができます。など、/[0–9/]と/[a–z/]。パターンにハイフンを含める場合は、ハイフンを最初の文字にします。たとえば、example/[–a–z/]などです。

f) 1 つ以上の RR タイプをカンマで区切って[RR タイプ]フィールド*に入力するか、「すべての RR」に使用します。否定された値は、感嘆符の接頭辞が付いた値で使用できます。たとえば、!PTRなどです。

g) Save をクリックします。

ステップ 3 地域レベルでは、ローカル クラスターに更新ポリシーをプッシュしたり、[DNS 更新ポリシーの一覧/追加]ページでレプリカ データベースからポリシーをプルすることもできます。

ステップ 4 更新ポリシーを編集するには、[リスト/DNS 更新ポリシーの追加]ページで更新ポリシーの名前をクリックし、[DNS 更新ポリシーの編集] ページを開き、Saveフィールドを変更して をクリックします。

CLI コマンド

更新ポリシーを作成または編集するポリシーの作成と編集 （321 ページ）(「」をupdate-policy参照) ルールをルールにして名前rulesaddルールを使用します。(ルールのワイルドカード値については、上の表を参照してください。次に例を示します。

```
nrcmd> update-policy policy1 rules add "grant 192.168.50.101 name host1 A,TXT" 0
```

ルールは引用符で囲まれます。例のルール構文を解析するには、次の手順を実行します。

- grant- サーバーが実行するアクションまたは. grantdeny
- 192.168.50.101— ACL (この場合は IP アドレス)。ACL は次のいずれかになります。
  - 名前: のDNS キャッシュ サーバーまたはゾーンでの ACL の割り当て （309 ページ）説明に従って、名前で作成された ACL。
  - 例のように IP アドレス。
  - マスクを含むネットワークアドレス。たとえば、192.168.50.0/24などです。
  - TSIG キー : トランザクション署名キー key=(フォームキー)トランザクションのセキュリティ （311 ページ）で 、(説明を参照)。
  - 予約語の 1 つ:

    any—任意の ACL

    none—ACL なし

    localhost：任意のローカル ホスト アドレス

    localnets：任意のローカル ネットワーク アドレス

ACL 値の前に感嘆符 ()!を付けて、ACL 値を否定できます。

- name- RR で実行するキーワード、またはチェックのタイプは、次のいずれかです。
  - name- RR の名前(名前の値を必要とする)
  - subdomainRR または RR のいずれか 1 つの RR を持つサブドメインの名前(名前またはサブドメインの値を必要とする)
  - wildcard— ワイルドカード値を使用した RR の名前 (上の表を参照)。
- host1— キーワードに基づく値(この場合は、host1 という名前のRR)。サブドメイン名を指定することも、キーワードがwildcard使用されている場合はワイルドカードを使用することもできます (上の表を参照)。
- A,TXTRR タイプ(それぞれカンマで区切られた)。これは、感嘆符 ( )!を前に付けて、各レコードの種類の値を否定する「リソース レコード」でCisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイド説明されている RR の種類の一覧にすることができます。
- この規則または割り当てられた規則が満たされない場合、デフォルトではすべての RR 更新が拒否されることに注意してください。

引用符の外側のルールの末尾に取り付け、インデックス番号、例では、.0です。インデックス番号は0から始まります。更新ポリシーに複数のルールがある場合、インデックスは、より低

い番号付きインデックスがリスト内で優先されるような特定の順序でルールを追加するのに役立ちます。ルールにインデックスが含まれていない場合は、リストの末尾に配置されます。したがって、ルールは、明示的に定義されているかどうかにかかわらず、常にインデックスを持っています。ルールを削除する必要がある場合に備えて、インデックス番号も指定します。

ルールを置き換えるにはupdate-policy 、name deleteを使用してから、更新ポリシーを再作成します。ルールを編集するには、update-policy 名前 rules remove インデックスを使用します(インデックスは明示的に定義されたインデックス番号またはシステム定義のインデックス番号です)、ルールを再作成します。前の例の 2 番目のルールを削除するには、次のように入力します。

```
nrcmd> update-policy policy1 rules remove 1
```

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。

- **更新ポリシー**<名前|**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**]
- **更新ポリシー**<名前|**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**]
- **更新ポリシー**名再**請求**クラスタリスト[-**レポートのみ**|-**レポート**]

ゾーンへの更新ポリシーの適用

更新ポリシーを作成した後、権限のあるサービスを使用してDNSサーバーを構成した場合は、更新ポリシーをゾーン (順方向および逆方向) またはゾーン テンプレートに適用できます。

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 [**デザイン**] メニューの [**認証**DNS]サブメニューの [**転送ゾーン**] を選択して、[転送ゾーンの一覧/追加] ページを開きます。

ステップ 2 ゾーン名をクリックして、[ゾーンの編集（Edit Zone）] ページを開きます。

ヒント また、ゾーン テンプレートの編集ページでゾーン テンプレート、プライマリ リバース ゾーンの編集ページでプライマリ リバース ゾーンに対してもこの機能を実行できます(の「ゾーンの管理」の章Cisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイドを参照してください)。

ステップ 3 [DNS 更新設定] セクションの [更新ポリシーリストの属性] フィールドに、1 つ以上の既存の名前付き更新ポリシーの名前または名前 (コンマ区切り) を入力します。

（注） サーバーは更新ポリシーリストを処理する前に、更新acl を処理します。

ステップ 4 [保存（Save）] をクリックします。

CLI コマンド

zone名前ポリシーの作成と編集 （321 ページ）を使用し、update-policy-list 属性とコンマ区切りの更新ポリシーの引用符付きリストを使用します。 set update-policy-list 次に例を示します。

```
nrcmd> zone example.com set update-policy-list="policy1,policy2"
```

DNS 更新マップの作成

DNS 更新マップを使用すると、更新の構成に基づいて、更新のプロパティが HA DNS サーバー ペアまたは DHCP フェールオーバー サーバー ペア間で同期されるように DNS 更新を構成しやすくなるので、冗長なデータエントリを減らすことができます。更新マップは、DNS ペアサービスのすべてのプライマリ ゾーン、または DHCP がサービスをペアにするすべてのスコープに適用されます。更新マップのポリシーを指定する必要があります。この機能を使用するには、管理者に DNS 管理または中央 DNS 管理ロールのサーバー管理サブロール、および dhcp 管理ロール (更新の構成用) が割り当てられている必要があります。

ローカルおよびリージョン Web UI

ステップ 1 メニューからDeploy[DNS UpdateMaps更新]サブメニューの下で選択し、[DNSアップデート マップの一覧/追加]ページを開きます。オプションは、サーバーが権限を持つサービスで設定されている場合に選択できます。

ステップ 2 [マップAddDNSUpdateのMap更新] ウィンドウのアイコンをクリックして、[DNS 更新マップの追加] ダイアログ ボックスを開きます。

ステップ 3 [名前（Name）] フィールドに更新マップ名を入力します。

ステップ 4 この設定に関連付けられた DNS サーバーまたは HA ペアを選択します。

ステップ 5 この構成に関連付けられている DHCP サーバーまたは DHCP フェールオーバー ペアを選択します。

ステップ 6 dns-configフィールドに、前のセクションの DNS 更新の構成を入力します。

ステップ 7 dhcp ポリシー セレクタ属性に対して、ポリシー選択の種類を設定します。次の選択項目があります。

- use-named-policy: dhcp 名前付きポリシー属性(プリセット値)に対して、名前付きポリシー セットを使用します。
- use-client-class-embedded-policy: dhcp-clientクラス属性に対して、クライアント クラス セットの組み込みポリシーを使用します。
- use-scope-embedded-policy- スコープの埋め込みポリシーを使用します。

ステップ 8 更新 ACL (を参照アクセス コントロール リストとトランザクション セキュリティの設定 （308 ページ）) またはDNS 更新DNS 更新ポリシーの設定 （320 ページ） ポリシー(を参照) を使用する場合は、dns-update-acl 属性またはDNS 更新ポリシー リスト属性を設定します。いずれの値も、コンマで区切られた 1 つ以上のアドレスにすることができます。dns 更新-aclは、dns 更新ポリシーリストよりも優先されます。

両方の値を省略すると、単純な更新の ACL が構築され、指定された DHCP サーバーまたはフェールオーバー ペアのみが更新を実行でき、dns-config属性に指定された更新構成で設定されたサーバー キー値も設定されます。

ステップ 9 Add DNS Update Map をクリックします。

ステップ 10 地域レベルでは、更新マップをローカル クラスターにプッシュするか、[DNS 更新マップの一覧/追加] ページのレプリカ データベースからプルできます。

CLI コマンド

名前、DHCPdns-update-mapサーバーと DNS サーバーのクラスタ (または DHCP フェールオーバーまたは HA DNS サーバーペア) と、名前dhcp-cluster dns-configを使用して更新マップを作成するときに DNS 更新の構成をcreate指定します。次に例を示します。

```
nrcmd> dns-update-map example-update-map create Example-cluster Boston-cluster
example-update-config
```

dhcpポリシー セレクタ属性値を、名前付きポリシー、use-client クラス埋め込みポリシー、または use スコープ埋め込みポリシーに設定します。名前付きポリシーの使用値を使用する場合は、dhcp 名前付きポリシー属性値も設定します。次に例を示します。

```
nrcmd> dns-update-map example-update-map set dhcp-policy-selector=use-named-policy

nrcmd> dns-update-map example-update-map set dhcp-named-policy=example-policy
```

地域クラスターに接続する場合は、dns-update-map 名**プッシュ**を使用できます [-report-only |-**レポート**]コマンド。

動的レコードの確認

Cisco プライムネットワーク レジストラー DHCP サーバーは、保留中のすべての DNS アップデート データをディスクに保存します。DHCP サーバーが DNS サーバーと通信できない場合は、定期的に通信の再確立をテストし、保留中のすべての更新を送信します。このテストは通常 40 秒ごとに行われます。

ローカルおよび地域 Web UI

[**デザイン**] メニューForward Zonesのサブメニューの下で選択し、[転送ゾーンのリスト/追加] ページを開きます。左側のペインで必要なゾーンを選択し、[ゾーンの編集] ページの [**リソース レコード**] タブをクリックします。

CLI コマンド

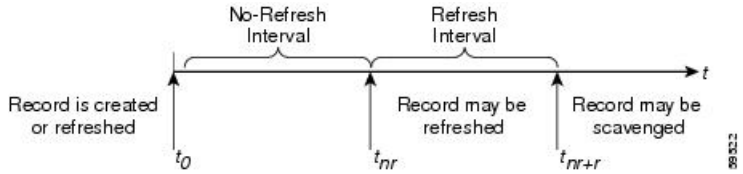
zone name listRR dns を使用します。

動的レコードのスカベンジング

DHCP リースを取得する Microsoft Windows DNS クライアントは、アドレス (A) レコードを DNS サーバーに直接更新 (更新) できます。これらのクライアントの多くは、永続的に接続されていないモバイル ラップトップであるため、一部の A レコードは時間の経過とともに古くなっている可能性があります。Windows DNS サーバーは、これらのプライマリ ゾーン レコードを定期的に清掃および削除します。Cisco Prime Network レジストラーは、古いレコードを定期的に削除するために使用できる同様の機能を提供します。

清掃は通常、既定では無効になっていますが、Windows クライアントのみを含むゾーンでは有効にする必要があります。ゾーンは、更新なしおよび更新間隔で構成されます。レコードは、最初の作成日とこれら 2 つの間隔を超えて経過すると期限切れになります。下の図は、清掃のタイム ラインの間隔を示しています。

図 *15:* アドレス レコードの清掃タイム ライン間隔



Cisco プライムネットワーク レジストラープロセスは次のとおりです。

1. クライアントが新しい A レコードで DNS サーバーを更新すると、このレコードはタイムスタンプを取得するか、クライアントがその A レコードを更新すると、タイムスタンプが更新される場合があります (「レコードが作成または更新されました」)。
2. 更新なし間隔 (既定値の 7 日) の間に、クライアントがアドレス変更なしで同じレコードを送信し続ける場合、レコードのタイムスタンプは更新されません。
3. レコードが非更新間隔を過ぎると、更新間隔 (7 日間の既定値) が入力され、その間に DNS 更新はタイムスタンプを更新し、レコードを更新しない間隔に戻します。
4. 更新間隔を過ぎたレコードは、清掃間隔に達したときに清掃に使用できます。

（注） 保護されていない R のみが清掃されます。R が清掃されないようにするには、それらを保護に設定します。ただし、ゾーンの最上位の (@) R は、保護されていない場合でも清掃されません。

次の DNS サーバー属性は、清掃に影響します。

- scvg-interval：DNS サーバーがゾーン内の古いレコードを確認する期間。値の範囲は 1 時間から 365 日です。また、サーバーに対して設定することもできます (既定値は 1 週間です) が、ゾーンの設定によって上書きされます。
- scvg-no-refresh-interval：動的または前提条件のみの DNS 更新などのアクションがレコードのタイムスタンプを更新しない間隔。この値は 365 日の範囲になります。ゾーンの設定は、サーバーの設定を上書きします (既定値は 1 週間です)。

- scvg-refresh-interval：DNS の更新がレコードのタイムスタンプを増分する間隔。更新なしと更新の間隔の両方が期限切れになると、レコードは清掃の候補になります。この値は 365 日の範囲になります。ゾーンの設定は、サーバーの設定を上書きします (既定値は 1 週間です)。
- scvg-ignore-restart-interval：サーバーを再起動するたびにサーバーがスカベンジング時間をリセットしないようにします。この間隔内で、Cisco Prime Network レジストラーはサーバー ダウン インスタンスと再起動の間の時間を無視します。

  値の範囲は 2 時間から 1 日です。この設定値より長い値を使用すると、Cisco Prime Network レジストラーは清掃期間を再計算し、サーバーの停止中に発生できないレコード更新を許可します。ゾーンの設定は、サーバーの設定を上書きします (既定値は 2 時間です)。

Cisco Prime Network レジストラー DNS サーバーが Windows クライアント(または自動定期的な DNS 更新を行うことがわかっているもの)から更新を受信するゾーンに対してのみ清掃を有効にします。上記の属性を設定します。Cisco プライムネットワーク レジストラー清掃マネージャは、サーバーの起動時に起動します。変更セットデータベースに対して清掃によって消去されたレコードがレポートされます。Cisco Prime Network レジストラーは、プライマリ ゾーンから清掃されたレコードのゾーン転送を通じてセカンダリ ゾーンに通知します。清掃が無効になっているゾーンを作成し(レコードにタイムスタンプがない)、その後有効にした場合、Cisco Prime Network レジストラーは各レコードのデフォルト タイムスタンプとしてプロキシ タイムスタンプを使用します。

1 つ以上のログ設定の清掃、清掃の詳細、ddns の更新、および ddns 更新の詳細を使用して清掃アクティビティを監視できます。

ローカル詳細 Web UI

[DNS サーバーの管理] ページで、[コマンド] をクリックして [DNS**コマンド**] ダイアログ ボックスを開きます。[すべてのゾーンを清掃する] の横にある [**実行**] アイコンをクリックします。

特定の前方ゾーンまたは逆ゾーンのみをスキャベンジするには、[ゾーンのゾーン コマンド] ページに移動します。 [スキャベンジ ゾーン] の横にある [**実行**] アイコンをクリックします。次に清掃がゾーンにスケジュールされている時刻を確認するには、[清掃開始時刻を取得] の横にある [**実行**] アイコンをクリックします。

CLI コマンド

清掃dns scavengeが有効になっているすべてのゾーンに使用します。ゾーンでgetScavengeStartTimeのアクションを使用して、清掃が次回開始される予定の時刻を確認します。

DHCPv4 の DHCID RR への移行

ネットワークが IPv4 から IPv6 アドレスへの移行を行うため、多くのネットワーク デバイスは IPv4 アドレスと IPv6 アドレスの両方を使用します。これらのデバイスは、同じホスト上の複数のインターフェイスを使用している場合や、異なるネットワークを使用している場合や、異

なる DHCP バージョンを使用している場合があります。これらのデバイスは、DHCP サーバーに関して一貫して識別する必要があり、それに応じて DHCP サーバーは DNS サーバーを更新します。

Cisco プライム ネットワーク レジストラー 8.1 以前では、DHCPv4 は TXT R を使用し、DHCPv6 は DHCID R を使用して DNS 更新を行います。クライアントが要求した名前の競合を避けるために、デュアルスタック クライアントは単一の前方 FQDN を使用できません。これらの競合は、主にクライアントが要求した名前に適用され、生成される名前には適用されません。これらの競合を避けるために、DHCPv4 と DHCPv6 の名前に異なるゾーンが使用されました。

Cisco プライムネットワーク レジストラー 8.2 以降では、DHCPv4 は TXT RR または DHCID RR を使用し、DHCPv6 は DNS アップデートに DHCID RR を使用します。DHCP サーバー全体の設定属性 dnsクライアント IDの既定値は txt であり、属性は個々の DNS 更新構成オブジェクトに対して構成されていません。DNS 更新は、次のいずれかの方法で設定できます。

- DHCPv4 の TXT RR と DHCPv6 の DHCID:この構成セットの dns クライアント IDを txt に有効にします。サーバーは、DHCPv4 DNS 更新で TXT RR を使用し、DHCPv6 DNS 更新には DHCID RR を使用します。この設定は、旧バージョンとの互換性のために使用されます。これは、Cisco Prime Network Registrar 8.1 以前では、DHCPv4 に TXT RR の使用のみをサポートしているためです。この設定は、Cisco Prime Network レジストラー 8.1 以前のクラスタがゾーンに対する DNS 更新に関与している場合に使用する必要があります。
- DHCPv4 と DHCPv6 の両方の DHCID RR—この構成を有効にするには、dns クライアント IDを dhcid に設定します。サーバーは、DHCPv4 および DHCPv6 DNS 更新の両方に DHCID RR を使用します。この設定は、デュアル スタック クライアントをサポートするために使用する必要があり、この構成をサポートするゾーンに対して DNS 更新を行うすべての DHCP サーバーが DHCID RR を使用するように構成されている場合にのみ使用できます。
- DHCID RR への移行— この構成を有効にするには、dns クライアント IDを dhcid への移行に設定します。強制DNS 更新属性を true に設定します。サーバーをリロードします。アップグレードする必要があるゾーンについては、dns-client-identity 属性を dhcid に設定し、サーバーに設定されている最長のリース時間が経過した後で force-dns-update 属性を以前の値に復元します。

（注） すべての DHCPv4 リソース・レコードが DHCID RR に更新されるまで、dhcid への移行属性を設定する必要があります。詳細については、DHCPv4 の DHCID RR への移行 （329 ページ）を参照してください。

- [TXT RR への後退]: この構成を有効にするには、dns クライアント IDをリグレスから txt に設定します。強制DNS 更新属性を true に設定します。サーバーをリロードします。アップグレードする必要があるゾーンについては、dns クライアント ID属性を txt に設定し、サーバーで最長のリース時間が設定された後で、force-dns-update属性を以前の値に復元します。

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 [**展開**] メニューの[DNS **更新**] サブメニューの[DNS **更新**構成] を選択して、[DNS 更新の一覧/ DNS 更新の構成の追加] ページを開きます。

ステップ 2 更新構成の名前を選択して、[DNS **更新の構成の編集**] ページを開きます。

ステップ 3 DNS 更新の設定で、DNS 更新設定で、移行から dhcid を DNS クライアント ID として設定します。

ステップ 4 必要に応じて、強制 DNS 更新を true に設定します。この設定を使用すると、TXT RR から DHCID RR への移行プロセスが迅速に行われます。

ステップ 5 前方ゾーンまたは反転ゾーンの清掃設定属性を次の値に設定します。

- scvg 有効に設定して true にします。

ステップ 6 DNS サーバーの清掃設定属性を次の値に設定します。

- scvg-intervalを最長リース時間に設定します。
- scvg-refresh-intervalを最長リース時間に設定します。
- scvg-no-refresh-interval を0 に設定します。

ステップ 7 すべての TXT R がゾーンの DR の DHCID R に変換されていることを確認します。すべての DHCPv4 リソース レコードが dhcid RR に更新されるまで、transition-to-dhcid 属性を設定する必要があります。一部の TXT RR エントリが DHCID RR に移行しない場合は、Cisco Prime Networkレジストラーの単一レコードの動的 RR 削除機能を使用して、これらの DNS エントリを手動で削除する必要があります。

ステップ 8 [保存（Save）] をクリックします。

Windows クライアントの DNS 更新の構成

Windows オペレーティング システムは DNS と、より少ない程度では DHCP に大きく依存しています。この依存性には、大規模な Windows 展開を行う前に、ネットワーク管理者側で慎重に準備する必要があります。Windows クライアントは、アドレス (A) レコードを使用して転送ゾーンを直接更新することで、自身のエントリを DNS に追加できます。逆ゾーンは、ポインター (PTR) レコードで更新できません。

クライアント DNS の更新

クライアントが DNS を直接更新することを許可することはお勧めしません。

Windows クライアントがアドレス レコードの更新を DNS サーバーに送信するには、次の 2 つの条件が適用される必要があります。

- Windows クライアントの [TCP/IP コントロール パネルRegister this connection’s addresses in DNS] 設定のDNSタブでチェック ボックスをオンにする必要があります。

- DHCP ポリシーは直接更新を有効にする必要があります(Cisco Prime Network レジストラーポリシーはデフォルトで有効にします)。

Windows クライアントは、DHCPREQUESTパケットでクライアント FQDN DHCP オプション (81) を送信して、DNS サーバーに A レコードを更新する意図を DHCP サーバーに通知します。完全修飾ドメイン名 (FQDN) を示すことによって、このオプションは、ドメイン名前空間内のクライアントの場所を明確に示します。FQDN 自体と共に、クライアントまたはサーバーは、クライアント FQDNオプションで次のいずれかのフラグを送信できます。

- 0クライアントは、その A レコードを DNS サーバーに直接登録し、DHCP サーバーは PTR レコードを登録します (有効になっているポリシーのクライアントレコード更新を許可する属性を使用して行われます)。
- 1:クライアントは、DHCP サーバーに対して、その A レコードと PTR レコードを DNS サーバーに登録するように要求します。
- 3DHCP サーバーは、クライアント要求に関係なく、A および PTR レコードを DNS サーバーに登録します(ポリシーの[クライアントのレコード更新を許可]属性を使用して行われる場合は、デフォルト値です)。このフラグを設定できるのは DHCP サーバーだけです。

DHCP サーバーは、DNS 更新が有効になっているかどうかに基づいて、DHCPACK 内のクライアントに対して、独自のクライアント FQDN応答を返します。ただし、0 フラグが設定されている場合(ポリシーでクライアントのレコード更新を許可する属性が有効になっている)、DNS 更新を有効または無効にすることは、クライアントがDNSサーバーに更新を送信できるため、無関係です。さまざまなプロパティの設定方法に基づいて実行されるアクションについては、次の表を参照してください。

表 *35 : Windows* クライアント *DNS* 更新オプション

DHCP クライアント アクション	DNS 更新	DHCP サーバーの操作
クライアント Registerthisconnection'saddressesinDNS FQDNをチェックして送信します。DHCP サーバーは、クライアント・ア・レコード更新を許可する	有効または無効	クライアントが A レコードを更新することを許可するクライアント fqdn (フラグ 0 を設定) で応答しますが、DHCP サーバーは引き続き PTR レコードを更新します。
をRegisterチェックします。クライアントFQDNを送信します。DHCPは、クライアントのレコード更新を許可することを無効にします。	[有効（Enabled）]	クライアントが DNS サーバーを直接更新することを許可しないことをクライアント fqdnで応答し (フラグ 3 を設定)、A および PTR レコードを更新します。
	無効	クライアント fqdnで応答せず、DNS サーバーも更新されません。

DHCP クライアント アクション	DNS 更新	DHCP サーバーの操作
チェックRegisterを解除..クライアントFQDNを送信します	[有効（Enabled）]	A レコードと PTR レコードを更新していることをクライアントFQDNで返します。
	無効	クライアント fqdnで応答せず、DNS サーバーも更新されません。
クライアント FQDNを送信しません。	[有効（Enabled）]	クライアント fqdnで応答しませんが、A レコードとPTR レコードを更新します。
	無効	クライアント fqdnで応答せず、DNS サーバーも更新されません。

DHCP サーバーは、クライアント要求を無視する client-fqdn オプションを設定できます。Cisco Prime Network レジストラーでこの動作を有効にするには、Windows クライアント用のポリシーを作成し、このポリシーのクライアントのレコード更新許可属性を無効にします。

Cisco プライムネットワーク レジストラーでは、次の属性がデフォルトで有効になっています。

- Server use-client-fqdn：サーバーは着信パケットで client-fqdn 値を使用しますが、host-name は確認しません。DHCP サーバーは、ドメイン名の値の最初のドットの後のすべての文字を無視します。クライアント名が予期しない文字を送信している可能性があるために、サーバーがクライアント名をクライアント fqdnから判別しないようにする場合にのみ、use-client-fqdnを無効にします。
- Server use-client-fqdn-first：サーバーは host-name オプション (12) を確認する前に、クライアントからの着信パケットで client-fqdn を確認します。クライアント fqdnにホスト名が含まれている場合、サーバーはそれを使用します。サーバーがオプションを見つけられない場合は、host-name値を使用します。use-client-fqdn-firstが無効になっている場合、サーバーはクライアント fqdnよりもホスト名の値を優先します。
- Server use-client-fqdn-if-asked：クライアントが要求した場合、サーバーは発信パケットの client-fqdn 値を返します。たとえば、クライアントはDNS アクティビティの状態を知りたい場合、DHCP サーバーがクライアント fqdn値を提示するように要求します。
- Policy allow-client-a-record-update：クライアントが client-fqdn フラグを 0 に設定（直接の更新を要求）している限り、クライアントは DNS サーバーで直接 A レコードを更新できます。それ以外の場合、サーバーは、他の構成プロパティに基づいて A レコードを更新します。

クライアント要求に返されるホスト名は、これらの設定によって異なります(下の表を参照)。

表 *36 :* クライアント要求パラメータに基づいて返されるホスト名

クライアントによるリクエスト	サーバー/ポリシー設定を使用する	結果のホスト名
host-name（オプション 12）を含む	使用ホスト名=真の使用クライアント-fqdn =false (または使用クライアント-fqdn-first =false)トリム ホスト名=true	最初のドットでトリムされたホスト名。例:ホスト名host1.bob が返されるホスト 1。
	(同じ以外)トリム ホスト名=false	host-name。例:ホスト名host1.bob が返されるホスト 1.bob。
クライアント FQDNを含む (オプション 81)	使用クライアント-fqdn=真の使用ホスト名=false (または使用クライアント-fqdn-first =true)	クライアント FQDNは最初のドットでトリムされます。例:クライアント fqdn host1.bob が返される例は、host1 です。
ホスト名(オプション 12) およびクライアント FQDN (オプション 81) を省略します。	または:使用ホスト名=偽の使用クライアント-fqdn =偽	クライアント/ポリシー階層別に設定します。
	(上記の場合と同じですが、次の場合は次の点を除き、ホスト名はクライアント/ポリシー階層で定義されず、合成名=true	合成規則に従って合成され、指定された合成名ステムの後にホストのハイフンで区切られた IP アドレスを追加します。
	(上記の場合と同じですが、次の場合は次の点を除き、合成名=偽	未定義。

Windows クライアント用デュアル ゾーンの更新

Windows DHCP クライアントは、2 つの DNS ゾーンに A レコードを持つ DHCP 展開の一部である場合があります。この場合、DHCP サーバーはクライアントがデュアル ゾーン更新を要求できるように、クライアント fqdnを返します。デュアル ゾーン更新を有効にするには、ポリシー属性の許可デュアル ゾーン DNS 更新を有効にします。

DHCP クライアントは、クライアント fqdnに 0 フラグを送信し、クライアントがメイン ゾーンの A レコードを使用して DNS サーバーを更新できるように、0 フラグを返します。ただし、DHCP サーバーは、クライアントの代わりにクライアントのセカンダリ ゾーンに基づいて A レコードの更新も直接送信します。クライアントのレコード更新と、デュアル ゾーンDNS の許可の両方が有効になっている場合、デュアル ゾーン更新が優先され、サーバーがセカンダリ ゾーン A レコードを更新できるようになります。

Windows クライアントの DNS 更新設定

Windows クライアントは、クライアント fqdnオプションの送信を有効にする詳細プロパティを設定できます。

ステップ 1 Windows クライアントで、コントロール パネルに移動し、[TCP/IP 設定] ダイアログ ボックスを開きます。

ステップ 2 [Advanced] タブをクリックします。

ステップ 3 [DNS] タブをクリックします。

ステップ 4 クライアントがクライアントの要求でクライアント fqdnオプションを送信するようにするには、Register this connection's addresses in DNSチェック ボックスをオンのままにします。これは、クライアントが A レコードの更新を実行することを示します。

DHCP サーバーの Windows クライアント設定

Windows クライアントを含むスコープに関連するポリシーを適用し、そのスコープの DNS 更新を有効にできます。

ステップ 1 Windows クライアントを含むスコープのポリシーを作成します。次に例を示します。

a) ポリシー win2k を作成します。ポリシーを作成する際には、前方または逆方向のゾーン名、メインおよびバックアップ サーバーの IP アドレスを指定する必要があります。

b) サブネット 192.168.1.0/24 と policywin2k をポリシーとして win2k スコープを作成します。アドレス範囲を 192.168.1.10 から 192.168.1.100 まで追加します。

ステップ 2 のDNS 更新設定の作成 （317 ページ）説明に従って、ゾーン名、サーバー アドレス (A レコードの場合)、逆引きゾーン名、および逆サーバー アドレス (PTR レコードの場合) を設定します。

ステップ 3 クライアントが DNS サーバーで A レコードを更新する場合は、ポリシー属性の[クライアント-レコードの更新を許可] を有効にします (これは事前設定値です)。これにはいくつかの注意点があります。

- クライアントのレコード更新を許可するが有効になっている場合、クライアントが更新ビットを有効にしてクライアント FQDNを送信すると、クライアントに返されるホスト名とクライアント FQDNはクライアントのクライアント fqdnに一致します。(ただし、サーバーでクライアント名の上書き fqdnも有効になっている場合、クライアントに返されるホスト名と FQDN は、構成されたホスト名とポリシー ドメイン名によって生成されます。
- その代わりに、クライアントが更新ビットを有効にしてクライアント fqdnを送信しない場合、サーバーは A レコードの更新を行い、クライアントに返されたホスト名とクライアント FQDN (要求された場合) は DNS 更新に使用された名前と一致します。
- クライアントのレコード更新を許可するが無効になっている場合、サーバーは A レコードの更新を行い、クライアントに返されるホスト名とクライアント FQDN (更新ビットが無効な) の値は、DNS 更新に使用された名前と一致します。
- 二重ゾーン DNS 更新が有効になっている場合、DHCP サーバーは常に A レコードの更新を行います。（Windows クライアント用デュアル ゾーンの更新 （334 ページ） を参照）。
- DHCP サーバーまたは DNS 更新の構成で use-dns-update-prereqs が有効（事前設定値）の場合、クライアントに返されるホスト名と client-fqdn は、DNS の更新と一致する保証はありません。ただし、リース データは新しい名前で更新されます。

RFC 2136 に従って、更新の前提条件により、プライマリ DNS サーバーが RR セットまたは名前のレコードが存在する必要があるかどうかに基づいて実行するアクションを決定します。まれな状況でのみ使用 dns 更新前の前提条件を無効にします。

ステップ 4 DHCP サーバーをリロードします。

SRV レコードと DNS 更新

Windows は、ネットワークへの広告サービスの DNS プロトコルに大きく依存しています。次の表は、Windows がサービスロケーション (SRV) DNS R および DNS 更新を処理する方法を示しています。

Cisco Prime Networkレジストラー DNS サーバーを設定して、Windows ドメイン コントローラがサービスを DNS に動的に登録し、それによってネットワークにアドバタイズできるようにすることができます。このプロセスは RFC 準拠の DNS アップデートによって行われるため、Cisco Prime Network レジストラーでは通常の方法で何もする必要はありません。

表 *37 : Windows SRV* レコードおよび *DNS* 更新

機能	説明

SRV レコード	Windows ドメイン コントローラは SRV RR を使用してネットワークにサービスをアドバタイズします。この RR は、RFC 2782 の「サービスの場所を指定するための DNS RR（DNS SRV）」で定義されています。RFC は SRV レコードの形式を定義します (DNS タイプ コード 33) は、次のように定義します。
	`_ service ._ protocol . name ttl class SRV priority weight port target`
	クライアントがホストにサービスを解決できるように、SRV レコードのターゲットに関連付けられた A レコードが常に必要です。SRV レコードの Windows 実装では、レコードは次のようになります。
	`myserver.example.com A 10.100.200.11`
	`_ldap._tcp.example.com SRV 0 0 389 myserver.example.com`
	`_kdc._tcp.example.com SRV 0 0 88 myserver.example.com`
	`_ldap._tcp.dc_msdcs.example.com SRV 0 0 88 myserver.example.com`
	アンダースコアは常にサービス名とプロトコル名の前に置きます。この例では、キー配布センター_kdcです。優先順位と重みにより、同じサービスを提供するターゲット・サーバー(優先順位が等しいサーバーを区別する重み)を選択できます。優先順位と重みがゼロの場合、リストされている順序によって優先順位が決まります。Windows ドメイン コントローラは、これらの SRV レコードを自動的に DNS に配置します。

SRV レコードの使用方法	Windows クライアントは、起動すると、ネットワーク ログイン プロセスを開始して、その Windows ドメイン コントローラに対して認証を試みます。クライアントは、まずドメイン コントローラの場所を検出し、動的に生成された SRV レコードを使用して検出する必要があります。net-login プロセスを起動する前に、クライアントはサービス名を使用して DNS を照会します。たとえば、_ldap._tcp.dc_msdcs.example.com です。たとえば、DNS サーバーの SRV レコード ターゲットは my-domain-controller.example.com。Windows クライアントは、ホスト名を使用して DNS にクエリを実行 my-domain-controller.example.com。DNS はホスト アドレスを返し、クライアントはこのアドレスを使用してドメイン コントローラを検索します。ネットログインプロセスは、これらの SRV レコードなしで失敗します。
DNS 更新	Windows サーバーをドメイン コントローラとして構成すると、Active Directory 管理コンソールを使用して、管理するドメインの名前を静的に構成することになります。この Windows ドメインには、対応する DNS ゾーンが関連付けられている必要があります。また、ドメイン コントローラのコントロール パネルの [TCP/IP プロパティ] で、一連の DNS リゾルバを構成する必要があります。Windows ドメイン コントローラは、起動時に次の手順を実行して、自身を DNS に登録し、そのサービスをネットワークにアドバタイズします。 1. 主に Windows ドメインを密封している DNS ドメインの権限 (SOA) レコードの開始を求めるクエリを実行します。 2. 主に Windows ドメイン名を密封している DNS ゾーン (SOA レコードから) のプライマリ DNS サーバーを識別します。 3. RFC 2136 DNS 更新プロトコルを使用して、このゾーンに一連の SRV レコードを作成します。

サーバーブートプロセスログファイルの例	通常の動作条件では、Cisco Prime Network レジストラープライマリ DNS サーバーは、Windows ドメイン コントローラが起動して SRV レコードを作成するときに、これらのログ エントリを書き込みます。
	`data time name/dns/1 Activity Protocol 0 Added type 33 record to name "_ldap._tcp.w2k.example.com", zone "w2k.example.com"`
	`data time name/dns/1 Activity Protocol 0 Update of zone "w2k.example.com" from address [10.100.200.2] succeeded.`
	このログには、1 つの SRV レコードに対して 1 つの DNS 更新のみが表示されます。Windows ドメイン コントローラは、通常、起動時にこれらの SRV レコードのうち 17 個を登録します。

Windows 環境に関連する問題

次の表では、Windows および Cisco Prime Network Registrar 間の接続相互運用性に関する問題について説明します。この表の情報は、現場で発生する可能性のある問題を事前に通知することを目的としています。Windows の相互運用性に関してよく寄せられる質問Windows の統合に関するよく寄せられる質問 （343 ページ）については、を参照してください。

表 *38 : Windows* および *Cisco* プライムネットワーク レジストラー相互運用性に関する問題

問題	説明

非表示動的に作成された R	Cisco プライムネットワーク レジストラーは、正しく設定されていれば、DHCP サーバーと Windows サーバーの両方から DNS アップデートを受け入れます。CLI を使用して、レコードの表示と削除のために DNS ゾーンの動的部分にアクセスできます。指定したゾーンのすべての DNS R を表示するには、次のコマンドを入力します。
	`nrcmd> zone myzone listRR dynamic myfile`
	これにより、出力が myfile ファイルにリダイレクトされます (次の例: 非表示の動的に作成された RRs セクションを示す出力を参照)。動的に生成されたレコードは、次のコマンドを入力して削除できます。
	`nrcmd> zone myzone removeDynRR myname [ type ]`
	nslookupを使用して、nslookup が存在するかどうかを確認したり、バージョン 5 を使用することもできます。動的 SRV レコードを表示する場合は、x (Windows に同梱されています)。このバージョンでは、**セット** type=SRVを使用して SRV レコードの表示を有効にします。

ドメイン コントローラ登録	Windows ドメイン コントローラは、DNS 更新を使用して自身を DNS に登録する必要があります。DNS RFC では、ゾーン データの編集を受け付けることができるのは、特定のゾーンのプライマリ DNS サーバーだけです。したがって、Windows ドメイン コントローラは、Windows ドメイン名を含むゾーンのプライマリ DNS サーバーを検出する必要があります。 ドメイン コントローラは、リゾルバー リスト (TCP/IP プロパティ コントロール パネルで構成) の最初の DNS サーバーに対してクエリを実行して、この問題を検出します。最初のクエリは、ドメイン コントローラの Windows ドメインを含むゾーンの SOA レコードを対象にしています。SOA レコードには、ゾーンのプライマリ サーバーの名前が含まれます。ドメイン名のゾーンが存在しない場合、ドメイン コントローラはドメイン名の左端のラベルを削除し続け、そのドメインに含まれるプライマリ サーバーを持つ SOA レコードが見つかるまでクエリを送信します。ドメイン コントローラは、そのドメインのプライマリ DNS サーバーの名前を持つと、DNS 更新を送信して必要な SRV レコードを作成します。 ゾーンのプライマリ DNS サーバーの名前が SOA レコードに含まれているかどうかを確認します。
レコード DNS 更新の失敗	Windows ドメイン コントローラがネットワークに対して自身をアドバタイズしようとすると、ドメインのレコードの DNS サーバーに複数の DNS 更新要求が送信されます。これらの更新要求のほとんどは SRV レコードに対する要求です。ただし、ドメイン コントローラは、Windows ドメインと同じ名前の単一の A レコードの更新も要求します。 Cisco Prime Network レジストラー DNS サーバーもこの Windows ドメインと同一のゾーンに対して権限を持っている場合、DNS A レコードの更新が静的 SOA および NS レコードと競合するため、A レコードの登録は拒否されます。これは、動的ホストが自分自身を登録し、サイトへの Web トラフィックを偽装するなど、セキュリティ侵害の可能性を防ぐためです。 たとえば、ドメイン コントローラは、Windows ゾーン w2k.example.comを制御できます。Cisco Prime Network レジストラー DNS サーバーに同じ名前のゾーンが存在する場合、これらの R はそのゾーンの一部である可能性があります。(例は以下の通りです。 `w2k.example.com. 43200 SOA nameserver.example.com. hostadmin.example.com. ( 98011312 ;serial 3600 ;refresh 3600 ;retry 3600000 ;expire 43200 ) ;minim w2k.example.com.86400 NS nameserver.example.com`

	ドメイン コントローラは、レコードを追加しようとします。例えば：
	`w2k.example.com. 86400 A 192.168.2.1`
	Cisco Prime Network レジストラーでは、DNS の更新がゾーン内の静的に設定された名前と競合することはありません。上記の例では、名前に関連付けられた A レコードを追加しようとすると、SOA レコードと NS レコードw2k.example.com競合します。
	ドメイン コントローラが起動すると、次のような DNS ログ ファイル エントリが表示されます。
	`0 8/10/2000 16:35:33 name/dns/1 Info Protocol 0 Error - REFUSED -`
	`Update of static name "w2k.example.com", from address [10.100.200.2]`
	Cisco プライムネットワーク レジストラーが静的 DNS データの DNS アップデートに応答する方法は、次のようになります。さらに、この DNS 更新の失敗を無視できます。Windows クライアントはこの A レコードを使用しません。ドメイン コントローラの割り当ては、SRV レコードを通じて行われます。マイクロソフトは、SRV レコードをサポートしない従来の NT クライアントに対応するために A レコードを追加しました。
	コントローラ A レコードの登録に失敗すると、ドメイン コントローラのブートアップ プロセスが遅くなり、ワーカー クライアントの全体的なログインに影響することに注意してください。前述のように、Windows ドメインを権限のあるゾーンのサブドメインとして定義するか、DNS サーバーのゾーントップ dynupdate属性をシミュレートする方法を使用します。これが不可能な場合は、シスコテクニカル アシスタンス センターに問い合わせてください。

RC1 DHCP クライアントを使用します。	マイクロソフトは、壊れた DHCP クライアントを使用して Windows ビルド 2072 (RC1) をリリースしました。このクライアントは、Cisco Prime ネットワークレジストラーが解析できない、不正な形式のパケットを送信します。Cisco Prime Network レジストラーはパケットを廃棄し、クライアントにサービスを提供できません。
	`08/10/2000 14:56:23 name/dhcp/1 Activity Protocol 0 10.0.0.15 Lease offered to Host:'My-Computer' CID: 01:00:a0:24:1a:b0:d8 packet'R15' until True, 10 Aug 2000 14:58:23 -0400. 301 ms.`
	`08/10/2000 14:56:23 name/dhcp/1 Warning Protocol 0 Unable to find necessary Client information in packet from MAC address:'1,6,00:d0:ba:d3:bd:3b'. Packet dropped!`
	Cisco Prime Network レジストラーには、この不適切に構築された FQDN オプションなどのエラーに対処するために特別に設計されたエラーチェックが含まれています。ただし、この問題が発生した場合は、DHCP クライアントの RC1 クライアントにマイクロソフトの修正プログラムをインストールします。この修正プログラムは、マイクロソフトから入手する必要があります。

Windows プラグ アンド プレイ ネットワーク インターフェイス カード (NIC) の構成	DHCP を使用するように構成されている場合、Windows システムは起動時に DHCP リースを取得しようとします。DHCP サーバーが利用できない場合、Windows は、プラグ アンド プレイ IP アドレスを使用してコンピュータ インターフェイスを自動的に構成することがあります。このアドレスは、ネットワーク管理者または DHCP サーバーが構成または選択したアドレスではありません。 これらのプラグアンドプレイアドレスは、169.254.0.0/16の範囲内にあります。ネットワーク上にこのアドレス範囲のデバイスが表示される場合は、DHCP サーバーからリースを取得できないため、Windows がインターフェイスを自動構成したことを意味します。 これにより、ネットワークやトラブルシューティングに関する重大な問題が発生する可能性があります。Windows システムは、DHCP クライアントがリースを取得できなかったことをユーザーに通知しなくなりました。すべてが正常に機能しているように見えますが、クライアントはローカルサブネットからパケットをルーティングできません。さらに、DHCP クライアントが 169.254.0.0/16 ネットワークからのアドレスを使用してネットワーク上で動作しようとしているのを見ることができます。これにより、Cisco Primeネットワーク レジストラー DHCP サーバーが壊れ、間違ったアドレスを配っていると考える場合があります。
	この問題が発生した場合、次のステップを実行します。 1. DHCP クライアントにアクティブなネットワーク ポートと正しく構成された NIC があることを確認します。 2. クライアントと DHCP サーバー間のネットワークが正しく構成されていることを確認します。すべてのルータ インターフェイスが正しい IPHelper アドレスで設定されていることを確認します。 3. DHCP クライアントを再起動します。 4. 必要に応じて、DHCP ログ ファイルを確認します。DHCP クライアントがパケットをサーバーに正常にルーティングできる場合、Cisco Prime Network レジストラーがパケットに応答しない場合でも、DHCPDISCOVER がログに記録されます。 ネットワークが正しく設定され、DHCP クライアントが破損していない場合、Cisco Prime Network レジストラーはパケットを受信してログに記録する必要があります。パケット受信のログ エントリがない場合は、ネットワークのどこか別の場所で問題が発生します。

Windows クライアント アドレス レコードの清掃	Windows クライアントは、動的レコード登録が無期限に残る可能性があり、自分自身の後にクリーンアップされません。これにより、古いアドレス レコードが DNS サーバーに残ります。これらの古いレコードが定期的に削除されるようにするには、ゾーンの清掃を有効にする必要があります (を参照 動的レコードのスカベンジング（328 ページ）)。

例: 非表示の動的に作成された R を示す出力

```
 Dynamic Resource Records _ldap._tcp.test-lab._sites 600 IN SRV 0
100 389 CNR-MKT-1.w2k.example.com. _ldap._tcp.test-lab._sites.gc._msdcs 600 IN
SRV 0 100 3268 CNR-MKT-1.w2k.example.com.
_kerberos._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 88
CNR-MKT-1.w2k.example.com. _ldap._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0
100 389 CNR-MKT-1.w2k.example.com. _ldap._tcp 600 IN SRV 0 100 389
CNR-MKT-1.w2k.example.com. _kerberos._tcp.test-lab._sites 600 IN SRV 0 100 88
CNR-MKT-1.w2k.example.com. _ldap._tcp.pdc._msdcs 600 IN SRV 0 100 389
CNR-MKT-1.w2k.example.com. _ldap._tcp.gc._msdcs 600 IN SRV 0 100 3268
CNR-MKT-1.w2k.example.com.
_ldap._tcp.1ca176bc-86bf-46f1-8a0f-235ab891bcd2.domains._msdcs 600 IN SRV 0 100
389 CNR-MKT-1.w2k.example.com. e5b0e667-27c8-44f7-bd76-6b8385c74bd7._msdcs 600
IN CNAME CNR-MKT-1.w2k.example.com. _kerberos._tcp.dc._msdcs 600 IN SRV 0 100
88 CNR-MKT-1.w2k.example.com. _ldap._tcp.dc._msdcs 600 IN SRV 0 100 389
CNR-MKT-1.w2k.example.com. _kerberos._tcp 600 IN SRV 0 100 88
CNR-MKT-1.w2k.example.com. _gc._tcp 600 IN SRV 0 100 3268
CNR-MKT-1.w2k.example.com. _kerberos._udp 600 IN SRV 0 100 88
CNR-MKT-1.w2k.example.com. _kpasswd._tcp 600 IN SRV 0 100 464
CNR-MKT-1.w2k.example.com. _kpasswd._udp 600 IN SRV 0 100 464
CNR-MKT-1.w2k.example.com. gc._msdcs 600 IN A 10.100.200.2
_gc._tcp.test-lab._sites 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com.
```

Windows の統合に関するよく寄せられる質問

Cisco Primeネットワーク レジストラー DNS サービスと Windows の統合について、次の質問がよく寄せられます。

Windows クライアントと DHCP サーバーの両方が同じゾーンを更新できる場合の動作これにより、古い DNS レコードがゾーンに残される可能性が生まれますか。もしそうなら、それについて何ができますか?

Windows クライアントがゾーンを更新することを許可しないことをお勧めします。代わりに、DHCP サーバーはすべてのクライアントの動的 RR レコードを管理する必要があります。DNS 更新を実行するように構成されている場合、DHCP サーバーはリースを提供したクライアントに関連付けられたすべての DR を正確に管理します。これに対し、Windows クライアント マシンは、毎日の DNS 更新をサーバーに盲目的に送信し、ネットワークから削除された場合は、古い DNS エントリを残します。

DNS 更新クライアントによって更新されるゾーンでは、一時的な Windows クライアントが残す古い R の長寿を短縮するために DNS の清掃機能を有効にする必要があります。DHCP サーバーと Windows クライアントの両方が同じゾーンを更新している場合、Cisco Prime Network レジストラーでは次の 3 つのことが必要です。

1. ゾーンの清掃を有効にします。

2. 各クライアントがリースを更新するたびに、DHCP サーバーが DNS 更新エントリを更新するように構成します。デフォルトでは、Cisco Primeネットワーク レジストラーは、作成から最終削除までの間に DNS レコードを再度更新しません。Cisco プライムネットワーク レジストラーがリースの開始からリースの期限が切れるまで、ライフを作成する DNS 更新レコード。この動作は、DHCP サーバー (または DNS 更新構成) 属性 (強制 DNS 更新) を使用して変更できます。次に例を示します。

```
nrcmd> dhcp enable force-dns-updates

100 Ok
force-dns-updates=true
```

3. 特定のゾーンで清掃が有効になっている場合、DHCP サーバーが代わりにそのゾーンを更新するクライアントに関連付けられているリース時間は、更新なし間隔および更新間隔の清掃設定の合計より小さくなければなりません。これらの設定は両方とも 7 日間に設定されています。これらのデフォルト値を変更しない場合は、リース期間を 14 日以下に設定できます。

重複する DNS ドメインと Windows ドメインを持たないと判断した場合に Windows ドメインを既存の DNS ドメインの命名構造と統合するのに必要な手順たとえば、example.com という既存の DNS ドメインがあり、w2k.example.comという Windows ドメインが作成されている場合、Windows ドメインを DNS ドメインに統合するには何をする必要がありますか。

この例では、Windows ドメイン フォレストのツリーにルートの w2k.example.com があります。example.comという名前の DNS ドメインが存在します。この DNS ドメインは、example.comという名前のゾーンで表されます。このゾーンに表される追加の DNS サブドメインが存在する可能性がありますが、このゾーンからそのゾーンに委任されるサブドメインはありません。すべてのサブドメインは常にexample.comに存在します。ゾーン。

この場合、ドメイン コントローラからの DNS 更新はどのように処理されますか。

Windows ドメイン コントローラからの SRV レコードの更新を処理するには、DNS の更新をexample.comに制限します。ゾーンは IP アドレスによってのみドメイン コントローラに接続されます。(後で、DHCP サーバーの IP アドレスも一覧に追加します)。ゾーンの清掃を有効にします。コントローラは、example.com ゾーン内のw2k.example.com サブドメインの SRV レコードと A レコードを更新します。w2k.example.comの A レコードは、EXAMPLE.COM ゾーン内の SOA、NS、またはその他の静的レコードと競合しないため、各ドメイン コントローラからの A レコードの更新を処理するために特別な構成は必要ありません。

example.comゾーンには、次のレコードが含まれる場合があります。

```
example.com. 43200 SOA ns.example.com. hostadmin.example.com. (
98011312 ;serial
3600 ;refresh
3600 ;retry
3600000 ;expire
43200 ) ;minimum
example.com.86400 NS ns.example.com
ns.example.com. 86400 A 10.0.0.10
_ldap._tcp.w2k.example.com. IN SRV 0 0 389 dc1.w2k.example.com
w2k.example.com 86400 A 10.0.0.25
...
```

この場合、個々の Windows クライアント マシンからのゾーン更新はどのように処理されますか。

このシナリオでは、クライアントは、example.comを更新しようとする可能性があります。w2k.example.com ドメインの更新を含むゾーン。これを回避する方法は、信頼できるソースからのゾーンを更新プログラムに閉じる方法です。Cisco Prime Network レジストラーでは、DHCP サーバーとexample.com ゾーンのプライマリ DNS サーバーの間でトランザクション シグニチャ(TSIG)を使用できます。

DHCP サーバーを構成して、example.com ゾーンに対して DNS 更新を行い、各クライアントに対して適切な逆ゾーンを使用し、オプション 81 を使用してクライアントが DNS 更新を実行できないようにします。

この場合、セキュリティは対処されていますか?

信頼された IP アドレスからの更新のみを受け入れるように、前方ゾーンと逆方向のゾーンを構成すると、ネットワーク上の他のデバイスからの更新プログラムに対してゾーンを閉じます。IP によるセキュリティは、なりすまし IP アドレス ソースからの悪意のある攻撃を防ぐことができないので、最も理想的なソリューションではありません。DHCP サーバーと DNS サーバーの間で TSIG を構成することで、DHCP サーバーからの更新をセキュリティで保護できます。

この場合、清掃は必要ですか?

いいえ。更新は、ドメイン コントローラと DHCP サーバーからのみ受け付けられます。DHCP サーバーは、追加するレコードのライフ サイクルを正確に維持し、清掃を必要としません。Cisco Prime Network レジストラーの単一レコード動的 RR 削除機能を使用して、ドメイン コントローラのダイナミック エントリを手動で管理できます。

名前空間を DNS ドメインと共有する Windows ドメインを統合するのに必要な手順たとえば、example.comという既存の DNS ゾーンがあり、example.comという Windows Active Directory ドメインを展開する必要がある場合、どうすればいいでしょうか。

この例では、Windows ドメイン フォレストのツリーにルート example.comが含まれます。example.comという名前のゾーンで表されるexample.comという名前の既存のドメインもあります。

この場合、個々の *Windows* クライアント マシンからの *DNS* 更新はどのように処理されますか。

SRV レコードの更新を処理するには、次のサブゾーンを作成します。

```
_tcp.example.com.
_sites.example.com.
_msdcs.example.com.
_msdcs.example.com.
_udp.example.com.
```

DNS の更新をこれらのゾーンに対して、IP アドレスのみでドメイン コントローラに制限します。これらのゾーンで清掃を有効にします。

各ドメイン コントローラからの A レコードの更新を処理するには、DNS サーバー属性であるゾーン トップ dynupdateをシミュレートする属性を有効にします。

```
nrcmd> dns enable simulate-zone-top-dynupdate
```

必須ではありませんが、必要に応じて、ドメインコントローラのA レコードを手動でexample.com ゾーンに追加します。

この場合、個々の Windows クライアント マシンからのゾーン更新はどのように処理されますか。

このシナリオでは、クライアントがexample.comゾーンを更新しようとする可能性があります。これを回避する方法は、信頼できるソースからのゾーンを更新プログラムに閉じる方法です。Cisco Prime Network レジストラーでは、DHCP サーバーとexample.com ゾーンのプライマリ DNS サーバーの間でトランザクション シグニチャ(TSIG)を使用できます。

DHCP サーバーを構成して、example.com ゾーンに対して DNS 更新を行い、各クライアントに対して適切な逆ゾーンを使用し、オプション 81 を使用してクライアントが DNS 更新を実行できないようにします。

この場合、セキュリティは対処されていますか?

信頼された IP アドレスからの更新のみを受け入れるように、前方ゾーンと逆方向のゾーンを構成すると、ネットワーク上の他のデバイスからの更新プログラムに対してゾーンを閉じます。IP によるセキュリティは、なりすましソースからの悪意のある攻撃を防ぐことができないので、最も理想的なソリューションではありません。DHCP サーバーと DNS サーバーの間で TSIG が構成されている場合、DHCP サーバーからの更新の方が安全です。

この場合、清掃は対処されていますか?

はい。サブゾーン _tcp.example.com、_sites.example.com、_msdcs.example.com、_msdcs_msdcs.example.com、および _udp.example.com ゾーンは、ドメイン コントローラーからのみ更新を受け入れ、これらのゾーンに対して清掃が有効になっています。example.com ゾーンは、DHCP サーバーからのみ DNS 更新を受け付けます。

GSS-TSIG の設定

AD と統合するための Cisco プライムネットワーク レジストラー DNS 設定

AD を Cisco プライムネットワーク レジストラー DNS 設定と統合するには、次の手順を実行します。

ステップ 1 Cisco プライムネットワーク レジストラー DNS をワークグループ マシンにインストールします。

ステップ 2 ゾーンを作成します (AD のドメインと同じです)。

DCpromo.exe を使用して WINDOWS サーバーに AD をインストールし、Cisco Prime Network Registrar DNS と統合します。

ステップ 3 Cisco プライムネットワーク レジストラー DNS に SRV レコードが追加されていることを確認します。

```
DCHOSTNAME. DOMAIN.COM A AD-IP-ADDRESS
_ldap._tcp.DOMAIN.COM. SRV 0 0 389 DCHOSTNAME.DOMAIN.COM.
_kerberos._tcp.DOMAIN.COM. SRV 0 0 88 DCHOSTNAME.DOMAIN.COM.
_ldap._tcp.dc._msdcs.DOMAIN.COM. SRV 0 0 389 DCHOSTNAME.DOMAIN.COM.
_kerberos._tcp.dc._msdcs.DOMAIN.COM. SRV 0 0 88 DCHOSTNAME.DOMAIN.COM.
```

（注） DCHOSTNAMEは AD ホスト名を参照し、DOMAIN.COMは AD に存在するドメインです。

（注） サーバー間の通信に Kerberos サーバーを使用する場合は常に、/etc/krb5.conf にある最新の暗号化アルゴリズムを使用することを推奨しています。

Cisco Prime Network Registrar および AD を、Windows 環境の同じドメインの下に置きます。

ステップ 1 ドメインを変更し、**コンピューター** > **プロパティ** > **コンピューター名** >ドメインのメンバーを変更します (AD のドメインと同じ)。

ステップ 2 コントロール パネル > ネットワークとインターネット > ネットワークと共有センター > ローカル エリア接続 > プロパティ > TCP/IPV4 > 優先 DNS (Cisco Cisco プライムネットワーク レジストラー DNS 実行 IP)。

ステップ 3 コンピューターを再起動し、AD に存在するユーザーでログインします。

ステップ 4 AD にログインし、次の操作を行います。

- DNS アクティブ ホスト名が追加されていることを確認する、 AD **サーバー マネージャー** > **コンピューター**

```
setspn -s DNS/ <hostname of the DNS server> <Computer Name>
```

DNS サーバーを AD-KDC に統合する

プライマリ DNS サーバーは AD-KDC に統合されています。

ステップ 1 SRV レコードを持つ /etc/krb5.conf または DNS サーバーが、必要な AD に到達するように構成されていることを確認します。

```
krb5.conf cofiguration
[libdefaults]
ticket_lifetime = 24h
default_realm = <AD REALM>
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
dns_lookup_realm = true
dns_lookup_kdc = false
forwardable = true
<AD REALM> = {
  kdc =< AD-HOSTNAME>:88
```

```
  admin_server = =< AD-HOSTNAME:749
  default_domain = <AD REALM>
}
```

（注） AD-HOSTNAME が解決可能であることを確認します。

（注） サーバー間の通信に Kerberos サーバーを使用する場合は常に、/etc/krb5.conf にある最新の暗号化アルゴリズムを使用することを推奨しています。

ステップ 2 Windows Server Active Directory にサービス アカウントを作成します。

1. Active Directory Users and Computers 管理ツールを使用して、新しいユーザー アカウントを作成します。
   - ユーザー名をスペースなしでアカウントに割り当てます。
   - アカウントにパスワードを割り当てます。

     （注） パスワードの有効期限が切れたり変更された場合は、**キータブ**ファイルを新しい関連付け kvnoで生成する必要があります。

2. SETSPN を使用するアカウントにサービス プリンシパル名 (SPN) を割り当てます。Exe。SPN は、デプロイメントに応じてサービス名/ホスト名/ドメインです。1 つのアカウントに複数の SPN を割り当てることができます。

   たとえば、<service-name> と <hostname> を指定します。

   ```
   setspn -s DNS/<DNS running Computer Name> <Service Name>
   ```

3. kvnoの詳細を取得します。

   ```
   ldifde -f <Filename> -d "DC=<DOMAIN>,DC=com" -l *,msDS-KeyVersionNumber -r
   "(serviceprincipalname=<service-principal name>)" -p subtree OR kvno.exe <service-principal
   name>@<REALM>
   ```

4. ktpass.exe コマンドを使用してキータブ ファイルをジェネタレします。

   ```
   ktpass -out<filename>  -princ <Principal name> -pass <password associated with the user> -crypto
    all -ptype KRB5_NT_PRINCIPAL -kvno <Kvno details>
   ```

   キータブファイルを Linux マシンに転送し、Kutil を実行して、Keytab 項目を既存のキータブファイルに追加します。

   ```
   > ktutil
    ktutil:  rkt <keytab file name>
    ktutil:  wkt /etc/krb5.keytab
    ktutil:  q
   ```

ステップ 3 以下を使用して、キー・タブ項目を表示します。

```
klist -k -t -e /etc/krb5.keytab
```

Linux 上のプライマリ DNS サーバー MIT-KDC に統合

サービス プリンシパル名を MIT KDC に関連付けるには、次の手順を実行します。

ステップ 1 Linux DNS サーバーにログインし、kadmin ユーティリティを使用して、MIT-KDC にプリンシパル名を追加します。

```
>kadmin
Authenticating as principal <MIT-KDC USER@REALM> with password.
Password for <MIT-KDC USER@REALM.COM > : <Enter the associated Password>
kadmin:  addprinc -randkey DNS/<hostname of the DNS server>
WARNING: no policy specified for DNS/<hostname of the DNS server>@REALM; defaulting to no policy
add_principal: Principal or policy already exists while creating " DNS/<hostname of the DNS
server>@REALM".
kadmin:  ktadd -randkey DNS/<hostname of the DNS server>
kadmin: Principal -randkey does not exist.
Entry for principal DNS/<hostname of the DNS server> with kvno x, encryption type AES-256 CTS mode
 with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal DNS/<hostname of the DNS server>with kvno x, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal DNS/<hostname of the DNS server>with kvno x, encryption type Triple DES cbc mode
 with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal DNS/<hostname of the DNS server>with kvno x, encryption type ArcFour with HMAC/md5
 added to keytab WRFILE:/etc/krb5.keytab.
kadmin:  quit
```

ステップ 2 次を使用して、keytab のエントリを表示します。

```
klist -k -t -e /etc/krb5.keytab
```

ステップ 3 Linux サーバーを実行している MIT-KDC にログインし、追加されたプリンシパル名に上記と同じ kvno が関連付けられているかどうかを確認します。

```
Kvno DNS/<hostname of the DNS server>
```

GSS-TSIG 設定のトラブルシューティング

GSS/SSPI の障害およびメジャー/マイナーステータスの詳細を取得するには、DNS サーバーで DEBUG オプションを有効にし、値 g=3 を設定します。

- "キー テーブルのプリンシパルのキー バージョン番号が正しくありません。

  KVno から返される, klist -k -t -e /etc/krb5.keytab DNS 実行中のマシンで kvno は KDC で同じ kvno でなければなりません。

  AD-KDCにおけるknvoの検証:

  ```
  ldifde -f c:\spn1_out.txt -d "DC=TIG,DC=com" -l *,msDS-KeyVersionNumber -r
  "(serviceprincipalname=DNS/WIN-CPNUV*)" -p subtree
  ```

  kvno の検証は、MIT- KDC です。

  ```
  Kvno <principal name>
  ```

- "間違ったプリンシパル名"

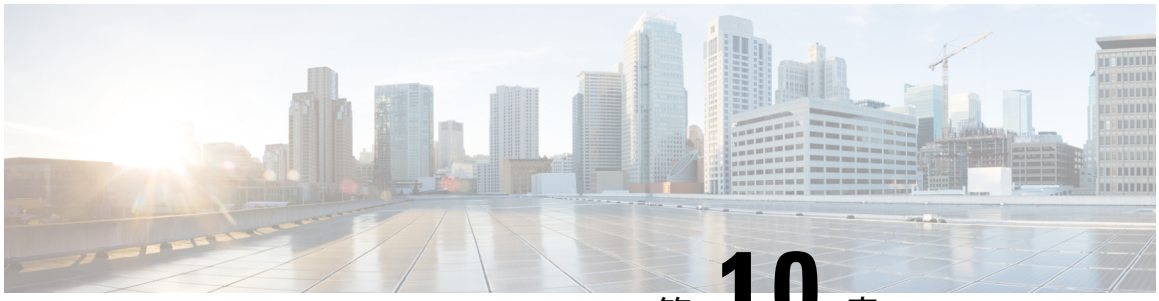
  GSS クライアントとサーバーが、サービス チケットの暗号化と復号化に使用されるのと同じサービス キーを使用していることを確認します。

DNS 更新のトラブルシューティング

などのdig標準 DNS ツールを使用nslookupして、サーバーに対してラールを照会できます。このツールは、動的に生成された RR が存在するかどうかを判断する際に役立ちます。次に例を示します。

```
$ nslookup
default Server: server2.example.com
Address: 192.168.1.2

> leasehost1.example.com
Server: server2.example.com
Address: 192.168.1.100

> set type=ptr
> 192.168.1.100
Server: server2.example.com
Address: 192.168.1.100
100.40.168.192.in-addr.arpa name = leasehost1.example.com
40.168,192.in-addr.arpa nameserver = server2.example.com
```

ログ設定属性をddnsに設定して DNS サーバーの DNS 更新を監視したり、dns-details に設定して詳細を表示したりできます。



第 10 章

クライアントクラスとクライアントの管理

Cisco Prime Network レジストラークライアントとクライアント クラスの概念を使用して、共通のネットワークを介してユーザーに差別化されたサービスを提供します。管理基準に基づいてクライアントをグループ化し、各グループが適切なサービス クラス (COS) を受け取ることを確認できます。クライアント クラスの処理を行わない場合、DHCP サーバーはネットワーク上の場所のみに基づいてクライアント リースを提供します。

クライアントクラスの設定

クライアント サービスは、次の方法で区別できます。

- Cisco Prime Networkレジストラーデータベース(このセクション)またはライトウェイト ディレクトリ アクセスLDAP を使用するように Cisco Prime Network Registrar を設定する （372 ページ） プロトコル(を参照)を使用してクライアントを登録します。
- アップストリーム クライアントをサービス クラス別に区別できるように、仲介デバイス (ケーブル モデムなど)を登録します。
- クライアント データの予知なしに、クライアント パケットの内容を使用します。
  - パケット内に存在する既知の DHCP オプション(dhcp-user-class-id DHCP オプション (77)、またはリレー エージェント情報DHCP オプションのradius 属性サブオプション外部ソースを含むクライアント データの処理 （357 ページ） (82 を参照)
  - クライアント クラス lookup-ID DHCP サーバー属性の式を使用して抽出するパケット内のその他クライアントクラスの計算とキーの作成 （369 ページ） のデータ (を参照)

- クライアントクラスを作成してクライアントを割り当て、次に特定のクライアントに対してクライアントルックアップIDを設定する2サブスクライバ制限のための式処理 （370ページ）段階のプロセスを使用します (「」を参照)。

関連項目

クライアントクラス処理 （352 ページ）
クライアントクラスの定義 （352 ページ）
スコープとプレフィックスの選択タグの設定 （354 ページ）
クライアントクラス ホスト名プロパティの定義 （356 ページ）
クライアントと組み込みポリシーの編集 （363 ページ）
外部ソースを含むクライアント データの処理 （357 ページ）
クライアントクラスのトラブルシューティング （360 ページ）

クライアントクラス処理

DHCP サーバーのクライアント クラス処理を有効または無効にし、一連のプロパティをクライアントのグループに適用します。クライアントクラスが有効な場合、サーバーは、一致する DHCPv4 スコープまたは DHCPv6 プレフィックスからクライアントをアドレスに割り当てます。サーバーはパケット内のデータに従って動作します。クライアント クラスを構成するには、次の手順に従います。

1. DHCP サーバーのクライアントクラス処理を有効にします。
2. 選択タグ (条件) を含む、または除外するクライアント クラスを定義します。
3. 選択タグを特定のスコープまたはプレフィックス (またはそのテンプレート) に適用します。
4. これらのクラスにクライアントを割り当てます。

このプロセスは Cisco Primeネットワーク レジストラーを通じて設定されたクライアントに対するものです。外部ソースからのデータの影響を受ける処理外部ソースを含むクライアント データの処理 （357 ページ）については、を参照してください。

クライアントクラスの定義

クライアント クラスは、サーバー レベルで有効にして定義します。

ローカル Web UI

ステップ 1 基本モードまたは詳細モードでクライアントクラスを有効にするには、次の手順を実行します。

a) Deploy メニューで、[DHCP] サブメニューから DHCP Server を選択し、[DHCPサーバーの管理（Manage DHCP Server）] ページを開きます。

b) [DHCP サーバー] ペインでサーバーを選択します。
c) [DHCP サーバーの編集] タブで、クライアント クラス属性を有効にします。
d) Save をクリックします。

ステップ 2 [**デザイン**] メニューの[DHCP **設定**] Classesサブメニューの [**クライアント**] を選択して、[DHCP クライアント クラスの一覧/追加] ページを開きます。

ステップ 3 [クライアント クラス] ウィンドウの [**クライアント クラスの追加**] アイコンをクリックして、[DHCP クライアント クラスの追加] ダイアログ ボックスを開きます。

ステップ 4 クライアントクラス名を入力します。

ステップ 5 その他のクライアント クラス プロパティを設定します。ホスト名とドメイン名の属性は、DNS 更新の構成を使用していない場合、主にDNS 更新設定の作成 （317 ページ）DNS 更新に使用されます (を参照してください)。ホスト名のプロパティについては、クライアントクラス ホスト名プロパティの定義 （356 ページ）で説明します。クライアント クラスに適したポリシーを選択することもできます。

ステップ 6 [**クライアント クラスの追加**]をクリックします。

ステップ 7 選択基準を定義します。

クライアントクラスを作成する際の重要なステップは、クライアントクラスを DHCPv4 スコープまたは DHCPv6 プレフィックスに関連付けることができるように、選択基準を定義することです。選択基準属性を使用します (表 39 : 使用する選択タグと基準属性 も参照)。

複数の選択タグをコンマで区切って入力できます。値は、目的のスコープまたはプレフィックスに設定された選択タグと一致するスコープとプレフィックスの選択タグの設定 （354 ページ）必要があります (を参照)。

ステップ 8 クライアント クラスに埋め込みポリシーを追加するにはクライアントと組み込みポリシーの編集 （363 ページ）、「」を参照してください。

ステップ 9 [保存（Save）] をクリックします。

ステップ 10 必要に応じてデバッグします。クライアント クラスのエラーをデバッグするには、[ローカルDHCP サーバー] ページの [ログ設定] セクションでクライアント基準処理属性を有効にします。

ステップ 11 クライアントクラスを削除するには、クライアントを選択し、左側の[クライアントクラス]ペインの[**クライアントクラスの削除**]アイコンをクリックして、削除を確認します。

CLI コマンド

クライアントクラスを有効にするには、dhcp enable client-class を使用します。クライアント クラスを作成するにはclient-class 、name createを使用します。名前は、その意図を明確に識別する必要があります。大文字と小文字は区別されません。クラスPCはクラスPCと同じです。

client-class nameset属性=値を使用して、クライアント クラスのクライアントのプロパティを設定します。たとえば、client-class 名前 set policy-name=値を使用して、クライアント クラスに関連付ける必要のあるポリシーを設定します。client-class nameをset使用して、スコープをクライアント クラスに関連付selection-criteriaけます。（スコープとプレフィックスの選択タグの設定 （354 ページ）を参照してください）。

client-class name [ ]showを使用して、作成したクライアント クラスのプロパティを表示します。作成されたすべてのクライアント クラスのプロパティを一覧表示したり、名前だけを一覧表示したりすることもできます。クライアント クラスの処理をデバッグするには、dhcp set log-settings=client-criteria-processingを使用します。クライアント クラスを削除するにはclient-class 、namedeleteを使用します。

DHCPv6 クライアントクラスの設定

DHCPv6 クライアント クラス属性は次のように設定できます。

- v6-client-lookup-id：クライアントデータベースで（ローカルに、または LDAP を介して）DHCPv6 クライアントを検索するために使用するキーの値。文字列（または有効な文字列としての BLOB）に対して評価する式として指定されます。
- v6-override-client-id：着信パケットで client-identity 値を置き換える値。BLOB に対して評価する式として指定します。

ローカル アドバンスド Web UI

メニューからDesignサブClientsメニューの下をDHCPSettings選択して、[DHCP クライアントの一覧/追加]ページを開きます。既存のクライアントを選択して[DHCP クライアントの編集]Add Clientsページを開くか、[クライアント] ペインのアイコンをクリックして新しいクライアント クラスを追加し、設定された DHCPv6 属性DHCPv6 クライアントクラスの設定 （354 ページ）を含むクライアント クラスを選択します (を参照Save) 。

ヒント　DHCP サーバーの検証クライアント名-mac属性を無効にします。

CLI コマンド

既存clientlistのclientクライアントを表示するには、または名前showを使用します。クライアントのクライアント クラス名を設定するにはclient 、name set client-class-name= valueを使用します。また、DHCP サーバーに対して、検証クライアント名-as-mac属性が無効になっていることを確認します。

スコープとプレフィックスの選択タグの設定

クライアントを異なるアドレス プールに割り当てるには、クライアント クラスの選択基準で指定した選択タグを使用して、DHCPv4 スコープ (またはテンプレート) または DHCPv6 プレフィックス(またはテンプレート)を定義する必要があります。スコープまたはプレフィックスに追加のタグがある場合でも、クライアント クラスに含まれるすべての選択基準タグは、スコープまたはプレフィックスが持つタグと一致する必要があります。クライアント・クラスがすべての選択基準を省略した場合、スコープ選択または接頭部選択に制限は適用されません。

次に例を示します。

スコープ A にはタグ 1、タグ 2 があります

スコープ B にはタグ 3、タグ 4 があります

両方のスコープが同じネットワーク上にある場合、クライアントクラスのクライアントは次の情報を持ちます。

- Tag1、tag2、またはその両方がスコープ A からリースを取得します。
- Tag3、tag4、またはその両方がスコープ B からリースを取得します。
- 両方のスコープ (tag1 や tag3 など) の 1 つ以上のタグがどちらのスコープからもリースを取得しません。
- どちらのスコープからもリースを取得するタグはありません。

次の表に、Cisco Prime Network レジストラーがネットワーク オブジェクトの選択タグまたは選択基準を参照するために使用する属性を示します。

表 *39 :* 使用する選択タグと基準属性

オブジェクト	属性（Attribute）
クライアント	selection-criteria
クライアントクラス	selection-criteria
範囲	selection-tag-list
スコープ テンプレート	selection-tag-list
[プレフィックス（Prefix）]	selection-tags
接頭語テンプレート	selection-tags
アドレス ブロック	selection-tags
サブネット（Subnets）	selection-tags

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

範囲あるいはプレフィックスまたはそのテンプレートを作成または編集します。範囲あるいはプレフィックス（またはそのテンプレート）の[追加（Add）]ページまたは[編集（Edit）]ページで selection-tags 属性を見つけ、この範囲あるいはプレフィックス（またはそのテンプレート）に関連付けるクライアントクラスの selection-criteria 属性で作成した一連のカンマ区切りの選択タグを入力します。その後、必要に応じて変更を保存し、DHCP サーバーを再ロードします。

CLI コマンド

scope 名前 set selection-tag-listを使用します。スコープ テンプレートの場合はscope-template、namesetselection-tag-listを使用します。プレフィックスの場合は、 prefix name set selection-tagsを

使用します。プレフィックス テンプレートの場合はprefix-template、 namesetselection-tagsを使用します。

クライアントクラス ホスト名プロパティの定義

クライアントクラスのホスト名(host-name)属性を使用して、各クライアントが採用するホスト名を指定できます。これは、DHCP クライアント要求に含まれるものを上書きする絶対の有効な DNS 値、または次のいずれかです。

- ***@host-name-option***-サーバーは、クライアントが送信したホスト名オプションを使用します。
- ***@no*ホスト名*-option***-サーバーはクライアントが送信するホスト名を無視します。DNS 名の生成が有効な場合、サーバーは、動的 DNS 更新用にそのように設定されている場合、生成された名前を使用します。
- ***@use-macaddress***:サーバーはクライアントのMACアドレスからホスト名を合成し、オクテットをハイフネーション処理してから、x前面にaを追加します。たとえば、クライアント MAC アドレスが 1,6:00:d0:ba:d3:bd:3b の場合、合成されたホスト名は x1-6-00-d0-ba-d3-bd-3b になります。

値を省略すると、ホスト名は指定されません。DNS 更新の構成を使用してホスト名を合成することもできます (DNS 更新設定の作成 （317 ページ）を参照)。

関連項目

クライアントと組み込みポリシーの編集 （363 ページ）

外部ソースを含むクライアント データの処理 （357 ページ）

クライアントクラスのトラブルシューティング （360 ページ）

オプション 82 を使用したサブスクライバの制限 （367 ページ）

LDAP を使用するように Cisco Prime Network Registrar を設定する （372 ページ）

クライアントクラスとその埋め込みポリシーの編集

クライアントクラスの編集には、クライアントクラスを作成するのと同じ属性が含まれます。また、クライアントクラスの埋め込みポリシーを追加および変更して、そのポリシーオプションを設定することもできます。埋め込みポリシーには、追加するまでプロパティやDHCP オプションは関連付けされません。(も参照組み込みポリシーの作成と編集（210 ページ）してください)。クライアントクラスの埋め込みポリシー設定は、DHCP サーバーがポリシー選択で使用する 3 番目の優先度であり、クライアントDHCPv4 ポリシー階層 （202 ページ）自体に設定された後です (「」を参照)。

ローカル アドバンスド Web UI

ステップ 1 クライアント クラスを作成します。

ステップ 2 左側の [クライアント クラス] ペインでクライアント クラスを選択して、[DHCP クライアント クラスの編集] ページを開きます。

ステップ 3 必要に応じて属性設定を変更します。

ステップ 4 クライアント クラスに新しい埋め込みポリシーを追加Create New Embedded Policyするには、 をクリックします。編集する既存の埋め込みポリシーがある場合は、Edit Existing Embedded Policyをクリックします。(既存の埋め込みポリシーの設定を解除するUnset場合は、[DHCP クライアント クラスの編集]Create New Embedded Policyページをクリックします。

a) このページのフィールド、オプション、属性を変更します。たとえば、[DHCPv4 オプション] の下で、ドロップダウン リストからdhcp-lease-time [51][リース期間] を選択してクライアントリース期間を設定し、[値] フィールドAdd Optionにリース間隔の値を入力して、 をクリックします。必要に応じて、属性値を設定解除します。

ステップ 5 [保存（Save）] をクリックします。

CLI コマンド

クライアント クラスに既に設定されている埋め込みポリシー値があるかどうかを確認するには、 client-class-policy client-class-name showを使用します。組み込みポリシーの属性を設定するにはclient-class-policy、クライアント クラス名set属性=valueを使用します。

DHCP オプションを設定するには、次のいずれかのコマンドを使用します。

```
nrcmd> client-class-policy client-class-name setOption {opt-name | id} value [-blob]
[-roundrobin]
nrcmd> client-class-policy client-class-name setV6Option {opt-name | id}[.instance] value
 [-blob] [-roundrobin]
nrcmd> client-class-policy client-class-name setVendorOption {opt-name | id} opt-set-name
 value [-blob]
nrcmd> client-class-policy client-class-name setV6VendorOption {opt-name | id} opt-set-name
 value [-blob]
```

リース時間を設定するには、クライアントclient-class-policy クラス名 setLeaseTimeの値を使用します。

外部ソースを含むクライアント データの処理

DHCPクライアントを実行しているネットワーク ホストとそのユーザーに関する情報は、複数の外部ソースから DHCP サーバーに到着できます。サーバーは、クライアント クラスの処理の一部としてこのデータを使用し、リースデータベースにキャプチャして、Cisco Prime Network レジストラー管理システムで使用できるようにします。

最近導入された外部要因は、クライアントの定義に影響を与える可能性があります。

- リレー エージェント情報DHCP オプション (82) のサブスクライバ IDサブオプションは、ネットワーク管理者がネットワークの加入者またはクライアントを定義し、このデータを DHCP サーバーに送信します。

- RADIUS 認証サーバーデータは、802.1x プロトコル導入の一部として使用され、RADIUS データは DHCP の意思決定に役立ちます。この場合、デバイスは、リレー エージェント情報DHCP オプション (82) のradius 属性サブオプション属性の一部としてデータを送信できます。

これらの外部オプションはどちらも DHCP オプション 82オプション 82 を使用したサブスクライバの制限 （367 ページ）を使用します (を参照)。RADIUS ソースは、次の属性を終了できます。

- クライアントユーザー名またはアカウント名（user 属性）
- 管理上定義されたクラス文字列（class 属性）
- ベンダー固有データ（vendor-specific 属性）
- セッションタイムアウト値（session-timeout 属性）
- クライアントに使用する IP アドレス プール（framed-pool 属性）
- クライアントに使用する IPv6 アドレスプール（framed-ipv6-pool 属性）

Cisco Prime Network Registrar は、 subscriber-id サブオプション、 RADIUS サブオプションの user、class、および framed-pool 属性の拡張サポートとすべてのサブオプションの式のサポートを提供します（式の使用方法 （389 ページ） を参照）。さらに、DHCP サーバーには、RADIUS の class 属性と framed-pool 属性をサーバーが処理する方法を設定する属性設定もあります。Cisco Prime Network Registrar は、サーバー属性を使用して RADIUS 属性値を選択タグまたは client-class 名としてマッピングするか、あるいはクライアントデータベースで検出された選択タグに値を追加できます。次に例を示します。

```
nrcmd> dhcp set map-radius-class=append-to-tags
```

RADIUS などの外部リソースから決定されたクライアントクラスと選択タグの場合、処理順序はでクライアントクラス処理 （352 ページ）説明されているよりもやや複雑です。次のサブセクションを参照してください。クライアントクラス機能を使用するには、DHCP サーバークライアント クラス属性を有効にする必要があります。

関連項目

クライアントクラスを判別する処理順序 （358 ページ）

選択タグを判別する処理順序 （359 ページ）

クライアントクラスを判別する処理順序

DHCP サーバーがクライアントクラス名を決定するために使用可能なソースを使用する順序は次のとおりです。

1. 拡張環境ディクショナリでクライアント クラス名を使用します。
2. データベース内に実際のクライアントエントリが見つかった場合は、そのクライアントクラス名を使用します。(データベース内のクライアントを検索する必要がなくなると思われる場合は、スキップ クライアントルックアップDHCP サーバー属性を有効にすることで、

データベース検索クライアントクラスのクライアントエントリのスキップ（365ページ）を回避できます。

3. RADIUS フレーム プール値をクライアント クラスにマップする場合(を使用dhcp set map-radius-pool-name=map-as-class) は、フレーム プールの値を使用します。
4. RADIUS クラスの値をクライアント クラスにマップする場合(dhcp set map-radius-class=map-as-classを使用) は、クラス値を使用します。
5. dhcp-user-class-id DHCP オプション (77) をクライアントクラスにマップする場合(を使用dhcpsetmap-user-class-id=map-as-class) は、オプション値を使用します。(このマッピングクライアントクラスの検索式の処理（369ページ）の代わりにルックアップID式を使用することもできます。
6. マッピングまたはユーザー クラス ID が見つからず、環境ディクショナリの default-client-class-name が使用されます。
7. クライアントエントリで構成されているデフォルトクライアントクラス名またはクライアントクラスが見つからない場合は、名前が付けられたdefaultクライアントからクライアントクラス名を使用します(見つかった場合)。

選択タグを判別する処理順序

サーバーが選択タグを決定するために使用可能なソースを使用する順序(最初のnullでないソースを使用) は次のとおりです。

1. 拡張環境ディクショナリ内の選択タグ。
2. データベース内に実際のクライアントエントリが見つかった場合は、クライアントエントリ選択タグを使用します。(この不要なデータベースの読み取りを防ぐには、スキップ クライアント ルックアップクライアントクラスのクライアント エントリのスキップ（365ページ）DHCP サーバー属性を有効にします。
3. クライアント クラスの選択タグ。
4. 使用可能な RADIUS フレーム プール値をタグにマップする場合dhcp set map-radius-pool-name=map-as-tag(を使用) 、そのタグが使用されます。
5. 使用可能なRADIUSクラス値をタグにマップする場合dhcp set map-radius-class=map-as-tag(を使用) は、そのタグを使用します。
6. 使用可能なdhcp-user-class-id DHCP オプション (77) をタグ(dhcp set map-user-class-id=map-as-tag を使用) にマップする場合、そのタグが使用されます。

次に、サーバーは、次のいずれかを選択タグ (存在する場合) のリストに追加できます。

1. RADIUS フレーム プールの値が使用可能で、map-radius-pool DHCP 属性を（dhcp set map-radius-pool=append-to-tags を使用して）タグに追加するように設定されている場合は、サーバーがその属性を追加します。

2. RADIUS クラスの値が使用可能で、map-radius-class DHCP 属性を（dhcp set map-radius-class=append-to-tags を使用して）選択タグに追加するように設定されている場合は、サーバーがその属性を追加します。
3. dhcp-user-class-id が使用可能で、map-user-class-id DHCP 属性が（dhcp set map-user-class-id=append-to-tags を使用して）選択タグに追加するように設定されている場合は、サーバーがその属性を追加します。

クライアントクラスのトラブルシューティング

クライアント クラスのトラブルシューティングを行うには、Web UI の [DHCP サーバーの編集] ページのlog 設定属性を使用してクライアント クラスのログ記録を有効にするか dhcpsetlog-settings=、CLI で設定してから DHCP サーバをリロードします(段階的 dhcp 編集モードの場合)。推奨設定値は以下のとおりです。

- client-detail- クライアントクラスのクライアント検索操作の最後に、1 行のログを記録します。この行には、クライアントに対して検出されたすべてのデータと、クライアント クラスで検出されたデータが表示されます。
- client-criteria-processing- サーバーが有効なリースを見つけたり、リースが既にリースを持っているクライアントに対して引き続き許容されるかどうかを判断するために、サーバーがスコープまたはプレフィックスを調べるたびにメッセージをログに記録します。
- ldap-query-detailDHCP サーバーが LDAP サーバーへのリース状態エントリの作成を開始した場合、LDAP サーバーからの応答を受信したとき、または LDAP サーバーから結果またはエラー メッセージを取得するたびにメッセージをログに記録します。
- 問題が LDAP サーバーに関連している可能性がある場合は、LDAPの照会可能設定も有効にします。

これらのログは、次の質問に答える上で役立ちます。

- サーバーは、予期されたデータベースからクライアントエントリを読み取っていますか。

  サーバーは、LDAP または CNRDB（Cisco Prime Network Registrar 内部データベース）からクライアントエントリを読み取ることができます。クライアント詳細ログには、サーバーがクライアントエントリを読み取っている場所が示されます。

- クライアントクラスは有効になっていますか?

  有効にしても予期しない結果が得られる場合は、どのデータベースから Cisco Prime Network レジストラーサーバー読み取りクライアントであるか確認します。LDAP または CNRDB から読み取っていますか?LDAP クエリ詳細ログは、LDAP から読み取り中かどうかを示します。ない場合は、DHCP の ldap クライアントデータのプロパティを有効にします。

(注) LDAP を使用するには、照会用に LDAP サーバーを構成する必要があります。LDAPの照会可能属性を有効にします。また、クエリにLDAPを使用するようにDHCPサーバーを構成する必要があります。

- サーバーがクライアントに適切なデータを提供していますが、そのデータから誤った結果が見られる (クライアントが予期した IP アドレスを受信していないなど)。

  ネットワーク上の明示的な関係を確認します。クライアント基準処理ログは、サーバーがアドレスを取得しているスコープまたはプレフィックスを示します。予期されるソースからアドレスを取得しない場合は、明示的な関係が正しく定義されていない可能性があります。2 次スコープであると考えたスコープは、そのように定義されていない可能性があります。

- エキスパートモードで、選択タグの包含基準と除外基準を適切に設定しましたか?

  一連の選択タグを定義して含める場合、スコープまたはプレフィックスのタグはクライアントのタグと一致する必要があります。エキスパートモードでは、クライアントクラスで選択基準除外属性を使用して、選択タグを除外することもできます。除外する系列を定義する場合、スコープまたはプレフィックスには、クライアントが構成パラメーターを取得できるように、これらのタグを定義する必要があります。選択タグの操作を開始するときに、複雑な包含および除外のシナリオを避けます。

クライアントの設定

DHCPクライアントのプロパティには、参加するクライアントクラスとクライアントに関連付けられたポリシー、実行するアクション、および選択タグの包含と除外の基準が含まれます。クライアントは、クライアント クラスからプロパティを継承します。

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

ステップ 1 メニューからDesignClients[DHCP **設定**]サブメニューの下で[DHCP クライアントの一覧/追加]ページを開きます。

ステップ 2 [クライアント] ウィンドウの [**クライアントの追加**] アイコンをクリックして [DHCP クライアントの追加] ダイアログ ボックスを開き、クライアント ID (通常は MAC アドレス) を入力しますが、DUID またはルックアップ キーを指定することもできます。(サーバー属性の検証-クライアント名-as-macを有効にすることで、クライアント名を MAC アドレスとして検証するように DHCP サーバーを設定できます。

特定のクライアント構成を持たないdefault名前のクライアントを作成することもできます。たとえば、クライアントが常にそのホスト名に MAC アドレスを使用できます。

ステップ 3 必要に応じて、定義済みのクライアント クラスのドロップダウン リストからクライアント クラス名を選択します。

ステップ 4 Add DHCP Client をクリックします。[DHCP クライアントの編集] ページが開きます。

クライアントを作成する際の重要なステップは、スコープまたはプレフィックスにクライアントを関連付けることができるように、選択基準を定義することです(クライアントに関連付けられたクライアントクラスに対して選択基準が既に設定されている場合を除く)。

[属性] リストの下にある選択基準属性を使用表 39 : 使用する選択タグと基準属性 （355 ページ）します (参照)。複数の選択タグをコンマで区切って入力できます。値は、目的のスコープまたはプレフィックスに設定された選択タグと一致するスコープとプレフィックスの選択タグの設定 （354 ページ）必要があります (を参照)。

（注） クライアントにクライアントクラスを選択した場合、このページは表示されず、クライアント名はリスト/クライアントの追加ページに表示されます。

ステップ 5 必要に応じて、他の属性を設定します。次に例を示します。

- host-name属性を@noホスト名オプションに設定して、不明なクライアントに仮のアドレスを提供します。
- 動的 DNS 更新を実行するときに使用するゾーンのドメイン名を設定します。
- クライアントのポリシーとアクションを設定します。excludeアクションを使用すると、サーバーはこのクライアントからのすべての通信を無視します（パケットは表示されません）。
- 認証の有効期限を示す時間単位 (秒、分、時間、日、週)、または UNIX スタイルの日付 (2002 年 3 月 24 日 12:00:00 など) を選択するか、または を使用しますforever。

ステップ 6 ページの一番下にある Save をクリックします。

ステップ 7 必要に応じてデバッグします。クライアント エラーをデバッグするには、DHCP ログ設定client-criteria-processingを に設定します。

ステップ 8 クライアントを削除するには、左側の [クライアント] ウィンドウの [**クライアントの削除**] アイコンをクリックし、削除を確認します。

CLI コマンド

クライアントを作成するにはclient 、name createを使用します。クライアント クラスをクライアントに関連付けるにはclient 、name set client-class-name=valueを使用します。スコープまたはプレフィックスの選択基準を設定するには、 client name set selection-criteriaを使用します。その他の属性を設定clientするには、名前set属性=値を使用します。

クライアントのプロパティを表示するにはclient 、name [ ]showを使用します。すべてのクライアントのプロパティを表示するには、client listを使用client listnamesするか、名前だけを一覧表示します。クライアントをデバッグするには、dhcp set log-settings=client-detailを使用します。クライアントを削除するにはclient 、name deleteを使用します。

関連項目

クライアントと組み込みポリシーの編集 （363 ページ）
Windows クライアント のプロパティの設定 （364 ページ）
クライアントクラスのクライアント エントリのスキップ （365 ページ）
クライアント認証の制限 （365 ページ）
クライアントのキャッシュ パラメータの設定 （366 ページ）

クライアントと組み込みポリシーの編集

クライアントの編集には、クライアントの作成と同じ属性が含まれます。また、クライアントの埋め込みポリシーを追加および変更して、ポリシーオプションを設定することもできます。埋め込みポリシーには、それを追加するまで、プロパティやDHCPオプションが関連付けられません。（組み込みポリシーの作成と編集（210ページ）も参照してください）。クライアントの埋め込みポリシー設定は、DHCP サーバーがポリシー選択で使用するDHCPv4 ポリシー階層 （202 ページ）最優先の優先順位です (を参照) 。

[ローカル基本（Basic）]または[アドバンスド（Advanced）]Web UI

ステップ 1 クライアントを作成します。

ステップ 2 [DHCP クライアントの一覧/追加] ページの [クライアント] ペインからクライアントを選択し、[DHCP クライアントの編集] ページを開きます。

ステップ 3 必要に応じて属性設定を変更します。

ステップ 4 クライアント クラスに新しい埋め込みポリシーを追加Create New Embedded Policyするには、 をクリックします。編集する既存の埋め込みポリシーがある場合は、Edit Existing Embedded Policyをクリックします。どちらの操作でも、[クライアントのDHCP埋め込みポリシーの編集]ページが開きます。(このページは、[クライアント クラスの DHCP 埋め込みポリシーの編集] ページとほぼ同じです)。

a) [クライアント用 DHCP 組み込みポリシーの編集] ページのフィールド、オプション、および属性を変更します。たとえば、[DHCPv4 オプション] の下で、ドロップダウン リストからdhcp-lease-time [51][リース期間] を選択してクライアントリース期間を設定し、[値] フィールドAdd Optionにリース間隔の値を入力して、 をクリックします。必要に応じて、属性値を設定解除します。

既存の埋め込みポリシーを設定解除する場合Unsetは、[DHCP クライアントの編集] ページをクリックします。これにより、ボタンが にCreate New Embedded Policyリセットされます。

ステップ 5 [保存（Save）] をクリックします。

CLI コマンド

クライアントに対して既に設定されている埋め込みポリシー値があるかどうかを確認client-policyするには、 client-nameshowを使用します。埋め込みポリシーを作成client-policyするには、クライアント名set属性=値を使用します。

これらの DHCP オプションを設定するには、次のコマンドのいずれかを使用します。

```
nrcmd> client-policy client-name setOption <opt-name | id> value [-blob] [-roundrobin]
nrcmd> client-policy client-name setV6Option <opt-name | id>[.instance] value [-blob]
[-roundrobin]
nrcmd> client-policy client-name setVendorOption <opt-name | id> opt-set-name value
[-blob]
nrcmd> client-policy client-name setV6VendorOption <opt-name | id> opt-set-name value
[-blob]
```

リース時間を設定するにはclient-policy、クライアント名setLeaseTimeの値を使用します。

DHCPv6 クライアントの設定

DHCPv6 クライアントを構成できます。

ローカル アドバンスド Web UI

メニューからDesignサブClientsメニューの下をDHCPSettings選択して、[DHCP クライアントの一覧/追加]ページを開きます。既存のクライアントを選択して[DHCP クライアントの編集]Add Clientsページを開くか、[クライアント] ペインのアイコンをクリックして新しいクライアントクラスを追加し、設定された DHCPv6 属性DHCPv6 クライアントクラスの設定 （354 ページ）を含むクライアント クラスを選択します (を参照Save) 。

ヒント　DHCP サーバーの検証クライアント名-mac属性を無効にします。

CLI コマンド

既存clientlistのclientクライアントを表示するには、または名前showを使用します。クライアントのクライアント クラス名を設定するにはclient 、name set client-class-name= valueを使用します。また、DHCP サーバーに対して、検証クライアント名-as-mac属性が無効になっていることを確認します。

Windows クライアント のプロパティの設定

Windows クライアントは、クラスベースのプロビジョニングをサポートします。クライアントクラスの処理に関連する特定のプロパティを設定できます。次のものがあります。

- クライアント・クラス処理のデフォルト・クライアントを判別するために、クライアント・エントリーを検索します。
- ユーザー・クラス ID をクライアント・クラスまたは選択タグにマップします。
- 選択タグ名にクラス ID を追加するかどうかを設定します。

Windows クライアントの設定

Windows クライアント ホストで、ipconfig /setclassidクラス ID を設定します。このクライアント ID をクライアント クラスまたは選択タグにマップする場合は、同じ名前を持つ必要があります。次に、ipconfig /showclassidを使用して確認します。次に例を示します。

```
DOS> ipconfig /setclassid adapter engineering

DOS> ipconfig /showclassid adapter
```

DHCP サーバーの設定

DHCP サーバーで Windows クライアントのプロパティを設定する必要があります。

ローカル・クラスターのWeb dhcp set UI または CLI のコマンド属性でDHCP サーバー属性を使用して、サーバーの Windows クライアント・プロパティーを設定します。スキップ クライアントルックアップ属性を true に設定した場合 (デフォルトは false)、DHCP サーバーはクライアントクラス処理のためにクライアントエントリをスキップします。（クライアントクラスのクライアント エントリのスキップ （365 ページ） を参照）。マップ ユーザークラス ID属性設定のいずれかを使用します。

- 0- ユーザー クラス ID を無視します(デフォルト)。
- 1- ユーザークラス ID を選択タグにマップします。
- 2- ユーザー クラス ID をクライアント クラスにマップします。
- 3- 選択タグのリストにユーザー クラス ID を追加します。

クライアントクラスのクライアント エントリのスキップ

不要なデータベースの読み取りを防ぐために、クライアント クラスのクライアント エントリを優先する必要はありません。これを実現するには、スキップクライアントルックアップDHCP サーバー属性dhcp enable skip-client-lookup (CLI) を有効にします。

クライアント認証の制限

デフォルトでは、クライアントエントリは無制限の認証を取得します。authenticate-until属性を使用すると、有効期限を指定してクライアント・エントリーの認証を制限できます。

クライアント エントリが認証されなくなった場合、DHCP サーバーは、この DHCP 要求の応答に使用するクライアント クラス エントリの名前に、認証されていないクライアント クラス名属性値を使用します。この属性が設定されていない場合、またはクライアント クラスのエントリが存在しない場合、DHCP サーバーは要求を無視します。

有効なクライアント認証値は次のとおりです。

- —num が 10 進数で、単位が秒、分、時間、日、週の場合は、以降の時間です。 +num unit たとえば、"+3w" は 3 週間後です。
- date—月、日、24 時間、2 桁または 4 桁の年。たとえば、「2002年6月30日20:00:00」とします。ローカル プロセス時間を入力します。サーバーが別のタイム ゾーンで実行されている場合は、タイム ゾーンを無視して、代わりにローカル時刻を使用します。

• forever— このクライアントの認証を期限切れにしません。

認証対象のクライアントと認証されていないクライアントを区別するために、authenticate-until属性を使用する例を次に示します。認証の期限が切れ、クライアントが別のアドレスを要求すると、DHCPサーバーはクライアントに認証されていないスコープ範囲のアドレスを割り当てます。

ステップ 1 認証済みおよび認証されていないクライアント クラスを作成します。必要に応じて、それぞれの選択基準を設定します。

ステップ 2 クライアントを作成し、認証期限の有効期限を含めます。必要に応じて、クライアント クラス名属性と認証されていないクライアント クラス名属性を設定します。

ステップ 3 認証されたスコープと認証されていないスコープを作成し、アドレス範囲を定義し、それぞれの選択タグに結び付けます。

ステップ 4 サーバーのクライアントクラス処理を有効にします。

ステップ 5 必要に応じて、DHCP サーバーをリロードします。

クライアントのキャッシュ パラメータの設定

DHCPサーバーからのアドレスに対するクライアントからの最初の要求は、多くの場合、DHCPディスカバー-DHCPオファー-DHCP要求-DHCPACK サイクルを通過します。このプロセスでは、サーバーがクライアント データの要求ごとにデータベースを 2 回調べなければなりません。クライアント キャッシュ パラメータが設定されている場合、DHCP サーバーはクライアントデータをメモリにキャッシュして、データベースを1回だけ参照する必要があります。クライアント・キャッシングを使用すると、クライアント情報をLDAPに保管するシステムのパフォーマンスが大幅に向上します。適用可能な属性を設定解除しない限り、クライアントキャッシュは既定で有効になっています。

クライアント要求の予想レートに基づいて、最大キャッシュ数と存続時間(TTL)パラメータを調整できます。要求の猛攻撃が予想される場合は、使用可能なメモリに基づいてキャッシュ数を上限まで増やしたい場合があります。要求サイクルが長くなると予想される場合は、TTLを増やしてください。目的は、要求サイクル中にサーバーがクライアントキャッシュを1回参照するようにすることです。

サーバーがクライアントキャッシュに保持するエントリ数の制限を設定するには、[DHCPサーバーの編集] ページdhcpsetclient-cache-countまたはCLIでクライアントキャッシュカウント属性を使用します。デフォルトでは、キャッシュする最大数は1000クライアントです。キャッシュを無効にするには、属性を 0 に設定します。

通常、クライアントキャッシュはキャッシュ TTL と呼ばれる 10 秒間だけ有効です。TTL の有効期限が切れると、サーバーは必要に応じてデータベースからクライアント情報を読み取ります。TTL は、[DHCP サーバーの編集] ページdhcpsetclient-cache-ttlまたはCLIのクライアントキャッシュ ttl属性を使用して調整できます。

クライアント キャッシュ数が指定された最大値に達すると、クライアント エントリ TTL が期限切れになるまで、サーバーはクライアントをキャッシュできません。

DHCP サーバーは、デフォルトでは、DISCOVER メッセージの処理中にのみクライアントデータをキャッシュします。REQUEST（更新またはリバインド）メッセージ中にクライアントデータをキャッシュする場合は、cache-client-for-requests 属性を true に設定する必要があります。この属性は、[DHCP サーバーの編集] ページで設定するか、または CLI で DHCP セットの**キャッシュ クライアントの要求を**使用して設定できます。この属性は、2 つの REQUEST (リニューアルまたは再バインド) メッセージ間の存続期間がキャッシュ TTL より短い場合にのみ true に設定する必要があります。

オプション 82 を使用したサブスクライバの制限

多くの場合、サービス プロバイダは、DHCP サーバーが顧客の設置型のデバイスに提供する IP アドレスの数を制限します。これらのデバイスは、DHCP サーバーが提供する"実アドレス"を持ち、その数を制限することを望んでいます。1 つの方法は、クライアント クラスを使用して各顧客デバイスを登録 (またはプロビジョニング) して、サーバーがクライアント/エントリ データベースに登録されているデバイスにのみ IP アドレスを発行するようにすることです。このアプローチの主な欠点は、MAC アドレスを知る必要があるすべての顧客デバイスを登録する必要があります。サービス プロバイダは、各デバイスについて知りたいとは思わないが、顧客ごとにデバイスの数が多すぎるという点が多すぎるという点が多い。

別のアプローチは、DHCP リレー エージェントが DHCPDISCOVER メッセージで送信するリレー エージェント情報DHCP オプション (RFC 3046 で説明されているオプション 82) の値に基づいて、加入者ごとに顧客デバイスを制限することです。このオプションには、お客様のデバイスが接続されているスイッチのポートに関するデータが含まれます。ケーブル モデム シナリオでは、オプション 82 サブオプションの 1 つに、通常、DHCP 要求がケーブル モデムの外に接続されたデバイスから来る場合、ケーブル モデムの MAC アドレスが含まれています。一般に、オプション 82 データを生成する多くのデバイスは、サブオプションに値を置き、その値が同じアップストリーム デバイス上のサブスクライバごとに変化します。場合によっては、この値は、すべての可能なサブスクライバ(ケーブル モデムの MAC アドレスなど)で一意です。その他の場合は、スイッチ上のポートになることができ、そのスイッチに接続されている他のサブスクライバ全体で固有のポートになります。ただし、スイッチ上のすべてのサブスクライバで一意であるとはいえない場合があります。

この方法を使用すると、ネットワーク管理者は、他の DHCP サーバーの機能に重大な影響を与えることなく、DHCP 割り当てアドレスの加入者の使用に関する制限を構成できます。多くの環境では、ネットワーク管理者は、デバイスのクラスによってはオプション 82 制限を使用し、他のクラスには使用しない場合があります。このサポートの重要な側面は、ネットワーク管理者がオプション 82 制限を使用するデバイスと使用しないデバイスを分離できるようにすることです。

関連項目

サブスクライバ制限への全般的なアプローチ （368 ページ）
一般的な制限シナリオ （368 ページ）
クライアントクラスの計算とキーの作成 （369 ページ）

サブスクライバ制限への全般的なアプローチ

クライアント処理の現在のアプローチは、クライアントエントリデータベース内のすべてのクライアントを検索することです。オプション 82 制限の目標の 1 つは、クライアント・エントリー・データベース (CNRDB または LDAP のいずれか) 内のすべての顧客デバイスを明示的に登録(プロビジョニング)する必要性を取り除く方法です。ただし、サブスクライバーが制限されている特定の番号を構成し、すべての未登録のサブスクライバーに与えられた既定の番号を上書きする必要があります。

（注） DHCPv6 クライアントでは、制限処理は現在利用できません。

大まかに言えば、サーバーが各着信パケットについて評価し、クライアントを行くクライアントクラスの名前を返す式を作成することによって、加入者制限を設定できます (式の使用式の使用方法（389ページ）の詳細については、「」を参照)。各クライアントクラスは、制限識別子(ID)、サーバーが着信パケットから決定し、実際にデバイスの数を制限するために後の処理で使用するキーの指定を可能にします。サーバーは、同じ制限 ID (制限 idプロパティ) を持つすべてのデバイスが同じサブスクライバーから取得されるとみなします。

一般的な制限シナリオ

たとえば、着信パケットは次のように評価されます。

1. オプション 82 のremote-idサブオプションがクライアントのハードウェア アドレス (chaddr) と一致するcm-client-class場合、サブスクライバはケーブル モデムであり、 .
2. dhcp クラス識別子オプションの最初の 6 バイトが文字列docsisに一致する場合、サブスクライバは DOCSIS モデムdocsis-cm-client-classであり、 .
3. ユーザークラスオプションの値が文字列alternative-classと一致する場合は、サブスクライバーalternative-cm-client-classが に含まれる必要があります。

クライアントクラスの計算とキーの作成

DHCP サーバーのクライアント クラス-lookup-id属性、またはdhcpsetclient-class-lookup-id=CLI の式のクライアントクラスを決定する式を設定します。属性定義で参照されるファイルに、属性定義に単純式を含めるか、より複雑な式式の使用方法 （389 ページ）を含める (を参照)。

クライアントとクライアントクラスでは、クライアントまたはクライアントクラスに対して制限 ID値を指定することもできます。サーバーはこの ID 値を使用して、同じネットワークまたはLAN セグメント上で同一 ID を持つデバイスの数に対するアドレス制限を設定します。要求側のクライアントがその ID に対して使用可能なアドレスの制限を超える場合、サーバーはそれを制限超過クライアント・クラス名(設定されている場合)に割り当てます。それ以外の場合は、パケットをドロップします。制限 IDは、実質的に、サブスクライバーを定義します。

クライアントクラスの検索式の処理

最初のクライアントクラスルックアップでは、クライアントが何らかの制限に参加するかどうかを決定できます。クライアント クラス検索 ID属性を使用して、式サーバー全体を構成します。サーバーは、パケットのクライアントクラスを決定することを目的として、すべての着信パケットに対してこの式を実行します。

この式は、パケットのクライアントクラス名である文字列、またはクライアント要求に対してクライアント クラスの値が考慮されなくなったことを示す識別文字列 <none> を返す必要があります。<none> 文字列を返すことは、クライアント クラスルックアップ ID値を構成しないことと同じであり、クライアント クラスの処理は行われません。式が null を返すか、クライアントクラスルックアップ IDを評価するエラーが発生した場合、サーバーはパケットを (付随するログ メッセージとともに) ドロップします。

制限の処理

DHCP サーバーは、同じネットワークまたは LAN セグメント内で同じ制限 ID値を持つ DHCP クライアントに割り当てられる IP アドレスの数を制限します。サーバーがクライアントに別のアドレスを割り当てることで制限を超える場合、クライアント パケットはオーバーフロー クライアントクラスに配置されます(指定されている場合)。これにより、構成された制限を超えるクライアントに対して特別な処理が可能になります。これらのクライアントを何らかの自己プロビジョニング方法で処理することは、ハードウェアではなく DHCP サーバーの制限を使用する利点の 1 つです (サポートされている場合もあります)。

クライアントクラスに制限超過がない場合、サーバーはパケットをドロップし、そのパケットのアドレス割り当てがその制限idの制限カウントを超える可能性があります。サーバーは、単一のネットワークまたは LAN セグメントでのみ制限を適用します。ネットワーク マネージャは、一度に1つのLANセグメントを介して接続している1つの加入者を見る傾向があるため、これは制限ではありません。

DHCP ポリシーで、制限数を同一の制限 IDで設定します。制限コードは、他のポリシー アイテムと同様に、ポリシー階層の制限数を検索します。つまり、クライアントクラスの埋め込み

ポリシーまたは名前付きポリシー、スコープの埋め込みまたは名前付きポリシー、またはシステムsystem_default_policyで制限カウントを構成できます。

クライアントクラスで制限 IDを設定すると、クライアントクラスの制限処理を追及するように合図されます。制限 IDを設定しない場合は、それを追求しないように信号を送ります。式を実行して制限 idを判別する場合、式が null を返す場合、このシグナルは、制限処理が行われ、リース状態データベースに保存されている制限 idを使用する必要があります。

サブスクライバ制限のための式処理

式は、制限処理の複数の場所に存在します。各式は、null または文字列 (通常はクライアントクラスを検索するときにクライアント クラス名を決定する) または制限 idを作成するときに一連のバイト (BLOB) に評価されます。式は、次の場所で使用できます。

- クライアントクラスの検索
- 同じサブスクライバのクライアントを制限するキーの作成 (制限 id)
- クライアント・エントリー・データベース (クライアント・ルックアップ ID)で検索するキーを作成する。

オプション 82 制限の設定

ステップ 1 クライアントを明示的に登録しない場合は、オプション 82 データを使用する場合は、DHCP サーバー プロパティとしてクライアント クラスを有効にしないでください。

ステップ 2 クライアントの数を制限し、他のクライアントを制限しないかを決定します。一部のクライアントを制限する場合は、次の手順を実行します。

a) 各クラスのクライアントからの DHCP 要求に含まれる値に基づいて、これらのクライアントを他のクライアントと区別する方法を見つけます。

b) 制限のないクライアントを配置するクライアント クラスの名前と、これらの無制限のクライアントに使用する選択タグとスコープを決定します。

ステップ 3 制限超過のクライアントを別のクライアント クラスに配置するか、単にパケットをドロップするかを決定します。クライアント クラスを制限超過にする場合は、クライアント クラス名と、超過クライアントを配置する範囲と選択タグとスコープを決定します。

ステップ 4 制限するクライアントを配置するクライアント クラスと、これらのクライアントに使用する選択タグとスコープを決定します。

ステップ 5 これらすべての選択タグ、クライアントクラス、およびスコープを作成します。

ステップ 6 ポリシー内の制限カウント(クライアントクラスに関連付けられた名前付きポリシー) を構成して、クライアントが制限する。

ステップ 7 入力するクライアントを制限するクライアントと制限されないクライアントに分離する式を記述します。クライアント クラス検索 ID属性を設定して、DHCP サーバー上で構成します。

ステップ 8 制限するデバイスの制限 ID を決定する式を記述し、クライアント クラスで制限id を設定して制限するようにクライアント クラスで構成します。

オプション 82 制限のリース更新処理

DHCP クライアントがブロードキャストするパケットのみが、オプション 82 データが付加されたサーバーに到着します。BOOTP または DHCP リレーエージェントは、クライアントデバイスから最初のアップストリーム ルータにオプション 82 データを追加します。DHCPRENEW パケットはサーバーにユニキャストされ、オプション 82 データなしで到着します。これにより、サブスクライバの制限をサーバーに構成するときに問題が発生する可能性があります。

更新を処理する場合、一般的に 2 つの方法があります。

- オプション 82 データを持たないパケットはすべて、関連する選択タグのないクライアントクラスに配置します。これはワイルドカード選択と同等であり、オプション 82 データのないパケットは受け入れられることを意味します。
- オプション 82 データを持つパケットを配置し、その制限 idを null と評価する場合と同じクライアント クラスに DHCPRENEW を配置します。これは、制限をチェックする際に、パケットから 1 つではなく、以前に保存された制限 IDを DHCP サーバーが使用する必要があるというシグナルです。

どちらのアプローチも機能します。2つ目の方が安全ですが、実際には最初のものよりはるかに優れているわけではありません。これは、DHCP サーバーが DHCPRENEW に応答するために IP アドレスを使用する必要があり、ほとんどのクライアントはサーバーの状態の一部を失わない限り、このアドレスを使用しないためです。この場合、クライアントにアドレスを与える必要があります。悪意のあるクライアントの場合、サーバーをクライアントにアドレスを渡すためにアドレスを使用する必要があり、それによってこのケースの公開を制限します。

オプション 82 制限の管理

制限idを持つクライアント クラスに含まれるクライアントが制限に関与している場合は常に、クライアント データ ログが発生するたびに、使用される制限 ID が DHCP ログ ファイルに表示されます。LID: nnn :nnn :nnn .. データは、現在制限カウントの 1 つを占有しているアクティブなリースを持つクライアントに対してのみ記録されます。

サブネット内の制限 IDを使用して、すべてのクライアントを決定できます。[DHCP サーバーの管理] ページで、[コマンド] 列の [**実行**] アイコンをクリックして、[DHCP サーバーコマンド] ページを開きます。[IP アドレス] フィールドに、現在アクティブなリースの IP アドレスを入力してから、[**実行**] アイコンをクリックします。また、limitation-id 自体を nn:nn:nn の形式で入力するか、または文字列（"nnnn"）として入力することもできます。この場合、IP アドレスが検索対象のネットワークになります。CLI で、次dhcp limitationListを使用します。

```
nrcmd> dhcp limitationList ipaddr [limitation-id] show
```

ipaddrと制限idの両方を指定すると、サーバーは、サブネットを決定するために、giaddrと同じようにそれを使用します。ネットワークのスコープ(プライマリまたはセカンダリ)に表示される可能性のある任意の IP アドレスを使用して、サブネットを指定できます。ipaddrだけを指定する場合は、DHCPサーバーが提供するアドレスを指定する必要があり、コマンドは、すべてのクライアントと、そのクライアントが使用するリースを返します。

制限カウントのオーバーフローによりクライアントがサービスを拒否された場合、DHCPサーバーのログ ファイルに次のようなメッセージが表示されます。

```
Warning Server 0 05646 Could not add Client MAC: '1,6,01:02:03:04:0c:03'  with
limitation-id: 01:02:03 using Lease: 10.0.0.23, already 3 Clients with that id.
No over-limit client class specified! Dropping packet!
```

制限dhcpカウントlimitationListを超えて拡張されるクライアントを特定できます。コマンドのipaddr値は「リースを使用する:」値にし、制限 id はログ ファイル内の "制限 id:" 値にする必要があります。ログ・ファイルの例を使用すると、コマンドは次のようになります。

```
nrcmd> dhcp limitationList 10.0.0.23 01:02:03 show
```

オプション 82 制限のトラブルシューティング

制限サポートをデバッグする方法はいくつかあります。最初に、DHCPサーバーのデバッグ値をVX=1(またはdhcp setDebug VX=1を使用して)に設定して、パケット トレースを有効にする必要がある場合があります。(デバッグVX=0値はパケット トレースを無効にします。次に、クライアント基準処理とクライアント詳細をログ設定に追加して、クライアントクラスのデバッグを有効にする必要があります。

サーバー全体の式トレース レベル、式トレース レベルもあり、さまざまなレベルに設定できます。6 に設定すると、式の評価の詳細なトレースが表示されます。この処理は、ログに少しのスペースを要し、サーバーの速度も大幅に低下しますが、式の評価に慣れる過程で非常に貴重です。デバッグ式 （431 ページ）を参照してください。

問題が変わったように見える場合や、ログファイルを送信して問題を報告する場合は、DHCPサーバーのデバッグ値をQR57=9(dhcpsetDebugQR57=9または を使用して) 設定して、追加のトレースを有効にすることが重要です。(デバッグQR57=0値はこのトレースを無効にします)。Q と R はどちらも大文字であることに注意してください。Q はクライアントクラスのデバッグで、R は応答デバッグです(ログ内の制御フローをクリアするために必要)。5は式処理であり、7 はクライアント・クラス・ルックアップ処理です。これにより、パケットごとに 1 ページほどの出力が生成され、サーバー内で何が起こっているのかを理解するのに役立ちます。

式の例

式を使用して、サブスクライバーにリースされる IP アドレスを制限する （427 ページ） を参照してください。

LDAP を使用するように Cisco Prime Network Registrar を設定する

ライトウェイト ディレクトリ アクセス プロトコル(LDAP)は、Cisco Prime Network レジストラークライアントとリース情報を統合するためのディレクトリ サービスを提供します。LDAPディレクトリに格納されているオブジェクトの既存の標準スキーマを構築することで、DHCP

クライアントエントリに関する情報を処理できます。したがって、DHCPサーバーデータベース内のクライアント情報を維持する代わりに、Cisco Prime Network レジストラー DHCP サーバーに対して、DHCP クライアント要求に応答するデータのクエリを 1 つ以上の LDAP サーバーに発行してもらうか、リース データを LDAP サーバーに書き込むことができます。

Cisco Prime Network Registrar は、Linux で使用可能な OpenLDAP クライアントを使用します。

関連項目

LDAP ディレクトリ サーバーについて （373 ページ）

LDAP リモート サーバーの追加と編集 （373 ページ）

LDAP での DHCP クライアント クエリの設定 （374 ページ）

DHCP LDAP 更新とサービスの作成の設定 （379 ページ）

LDAP のトラブルシューティング （385 ページ）

LDAP ディレクトリ サーバーについて

LDAP ディレクトリ サーバーは、属性/値ペアのコレクションに名前を付け、管理し、アクセスする方法を提供します。Cisco Prime Network レジストラーは特定の LDAP オブジェクト クラスまたはスキーマに依存しないため、LDAP サーバーに情報をいくつでも入力できます。

- DHCP クライアント情報は、使用されていない属性に格納できます。たとえば、指定された名前属性を使用して、DHCP クライアントクラス名の値を保持できます。
- LDAP スキーマ検査を無効にした場合、LDAP スキーマを変更せずに、オブジェクト・クラスに新しい属性を追加できます。たとえば、組織の人物オブジェクトクラスにクライアント クラス名属性を追加できます。
- 新しいオブジェクトクラスを作成し、適切な属性を定義できます。たとえば、DHCP クライアント オブジェクト クラスを作成し、使用するクライアント属性を定義できます。

LDAP から読み取るように DHCP サーバーを構成すると、照会辞書は照会する LDAP 属性をサーバーに指示します。サーバーは、結果のデータを DHCP クライアントデータ属性に変換します。

ヒント LDAP サーバーが応答を停止したり、DHCP サーバーからの要求に応答を再開したりしたときに SNMP トラップを生成するように Cisco Prime Network レジストラーを設定できます。

LDAP リモート サーバーの追加と編集

LDAP サービスの使用を開始するには、リモート LDAP サーバーを追加する必要があります。

ローカル アドバンスド Web UI

メニューからDeployLDAP[DHCP]サブメニューの下で[LDAPリモートサーバーのリスト/追加]ページを開きます。Add LDAP [LDAP] ペインのアイコンをクリックして、[DHCP LDAP サーバーの追加] ダイアログ ボックスを開きます。リモート サーバーを編集するには、[LDAP] ペインで LDAP を選択し、[LDAP リモート サーバーの編集] ページを開きます。

このページでは、LDAP サーバーの名前と完全修飾ドメイン名または IP アドレス (IPv4 または IPv6) を少なくとも指定する必要があります。操作を正常に実行するには、ユーザー名とパスワードが必要です。

（注） クエリ設定と作成設定は、ローカルで DHCP リースの作成に使用されます。

CLI コマンド

ldap name create domain-name を使用します。次に例を示します。

```
nrcmd> ldap ldap-1 create ldap.example.com
```

IP アドレスldap server(IPv4 または IPv6) を使用することもできます。次に例を示します。

```
nrcmd> ldap ldap-1 create 192.0.2.1
nrcmd> ldap ldap-1 create 2001:DB8:1::1
```

LDAP での DHCP クライアント クエリの設定

LDAP クライアントエントリでは、DHCP クライアントクエリの設定とプロビジョニング解除、および組み込みポリシーの設定ができます。

DHCP サーバーから LDAP へのクライアント クエリの設定

DHCP サーバーがクライアント データを LDAP サーバーに照会できるようにするには、次の手順を実行します。ローカル クライアント エントリと同様に、LDAP クライアント エントリはクライアントの MAC アドレスによってキー設定されます。

（注） LDAP サーバーに接続する場合は、ユーザーの識別名(dn) を使用します。LDAP スキーマ内のオブジェクトを一意に識別し、データベース内の一意キーまたはファイルの完全修飾パス名に似ています。たとえば、人の dn は dn: cn=ベス・ジョーンズ、ou=マーケティング、o＝サンプル・コーポレーションです。この会社には、ベスという名前の人やジョーンズという名前の人がたくさんいるかもしれませんが、ベス・ジョーンズという名前の人は他に例のコーポレーションでマーケティングで働いていません。

ステップ 1 LDAP サーバーのホスト名を指定します。[LDAP リモートサーバーの追加（Add LDAP Remote Server）] ページで、[名前（name）]フィールドに値を入力します。ローカルCLIで、次のコマンドを使用します。

```
nrcmd> ldap ldap-1 create ldap.example.com
```

後でサーバーを削除する必要がある場合はldap 、server を delete使用します。

ステップ 2 接続の認証情報を設定します。ユーザーに識別名（dn）を使用します。[ユーザー名（username）] フィールドに値を入力します。CLI で、次のコマンドを使用します。

```
nrcmd> ldap ldap=1 set username="cn=joe,o=Example Corp,c=US" password=access
```

ステップ 3 検索パス (および必要に応じて検索範囲) を設定します。パスは、検索を開始するディレクトリ内のポイントです。検索範囲が次の場合:

- SUBTREE を使用すると、サーバーは検索パスのすべての子を検索します。
- ONELEVEL を指定すると、サーバーは基本オブジェクトの直接の子のみを検索します。
- BASE の場合、サーバーはベース オブジェクト自体だけを検索します。

この例では、検索のベースを組織 Example Corp と国 US に設定し、サブツリー検索範囲を設定します。[検索パス（search-path）] フィールドに値を入力します。CLI で、次のようなコマンドを使用します。

```
nrcmd> ldap ldap-1 set search-path="o=Example Corp,c=US" search-scope=SUBTREE
```

ステップ 4 検索フィルタを、DHCP がクライアントの MAC アドレス（DHCPv4 の場合）または DUID（DHCPv6 の場合）に置き換える属性に設定します。この例では、属性は共通名 (cn) です。[検索フィルタ（search-filter）] フィールドに値を入力します。CLI で、次のようなコマンドを使用します。

```
nrcmd> ldap ldap-1 set search-filter=(cn=%s)
```

ステップ 5 LDAP と DHCP のマッピングをすべて含むクエリ ディクショナリを設定します。これらのマッピングを設定するには、ldap サーバー名 setEntryを使用します。

1. sn LDAP 属性から DHCP 姓を取得します。

   ```
   nrcmd> ldap ldap-1 setEntry query-dictionary sn=host-name
   ```

2. 最初の名前 LDAP 属性からクライアント・クラス名を取得します。

   ```
   nrcmd> ldap ldap-1 setEntry query-dictionary givenname=client-class-name
   ```

3. ローカルの LDAP属性からドメイン名を取得します。

   ```
   nrcmd> ldap ldap-1 setEntry query-dictionary localityname=domain-name
   ```

4. いずれかのエントリを設定解除する必要がある場合は、ldap サーバー unsetEntry 属性キーを使用します。また、ldap サーバーのgetEntry 属性キーを使用して、任意の設定を確認することもできます。

ステップ 6 LDAP サーバーに対する照会を使用可能にします。この例では、myserverのクエリを有効にします。can-query 属性を enabledに設定します。CLI で、次のコマンドを使用します。

```
nrcmd> ldap ldap-1 enable can-query
```

ステップ 7 DHCP サーバーのクライアントクラス処理を有効にします。[DHCP サーバーの編集（Edit DHCP Server）] ページで、 client-class 属性を enabled に設定します。CLI で、次のコマンドを使用します。

```
nrcmd> dhcp enable client-class
```

ステップ 8 DHCP サーバーがクライアント エントリクエリに LDAP を使用できるようにします。[DHCP サーバーの管理] ページで、クライアント クラス属性を有効に設定します。CLI で、次のコマンドを使用します。

```
nrcmd> dhcp enable use-ldap-client-data
```

ステップ 9 複数の LDAP サーバーを構成している場合は、ラウンドロビンモードまたはフェイルオーバーモードで動作するように設定することもできます。

- **ラウンドロビン**-LDAP サーバーのプリファレンス値は無視され、クライアント クエリを処理し、リース状態の更新を受け入れるように構成されているすべてのサーバーが等しく処理されます。
- **フェールオーバー**:DHCP サーバーは、最も優先度の高い(最も低い設定番号)のアクティブ LDAP サーバーを使用します。優先サーバーが接続を失ったり、失敗したりすると、DHCP サーバーは次の低い優先順位の LDAP サーバーを使用します (優先順位が高くなります)。設定値が同じ (または設定されていない) 場合、DHCP はこれらのサーバーとのラウンドロビン モードに戻ります。

[DHCP サーバーの編集] ページで LDAP モードを設定して、LDAP サーバー モードを設定します。LDAP フェールオーバー モードは、実際には優先的なロード バランシングを実行します。DHCP サーバーは、LDAP 接続とエラー状態、および LDAP サーバーの応答速度を評価します。最適な状態では、DHCP サーバーは、最も高い優先順位 (最も低い優先順位番号) を割り当てた LDAP サーバーを使用します。最適ではない状態では、DHCP サーバーは、次の低い優先順位の LDAP サーバーを使用します (優先順位の数が増加します)。設定値が同じ (または設定されていない) 場合、DHCP サーバーはラウンドロビン モードに戻ります。

CLI で、dhcp set ldap **モード**を使用してモードを設定し、ldap サーバーが**設定**設定してサーバーの基本設定を設定します。例えば：

```
nrcmd> dhcp set ldap-mode=failover
nrcmd> ldap ldap-1 set preference=1
nrcmd> ldap ldap-2 set preference=2
```

また、DHCP サーバーと LDAP サーバー間の接続属性 (を参照LDAP の推奨値 （386 ページ）) を使用して設定した、開いているスレッドの数によっては、DHCP サーバーは、クエリ タイムアウトが切れる前に、できるだけ多くのスレッドを開くだけであることに注意してください。LDAP サーバーがこれらのスレッドを処理している可能性がありますが、フェイルオーバー・サーバーが引き継いだため、要求を処理していません。

ステップ 10 DHCP サーバーがクライアント エントリクエリに LDAP を使用できるようにします。[DHCP サーバーの管理（Manage DHCP Server）] ページで、client-class 属性を enabled に設定します。CLI で、次のコマンドを使用します。

```
nrcmd> dhcp enable use-ldap-client-data
```

ステップ 11 LDAP 構成を表示または一覧表示します。[LDAP リモート サーバーの一覧/追加] ページに移動します。CLI で、次のコマンドを使用します。

```
nrcmd> ldap ldap-1
nrcmd> ldap list
nrcmd> ldap listnames
```

ステップ 12 DHCP サーバーをリロードします。

（注） DHCP サーバーは通常、%s をクライアントの MAC アドレス（DHCPv4 の場合）または DUID（DHCPv6 の場合）に置き換えます。ただし、他のクライアント指定子を使用できます。他のクライアント指定子（拡張によって生成されるなど）を使用する場合は、文字列を使用して LDAP インジェクションを実行できないようにしてください。これは、クライアントから送信されたデータによって次の文字が挿入されないようにするため、または、可能であれば次の文字列が適切にエスケープされるようにする必要があるためです。

カンマ（,）、バックスラッシュ文字（\）、ポンド（ハッシュ）記号（#）、プラス記号（+）、小なり記号（>）、セミコロン（;）、二重引用符（"）、等号記号（=）、および先頭または末尾のスペース

場合によっては、他の文字が問題になることもあります（LDAP サーバーまたは RFC 4514 で確認してください）。受信パケットのデータを使用する場合、問題になることがあります。DHCP サーバーでは、指定された文字列は変更されません。提供された文字列がそのまま安全に使用できることを前提としています。

クライアント エントリのプロビジョニング解除

LDAP クライアント情報が LDAP に残るように LDAP クライアントエントリをアンプロビジョニングできますが、DHCP サーバーはクライアントをその情報が存在しないものとして扱います。DHCP サーバーは、クライアントにデフォルトの動作を提供します。LDAP サーバーが値を持つ指定DHCP サーバーから LDAP へのクライアント クエリの設定（374 ページ）された属性を含むクライアントエントリを返さないように、前のセクションのステップ 4 で検索フィルタ セットを設定します。

LDAP エントリ givenname のプロビジョニングを解除する場合は、それに応じた検索フィルタを設定します。次に例を示します。

```
nrcmd> ldap ldap-1 set search-filter=(&(cn=%s)(!(givenname=unprovision)))
```

LDAP クライアント エントリの指定された名前属性が"準備解除"文字列に設定されている場合、LDAP サーバーはクライアント エントリを DHCP サーバーに返しません。つまり、DHCP サーバーは、クライアントを LDAP クライアント エントリがないかのように扱います。この手順では、DHCP サーバーまたは LDAP サーバーに対してパフォーマンスに対する測定可能な影響はありません。

LDAP での埋め込みポリシーの設定

ステップ 1 たとえば、LDAP サーバーを構成し、そのサーバーに my サーバーという名前を付けます。

ステップ 2 DHCP サーバーが組み込みポリシーとして解釈する LDAP 属性を、内部組み込みポリシー プロパティにマップします。この例では、ビジネス カテゴリ LDAP 属性をマップします。

```
nrcmd> ldap myserver setEntry query-dictionary businessCategory=embedded-policy
```

ステップ 3 DHCP サーバーが組み込みポリシーとして解釈できる LDAP 属性に文字列を追加します。この文字列の外観を決定する最も実用的な方法は、Cisco Prime Network レジストラーデータベースにダミー クライアントを作成し、クライアントの組み込みポリシー設定からデータを抽出することです。このダミー クライアントは、LDAP を使用しているため、使用されることはないので、後で削除できます。必要なオプションデータタイプを埋め込みポリシーに含めます。

1. たとえば、ダミー クライアント 1,6,00:d0:ba:d3:bd:3b 用の組み込みクライアント ポリシーを作成します。応答オプションと、IP アドレスデータタイプの複数値オプション (ルーター) を追加します。

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b create
nrcmd> client-policy 1,6,00:d0:ba:d3:bd:3b set v4-reply-options=routers
nrcmd> client-policy 1,6,00:d0:ba:d3:bd:3b setOption routers 1.2.3.4,5.6.7.8
nrcmd> save
```

2. 値を表示できるように、クライアントの埋め込みポリシー データを取得します。

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b get embedded-policy
100 Ok
embedded-policy="((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list [((ClassName
 Option)(number 3)(option-definition-set-name dhcp-config)(value
01:02:03:04:05:06:07:08))])(v4-reply-options [routers ])"
```

3. 前のサブステップのクライアント出力の引用符の間にある内容をコピーし、それを businessCategory LDAP 属性の定義に貼り付けます。

```
businessCategory:((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list [((ClassName
 Option)(number 3)(option-definition-set-name dhcp-config)(value
01:02:03:04:05:06:07:08))])(v4-reply-options [routers ])
```

4. LDAP の新しい組み込みポリシー エントリごとに、構文をモデルとして使用します。LDAP 文字列内の他のオプション データ型がどのように表示されるか確認するには、これらのオプションをクライアントに追加するか、またはクライアントと共にさらにダミー クライアントを作成します。データを抽出したら、ダミークライアントを削除できます。

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b delete
nrcmd> save
```

LDAP での組み込みポリシーの設定 (複数のオプション定義を使用)

複数のオプション定義を持つ別の例を次に示します。

ステップ 1 ダミー・クライアント 1,6,00:d0:ba:d3:bd:3b およびそのクライアントにアタッチされた埋め込みポリシーを作成します。

```
3 routers 10.1.1.1,10.2.1.1
66 tftp-server tftp-server.com
67 bootfile device-boot-file.txt
```

ステップ 2 埋め込みポリシーへの変更を保存し、クライアントを保存してから、次の出力文字列を LDAP クライアント構成に抽出します。

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b get embedded-policy
100 Ok
embedded-policy="((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list [((ClassName
```

```
Option)(number 3)(option-definition-set-name dhcp-config)(value 0a:01:01:01:0a:02:01:01))((ClassName
 Option)(number 66)(option-definition-set-name dhcp-config)(value
74:66:74:70:2d:73:65:72:76:65:72:2e:63:6f:6d))((ClassName Option)(number 67)(option-definition-set-name
 dhcp-config)(value 64:65:76:69:63:65:2d:62:6f:6f:74:2d:66:69:6c:65:2e:74:78:74))])"
```

DHCP LDAP 更新とサービスの作成の設定

Cisco プライム ネットワークレジストラー DHCP サーバーを設定して、リースおよびクライアント データを LDAP サーバーに書き込むことができます。DHCP サーバーは、クエリ構成を使用して、DHCP クライアント要求に応答するときにクライアント データを使用できます。LDAP サーバーのクライアント・オブジェクトの属性にリース状態データをコピーするように DHCP LDAP サービスを構成できます。DHCP サーバーは、リース状態データを文字列形式に変換し、更新ディクショナリを使用して DHCP データ値を LDAP 属性にマップします。

リース状態が変更されるたびに、DHCP サーバーはデータを格納するように構成した LDAP サーバーに変更を書き込みます。DHCP サーバーが LDAP に書き込むリース データは、リース状態データベース内の権限のあるデータのコピーであるという「書き込み専用」です。

関連項目

リース状態属性 （379 ページ）

LDAP にリース状態を書き込むための DHCP の設定 （380 ページ）

LDAP 更新の使用 （382 ページ）

LDAP 状態の更新の設定 （382 ページ）

LDAP エントリ作成の設定 （384 ページ）

リース状態属性

LDAP サーバーにリース状態情報に関する以下の属性を保存できます。

- address：このリースの IP アドレス。
- client-dns-name：DHCP サーバーがこのクライアントの DNS サーバーに入力しようとした名前。
- client-domain-name：クライアント名を配置するドメイン。
- client-flags：クライアントに関連するさまざまなフラグ。
- client-host-name：クライアントが DNS サーバーに配置するように DHCP サーバーに対して要求した DNS 名。
- client-id：クライアントによって指定されたクライアント ID。またはこのクライアントの DHCP サーバーによって合成されたクライアント ID。
- client-mac-addr：クライアントが DHCP サーバーに提示した MAC アドレス。

> **注** LDAP の MAC アドレスは、ローカル クライアント エントリを作成するときに Cisco Prime Network レジストラーによってフォーマットされるとおりにフォーマットする必要がありますが、それらは個別のインスタンスであり、リース データに固有です。

- expiration：リースの有効期限が切れる時刻。
- flags：リースのフラグ（reserved や deactivated）。
- lease-renewal-time：クライアントがリースの更新を発行する予定の最も早い時刻。Cisco プライムネットワーク レジストラーを使用dhcp enable save-lease-renewal-timeして、リース状態の一部として保存できます(デフォルトでは保存されません)。
- start-time-of-state：状態が現在の値に最後に変更された時刻。
- state：次のようなリース状態があります。
  - 利用可能 (1)
  - Deferred (2)
  - リース (3)
  - Expired (4)
  - Unavailable (5)
  - リリース済み (6)
  - Other_available (7)
  - Disconnected (8)
  - 削除済み (9)
- vendor-class-identifier：ベンダー固有の情報を交換するためにクライアントとサーバーが使用するベンダーの名前。

すべてのリースにこれらすべての属性があるわけではありません。クライアントがリースを解放するか、Cisco Prime Network レジストラーIP Expressを通じて強制的に利用可能にされる場合、クライアント-mac-addrおよびクライアントのリース状態属性は存在しません。また、DHCP を使用してリース更新の保存時プロパティが無効になっている場合、リース更新時間属性が存在しない場合があります。同様に、ベンダクラス識別子プロパティは、DHCP を使用して SAVE-Vendor-class-idプロパティが無効になっている場合は、CLI を使用して存在しない場合があります。

LDAP にリース状態を書き込むための DHCP の設定

DHCP 書き込みリース状態を LDAP に更新するには、次の手順を実行します。

ステップ 1 LDAP リース状態更新スキームを選択します。

ステップ 2 ディレクトリにエントリを追加するか、リース状態情報を格納する既存のエントリを変更します。属性またはカスタム オブジェクト クラスを追加してエントリを拡張する必要がある場合があります。

ステップ 3 更新を実行するには、Cisco プライムネットワーク レジストラーを設定します。

ディレクトリの柔軟性を考えると、ディレクトリにリース状態属性のコピーを格納する方法はさまざまです。たとえば、リース状態データを既存のエントリの一部として格納するか、リース状態データを個別に保存することができます。

既存のエントリの一部としてリース状態データを保存

リース状態データは、既存のエントリの一部として格納できます。クライアントエントリ、リース状態、従業員データを同じエントリに格納することも可能です。このメソッドのセットアップの一部として、リース データ属性の格納方法を決定する必要があります。データ属性は、次の方法で格納できます。

- エントリから属性をマップする
- エントリに属性を追加する
- 新しいオブジェクト クラスを作成してエントリを拡張する

利点は、リース データが他のクライアント情報と共に直接格納されるということです。欠点は、クライアント クラスや予約に関連するシナリオが存在する可能性があり、サーバーがクライアントをリースから移動するときに、ディレクトリ内に古いデータが短時間存在する可能性があることです。

（注） 更新される状態のリースにクライアントがない場合、関連付けられた MAC アドレスは存在しません。この状況は、クライアントがリースを取得し、クライアント クラスの処理によってリースから移動された場合に発生します。また、クライアントが既存のリースを持ち、同じ LAN セグメント内の別のリースの予約を行う場合にも発生します。予約済みリースが使用可能な場合、サーバーはクライアントを既存のリースから予約に移動します。これらの転送の両方は、クライアント MAC アドレスなしで古いリースの LDAP 更新になります。新しいリース (関連 MAC アドレスを持つ) の更新が行われる必要があるため、これは一般的に問題ではありません。

また、この方法では、リース情報を書き込むために2つのLDAP対話が必要です。DHCP LDAP サービスは、エントリを更新する際にエントリを見つける方法を知るだけでは不十分であるため、リース状態情報を更新する場合、ディレクトリに2回接続します。具体的には、エントリのdnを知っている必要があります。

DHCP LDAP サービスは、まず、選択したリース状態属性 (できれば MAC アドレス) を検索条件として使用して、ディレクトリ内の適切なエントリを検索します。これは、リース状態属性のいずれもエントリのdnの一部ではないため、必要です。DHCP LDAP サービスがエントリを見つけると、dnが返されます。DHCP LDAP サービスは、適切な情報を使用して同じエントリを更新します。このメソッドの使用例については、「」を参照してくださいLDAP 状態の更新の設定 （382 ページ）。

リース状態データを個別に保存

IPアドレスによってリース状態データを独自のエントリに格納できます。この方法は、ディレクトリ内のサーバー リース データベースのコピーとなり、データベースを構成する最も簡単な方法です。この方法のセットアップの一部として、サーバーがサービスを提供できる各 IP アドレスに対して新しいエントリを作成します。この方法の利点は、ディレクトリ内のリース状態データが古くなるシナリオが存在しない点です。欠点は、リースデータが他の関連するクライアント情報と直接格納されないことです。

リース状態情報を更新するには、DHCP LDAP サービスがディレクトリ サービスに 1 回接続します。更新を実行すると、サービスは IP アドレスを使用してdnを構築します。

LDAP 更新の使用

LDAP 更新機能を使用するには、次の 2 つの方法があります。

- LDAP クライアント・エントリー情報を使用するクライアントを追跡し、その LDAP ホストの属性の一部をリース状態属性に関連付けます。
- IP アドレスで見つけることができるオブジェクトを作成および更新します。Cisco Prime Network レジストラーがこれらのオブジェクトを作成する場合、DHCP サーバーのリース状態に一致する(またはある)LDAP オブジェクトのレベルを作成できます。

Cisco プライムネットワーク レジストラーを使用する場合は、次の点に注意する必要があります。

- DHCPサーバーは、単一のオブジェクトからの読み取りと書き込みのみを行います。クライアントエントリデータの読み取りとリース状態の日付を保持するために別々のオブジェクトを使用できますが、Cisco Prime Network レジストラーは、あるオブジェクトと別のオブジェクトから属性を読み取ることはできません。
- すべてのデータベースアクセスと同様に、LDAP クエリのパフォーマンスは、インデックス付き属性によって異なります。クエリ フィルターで使用するように構成した属性にインデックスを付けていない場合は、パフォーマンスが低下します。
- LDAP 属性は、サーバーのインストール時に LDAP スキーマで事前設定されるか、または Cisco Prime Networkレジストラー以外の他の方法で作成する必要があります。

LDAP 状態の更新の設定

LDAP サーバーに対してリース状態更新を実行するには、次の 2 つのオプションを使用できます。

- **更新検索パス**: DHCP サーバーは、まず更新のdnを検索するためにクエリを実行します。
- ***dn-format***—サーバーには、更新用のdnが提供されます。つまり、DHCP は更新前にクエリを実行しなくても直接更新を実行します。

オプション 1: update-search-path オプションの使用

次の例は、最初のオプションである更新検索パスを示しています。LDAP オブジェクトの識別名 (dn) をリース状態で使用可能なデータから構築できない場合の処理を示します。DHCP サーバーは、更新検索 xxx情報に基づいて LDAP クエリを作成し、LDAP オブジェクトを検索し、その dnを使用して LDAP 更新を発行します。

次の表に示す例では、標準 LDAP 組織の個人オブジェクト クラス属性を使用して、リース更新データを保持していることを前提としています。

表 *40 : LDAP* と *DHCP* のマッピングの例

属性（Attribute）	DHCP リースエントリマッピング
uid	アドレス（IP アドレス）
カーライセンス	状態 (リース状態)

ステップ 1 LDAP 構成でサーバーのホスト名を指定して、LDAP サーバーについて DHCP に伝えます。

ステップ 2 LDAP サーバーに接続するときに使用するログイン情報を設定します。この CLI の例では、管理者に joe と、アクセスするパスワードを設定します。ユーザーに識別名（dn）を使用します。

```
nrcmd> ldap myserver set username="cn=joe,o=Example Corporation,c=US" password=access
```

ステップ 3 DHCP サーバーが更新するオブジェクトのディレクトリ内の開始点である更新検索パス属性を構成します。また、更新検索の範囲も設定できます。この CLI の例では、組織単位 (ou) IT、組織のサンプル コーポレーション、および国 US から開始する検索パスを設定します。更新検索範囲は、サブツリーに設定されます。

```
nrcmd> ldap myserver set update-search-path="ou=IT,o=Example Corp,c=US"
update-search-scope=SUBTREE
```

ステップ 4 更新する LDAP オブジェクトの検索に使用する属性の ID を設定します。次の CLI の例では、検索属性をクライアント MAC アドレスに設定します。

```
nrcmd> ldap myserver set update-search-attribute=client-mac-addr
```

ステップ 5 更新検索属性の書式を設定するフィルタ式を構成します。この式には、検索属性データを置換する場所を示す「%s」を含める必要があります。次は CLI の例です。

```
nrcmd> ldap myserver set update-search-filter=(cn=%s)
```

ステップ 6 update-dictionary 属性を設定すると、対応するリース状態属性の値を使用して設定する LDAP 属性を識別できます。この例では、LDAP UID を更新して IP アドレスを含め、カーライセンス属性を更新して DHCP リース状態情報を含める必要があることを指定します。CLI の使用：

```
nrcmd> ldap myserver setEntry update-dictionary uid=address carlicense=state
```

ステップ 7 新しい LDAP サーバーの更新を有効にします。次は CLI の例です。

```
nrcmd> ldap myserver enable can-update
```

ステップ 8 DHCP サーバーをリロードします。

オプション 2: dn-format オプションの使用

この例では、2 番目のオプションdn-formatを使用する方法を示します。

ステップ 1 LDAP 構成でサーバーのホスト名を指定して、LDAP サーバーについて DHCP に伝えます。

ステップ 2 LDAP サーバーに接続するときに使用するログイン情報を設定します。この CLI の例では、管理者に joe と、アクセスするパスワードを設定します。ユーザーのdnを使用します。

```
nrcmd> ldap myserver_option2 set username="cn=joe,o=Example Corporation,c=US"
password=access
```

ステップ 3 dn-format 文字列を使用して、更新の検索を開始する LDAP サーバーのデータベース階層内の場所を指定します。次は CLI の例です。

```
nrcmd> ldap myserver_option2 set dn-format="cn=\"%s\",ou=IT,o=Example Corp,c=US"
```

ステップ 4 dn-format 文字列が参照する dn-attribute 属性を設定します。次の CLI の例では、dn 属性をクライアント MAC アドレスに設定します。

```
nrcmd> ldap myserver_option2 set dn-attribute=client-mac-addr
```

ステップ 5 更新するエントリを指定します。CLI の使用：

```
nrcmd> ldap myserver_option2 setEntry update-dictionary uid=address carlicense=state
```

ステップ 6 can-update 属性を有効にします。次は CLI の例です。

```
nrcmd> ldap myserver_option2 enable can-update
```

ステップ 7 DHCP サーバーをリロードします。

LDAP エントリ作成の設定

このセクションでは、LDAP エントリについて説明します。LDAP エントリの作成機能を使用すると、エントリを検索し、現在のリース情報で更新することができます。エントリが作成されるのは、エントリが見つからないために状態更新操作が失敗した場合だけです。

前の例の手順を実行した後、CLI の次の手順を実行します。

ステップ 1 client-mac-addr フィールドなどのリースオブジェクト属性の LDAP サーバーに対して dn-attribute プロパティを設定し、 dn-format 文字列を設定します。次に CLI の例を示します。

```
nrcmd> ldap myserver set dn-attribute=client-mac-addr dn-format="cn=\"%s\",ou=IT,o=Example Corp,c=US"
```

この手順は、更新検索パスオプションを使用してリース状態の更新を構成する場合にのみ必要です。（オプション 1: update-search-path オプションの使用 （383 ページ） を参照）。dn フォーマット文字列を使用してリース状態の更新を構成する場合は、この手順をスキップします。（オプション 2: dn-format オプションの使用 （384 ページ） を参照）。

ステップ 2 既存の dn-attribute プロパティと組み合わせるときに作成するエントリの dn を指定します。次は CLI の例です。

```
nrcmd> ldap myserver set dn-create-format="cn=\"%s\",ou=IT,o=Example Corp,c=US"
```

The Cisco Prime Network Registrar client-mac-addr フィールドでは、 1,6:xx:xx:xx:xx:xx:xx 形式を使用します。コンマ文字は LDAP の特殊な区切り文字であるため\"、dnを引用符で囲むには、その文字を使用する必要があります。

ステップ 3 ディクショナリ作成プロパティーを使用して、一連の名前と値のペアを入力して、LDAP 属性とリース状態属性の間のマッピングを確立します。LDAP 属性は、対応するリース状態の属性の値に設定されたエントリ属性を示します。CLI：

```
nrcmd> ldap myserver setEntry create-dictionary sn=client-host-name

nrcmd> ldap myserver setEntry create-dictionary givenname=client-class-name

nrcmd> ldap myserver setEntry create-dictionary localityname=client-domain-name
```

ステップ 4 create-object-classes プロパティを使用して、エントリを作成するときに使用するオブジェクトクラスを指定します。次は CLI の例です。

```
nrcmd> ldap myserver set create-object-classes="top,person,organizationalPerson,inetorgperson"
```

ステップ 5 LDAP サーバーの myserver のエントリ作成を有効にします。次は CLI の例です。

```
nrcmd> ldap myserver enable can-create
```

（注） 属性を作成できる属性を有効にする前に、更新可能属性を有効にします。例については、LDAP 状態の更新の設定 （382 ページ）を参照してください。

ステップ 6 DHCP サーバーをリロードします。

ステップ 7 作成、クエリ、および更新が正常に行われたかどうかを確認するには、LDAP ログの設定を表示します。

LDAP のトラブルシューティング

以下のセクションでは、LDAP サーバーの障害の微調整と検出に関するアドバイスを示します。

関連項目

LDAP 接続の最適化 （385 ページ）

LDAP の推奨値 （386 ページ）

LDAP 接続の最適化

個別に微調整が可能な読み取りオブジェクトと書き込みオブジェクトを使用して、LDAP 接続を最適化できます。この CLI の例では、書き込み (作成および更新) 操作を調整し、より長いサーバー処理を必要とします。

```
nrcmd> ldap LDAP-Write create csrc-ldap password=changeme port=389 preference=1

nrcmd> ldap LDAP-Write setEntry query-dictionary csrcclientclasas=client-class-name

nrcmd> ldap LDAP-Write set
search-filter=(&(macaddress=%s)(|(crsccclassname=Computer)(csrcclassname=Modem)))

nrcmd> ldap LDAP-Write set search-path=csrcprogramname=csrc,o=NetscapeRoot

nrcmd> ldap LDAP-Write set
username=uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot

nrcmd> ldap LDAP-Write disable can-query

nrcmd> ldap LDAP-Write enable can-create

nrcmd> ldap LDAP-Write enable can-update

nrcmd> ldap LDAP-Write enable limit-requests

nrcmd> ldap LDAP-Write set connections=2 max-requests=8 timeout=10s
```

次の CLI の例では、読み取り (クエリ) 操作を調整します。

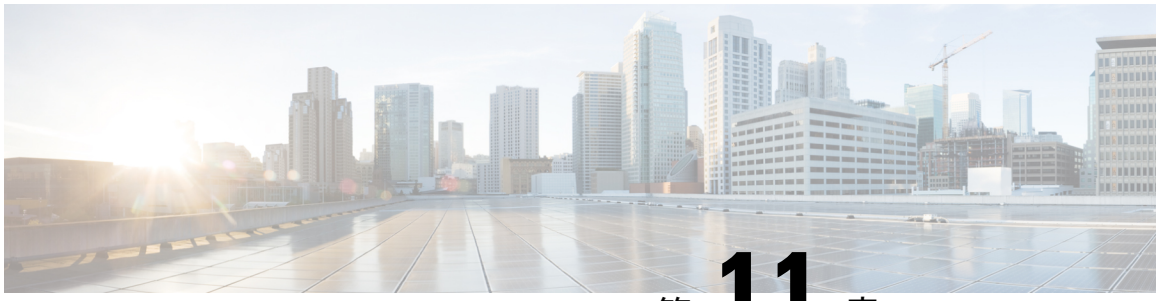
```
nrcmd> ldap LDAP-Read create csrc-ldap password=changeme port=389 preference=1

nrcmd> ldap LDAP-Read setEntry query-dictionary csrcclientclasas=client-class-name

nrcmd> ldap LDAP-Read set
search-filter=(&(macaddress=%s)(|(crsccclassname=Computer)(csrcclassname=Modem)))

nrcmd> ldap LDAP-Read set search-path=csrcprogramname=csrc,o=NetscapeRoot

nrcmd> ldap LDAP-Read set
username=uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot

nrcmd> ldap LDAP-Read enable can-query

nrcmd> ldap LDAP-Read disable can-create

nrcmd> ldap LDAP-Read disable can-update

nrcmd> ldap LDAP-Read enable limit-requests

nrcmd> ldap LDAP-Read set connections=3 max-requests=12 timeout=4s
```

LDAP の推奨値

以下の表は、いくつかの重要な LDAP 属性の推奨値を示しています。

表 *41 : LDAP* 属性の推奨値

属性と値	説明
接続=5 ~ 25	サーバーが LDAP サーバーに対して行う必要がある接続の数。これは、主にパフォーマンス・チューニング・パラメーターです。デフォルト値は 1 接続です。場合によっては、複数の接続によって全体的なスループットが向上することがあります。この量は、LDAP サーバーの負荷によって異なります。LDAP を使用するアプリケーションが多数ある場合は、5 つの接続が適切です。LDAP を使用した Cisco プライムネットワーク レジストラーだけで、25 が適切です。
スレッド待機時間= 2	LDAP クライアント接続が結果をポーリングする間隔 (ミリ秒単位)。
query-timeout =3	Cisco プライムネットワーク レジストラー DHCP サーバーは、フェールオーバーとクエリが設定されている場合は、クエリ タイムアウト間隔でフェールオーバーします。デフォルト設定は 3 秒で、推奨されます (DHCP サーバーのデフォルトの 4 秒のドロップ・オールド・パケット値よりも小さいため、接続が非アクティブで LDAP サーバーが「異常」と見なされます)。
timeout =10	LDAP 要求が接続キューに残っている秒数で、失効とタイムアウトが宣言されます。クライアントのタイムアウト期間の後に DHCP クライアントが受信した応答は、古くなっています。デフォルトは 10 秒で、推奨されます。Cisco Prime Network レジストラー DHCP サーバーは、フェールオーバーと更新可能または作成が有効な場合にタイムアウト間隔でフェールオーバーします。



第 11 章

式の使用方法

Cisco プライムネットワーク レジストラーは、クライアント クラスのサポートを強化します。クライアントデータベースにクライアントを登録しなくても、要求の内容に基づいてクライアントクラスに要求を配置できるようになりました。また、サブスクライバのアクティブなリース数に基づいてクライアントクラスに要求を配置できるようになり、さまざまな加入者に提供されるサービスのレベルに制限が与えることができるようになります。これは、式を使用した特別な DHCP オプションの処理によって可能です。

DHCPリレー エージェント情報オプション (RFC 3046 で説明されているオプション 82) の値に基づいて、加入者アドレスの制限を設定できます。これらの値は、機密性の高いアドレスを明らかにする必要はありません。オプション 82 サブオプション (リモート IDまたは回線 ID)またはその他の DHCP オプションに対して着信 DHCPDISCOVER 要求パケットを評価する式を作成することによって、個々の加入者に関連付ける値を作成できます。この式は、パケット内で評価される内容に応じて異なる値を返す一連のifステートメントです。これは、事実上、サブスクライバが属するクライアントクラスを計算し、アドレスの割り当てをそのクライアントクラスのスコープに制限します。

（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 （433 ページ） 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。ここで説明する式も正規表現と同じではありません。

式の使用方法

式の使用方法

式処理は、次の場所で使用されます。

- — クライアントクラス検索 ID .Calculating a client-class この式は、着信パケットの内容に基づいてクライアントクラスを決定します。
- Creatingthe key to look —upクライアント検索 ID in . the client-entry database 式の評価結果のキーを使用して、クライアント エントリ データベースにアクセスします。
- Creatingthe ID to use to —limit制限 ID clients . of the same subscriber これは、他のクライアントがこのサブスクライバに関連付けられているかどうかを確認するために使用する ID です。これは DHCPv4 (DHCPv6 ではない) に対してのみサポートされます。
- **オプション値の作成**：オプションに対して式を使用する （426ページ） を参照してください。

この種の処理は、次のシナリオで発生します。

1. DHCP サーバーは、クライアント クラスルックアップ ID 式に基づいてクライアント クラスを取得しようとします。クライアントクラスを計算できない場合は、通常のMACアドレスメソッドを使用してクライアントを検索します。
2. サーバーがクライアントクラスを計算できる場合は、クライアント参照IDを返すクライアントルックアップ ID式の評価に基づいて、クライアント エントリ検索を実行する必要があるかどうかを判断します。そのような ID を持つ場合は、それを使用してクライアントを検索します。そのような ID がない場合は、計算されたクライアント クラス値を使用してアドレスを割り当てます。
3. サーバーがクライアントルックアップIDを使用し、クライアント・エントリーを見つけた場合、クライアントのデータを使用します。クライアント エントリが見つからない場合は、計算されたクライアント クラス データまたは既定のクライアント クラス データが使用されます。

DHCPv4 の場合、割り当てられたアドレスの上限を、ポリシー・レベルで同一の制限 id値を持つネットワークまたはLANセグメント上のクライアントに設定することもできます。ポリシーの制限カウント属性を使用して、この上限を正の整数として設定します。同様の処理は、v6 クライアント クラスルックアップ IDとv6 クライアントルックアップ ID式を使用して DHCPv6 で可能です。

IP アドレスを加入者に制限するために設定する値は次のとおりです。

- ポリシーの場合は、制限カウント属性を正の整数に設定します。
- クライアント クラスの場合、limit-id属性とクライアントルックアップ ID属性を式に設定し、クライアント クラスに対してlimit-limit-client-class-name属性を設定します。
- クライアントの場合は、クライアント クラスに対して、クライアント クラス名の上限属性を設定します。

使用する式については、 を式の作成 （392 ページ）参照してください。

式の入力

属性定義に単純な式を含めるか、式ファイルに複雑な式を含め、属性定義でファイルを参照することができます。いずれの場合も、最大許容文字は 16 KB です。

CLI で設定されるほとんどの式はテキストファイルに格納され、その後、必要な設定属性に関連付けられます。このファイルのデフォルトパスは、現在の作業ディレクトリです。テキストファイルに格納せずに、CLI で単純な式を直接設定できます。単純な式は、CLI に入力する際に、次の規則に従う必要があります。

- 1 つのコマンド ラインに制限する必要があります。
- 式全体を二重引用符 ( )" "で囲む必要があります。
- 埋め込まれた二重引用符はバックスラッシュ(\) でエスケープする必要があります。

クライアントクラスルックアップ IDを設定する単純な式の例を次に示します。

```
\"limit\"
```

クライアントクラスの制限 idを設定するために、もう少し詳しい例を使用する場合は、

```
(request option 82 "circuit-id")
```

CLI のコマンド解析に制限があるため、この式を CLI に直接入力することはできません。複雑な式をテキスト ファイルに配置して入力し、そのファイルを属性定義内の "at" 記号 ( )@で参照する必要があります。たとえば、その式が cclookup.txt ファイルに置かれている場合、CLI コマンドは次のようになります。

```
nrcmd> dhcp set client-class-lookup-id=@cclookup.txt
```

ファイル内の式の構文には、単純な式の余分な要件(文字の間隔とエスケープ)はありません。また、シャープ記号(#)、ダブルスラッシュ(/)、セミコロン(;))、行末で終了するコメント行を含めることもできます。次の例を参考にしてください。

```
// Expression to set client-class based on remote-id
(if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
  "no-limit"
  "limit")


// Expression to calculate client-class based on remote-id
(try
  (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
    "cm-client-class"
    "cpe-client-class")
  "<none>")
```

前の例の IPv6 バージョン (オプション番号を使用) は、次のとおりです。

```
// Expression to calculate client-class based on DOCSIS 3.0 cm-mac-address
(try
  (if (equal (request option 17 enterprise-id 4491 36)
```

```
          (or (request relay option 17 enterprise-id 4491 1026) "none"))
    "v6-cm-client-class"
    "v6-cpe-client-class")
  "<none>")
```

数値の代わりにオプション名を置き換えて、前の式を記述することもできます。

```
// Expression to calculate client-class based on DOCSIS 3.0 cm-mac-address
(try
  (if
    (equal
      (or
        (request option
        "vendor-opts" enterprise-id "dhcp6-cablelabs-config" "device-id")
         (substring (request option "client-linklayer-address") 3 8))
      (or
        (request relay option
          "vendor-opts" enterprise-id "dhcp6-cablelabs-config" "cm-mac-address")
        "none"))
    "v6-cm-client-class"
    "v6-cpe-client-class")
  "<none>")
```

例orの機能により、パケットがリレーされなかった場合、またはリレー エージェントがオプションを追加しなかった場合、サーバーはクライアントをCPEと見なし、ケーブルモデム(CM)ではないと見なします。

式の作成

DHCP式を使用すると、受信したDHCPパケットのデータに基づいて、取得、処理、および決定を行うことができます。着信パケットのクライアントクラスを決定するために使用し、オプション82制限サポート用の同等キーを作成することができます。パケットと個々のオプションから情報を取得する方法、パケット内の情報に基づく決定を可能にするさまざまな条件関数、およびクライアントクラスの名前またはキーを作成できるデータ合成機能を提供します。

例を記述する式ファイルに含める式一般的な制限シナリオ（368ページ）は次のようになります。

```
// Begins the try function
(try
  (or
    (if (equal
          (request option "relay-agent-info" "remote-id")
          (request chaddr))
      "cm-client-class")
    (if (equal
          (substring (request option "dhcp-class-identifier") 0 6)
          "docsis")
      "docsis-cm-client-class")
    (if (equal
          (request option "user-class")
          "alternative-class")
      "alternative-cm-client-class"))
  "<none>")
```

```
// Ends the try function
```

式は関数をor使用し、3ifつの関数を評価します。より簡単な形式では、クライアントクラスを計算し、この式を cclookup.txt ファイルに含めることができます。

```
// Expression to calculate client-class based on remote-id
(try
  (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
    "cm-client-class"
    "cpe-client-class")
  "<none>")
```

式を使用してサーバーのクライアント クラスルックアップ ID を設定するには、次のファイルを参照してください。

```
nrcmd> dhcp set client-class-lookup-id=@cclookup.txt
```

制限キーは、オプション 82 からremote-idサブオプションを取得し、できない場合は標準 MAC BLOB キーを使用して、制限キーを生成できます。ファイルに式を含め、ファイル内の制限 ID をcclimit.txt設定します。

```
// Expression to use remote-id or standard MAC
(try (request option "relay-agent-info" "remote-id") 00:d0:ba:d3:bd:3b)
```

式の構文

式は、関数とリテラルだけで構成されます。その構文は、Lispの構文に似ています。それは同じ規則の多くに従い、可能であればLisp関数名を使用します。基本のシンタックスは次のとおりです。

```
(function argument-0 ... argument-n)
```

より便利な例は次のとおりです。

```
(try
  (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
    "cm-client-class"
    "cpe-client-class")
  "<none>")
```

この例では、Relay エージェント情報オプション (オプション 82) のremote-idサブオプションをパケット内の MAC アドレスと比較し、それらが同じ場合は "cm-client-class" を返し、異なる場合は "cpe-client-class" を返します。(式がデータを評価できない場合、try関数は "<none>" 値を返式が失敗する可能性 （395 ページ）します。目的は、デバイスがケーブルモデムであるかどうかを判断すること (リモートIDが MAC アドレスと等しいと考えられます) を確認し、その場合は、デバイスを顧客宅内の機器や PC とは別のクライアント クラスに配置します。関数とリテラルの両方が式であることに注意してください。前の例では、関数を式として示しています。リテラルについては、「」を式のリテラル （394 ページ）参照してください。

式のデータタイプ

式でサポートされるデータ型は次のとおりです。

- Blob- カウントされた一連のバイト数、推奨される最大長は 1 KBの。
- String- 数え切られた一連の NVT ASCII 文字は 0 バイトで終わらず、推奨される最大長は 1 KBのです。
- Signed integer：32 ビット符号付き整数。
- Unsigned integer：32 ビットの符号なし整数。

IP アドレスデータ型はありません。IPv4 アドレスは 4 バイトの BLOB で、IPv6 アドレスは 16 バイトの BLOB です。すべての数字はネットワークバイト順です。データタイプの変換 （395 ページ） を参照してください。

式のリテラル

式機能には、次のようなさまざまなリテラルが含まれています。

- Signed 32 ビットに収まる必要がある標準integersの数値。
- Unsigned 32 ビットに収まる符号なしのintegers正規数。
- Blobs：コロン区切りの 16 進バイト。たとえば、01:02:03:04:05:06 は、バイト 1 から 6 までの 6 バイトの BLOB です。これは"01:02:03:04:05:06" (17 バイトの文字列) とは異なります。文字列は、BLOB のテキスト表現によって BLOB に関連付けられています。たとえば、式(to-blob "01:02:03")は BLOB 01:02:03 を返します。01 は整数に変わるので、1 バイトの BLOB のリテラル表現を作成できないことに注意してください。1 を含む 1 バイトの BLOB を(byte 1)取得するには、01 の BLOB を返すように使用できます。または、(substring(to-blob1)3式1)を使用することもできます。3 は、4 バイト整数の 4 バイト目 (00:00:00:01) を抽出するオフセットを示し、1 は抽出されたバイト数で、結果は "01" です。
- String：二重引用符で囲まれた文字。たとえば、"example.com" は文字列で、"01:02:03:04:05:05"と入力します。リテラル文字列に引用符を入れるには、次の例に示す円記号 (\) を使用してエスケープします。

```
"this has one \"quote"
```

整数リテラル(符号付きおよび符号なし)は、10の底にあると見なされます。0から始まる場合は 8 進数とみなされます。0x で始まる場合は、16 進数と見なされます。リテラルの例を次に示します。

- "hello world" は文字列リテラル (および完全に有効な式) です。
- 1 は符号なし整数リテラルです (完全に有効な式でもあります)。この値には 4 バイトが含まれ、最初の 3 バイトは 0 で、最後のバイトは最下位ビットに 1 を含みます。
- 01:02:03 は、3 バイト、01、02、および 03 を含む BLOB リテラルです。
- –10 は、10 進数 -10 の 2 の補数表現を持つ 4 バイトを含む符号付き整数リテラルです。

式の戻り型の値

例外が少ない場合は、式のポイントは値を返す点です。クライアントクラスを決定するように構成された式は、 DHCP サーバー プロパティクライアント クラス検索 IDで構成されます。こ

の式が評価されると、DHCPサーバーは、クライアントクラスの名前または文字列を含む文字列"<none>"を返すことを DHCP サーバーが想定します。

すべての関数は値を返します。値のデータ型は、引数のデータ型によって異なります。式によっては、特定のデータ型の引数しか受け付けなれないものがあります。例えば：

```
(+ argument0  argument1)
```

ほとんどの場合、特定の引数に特定のデータ型を必要とする関数は、取得した引数を適切なデータ型に変換しようとします。たとえば(+"1"、2)文字列リテラル "1" を数値 1 に変換できたため、3 を返します。ただし、「1」(+ "one" 2)は正常に数値に変換されないため、エラーが発生します。一般に、式エバリュエーターは、データ型変換の決定を行う際に、可能な限り正しいことを行おうとします。

式が失敗する可能性

式を構成する関数の中には、データ型や値に対して正しく動作するものもありますが、多くの関数は正しく動作しません。前のセクションでは、+この関数は文字列リテラル "one" を有効な数値に変換しなかったため、その関数の評価に失敗しました。関数が評価に失敗すると、その呼び出し関数も失敗し、式全体が失敗するまで失敗します。式の評価が失敗した場合、関係する式によって結果が異なります。場合によっては、パケットがドロップされる可能性がありますが、警告メッセージを生成する場合もあります。

(try 式の失敗式) 関数を使用して、評価が失敗するのを防ぐことができます。関数tryは式を評価し、成功した場合は関数の値が式の値になります。評価が失敗した場合(何らかの理由で)、関数の値は失敗式の値になります。関数自体が失敗するtry唯一の状況は、失敗式の評価が失敗した場合です。したがって、どの式をエラー式として定義するか注意する必要があります。文字列リテラルは安全な賭けです。したがって、関数を使用してクライアントクラスルックアップ IDの評価をtry保護することをお勧めします。前に引用した例は、これがどのように機能するかを示しています。

```
(try
  (if (equal (request option "relay-agent-info" "remote-id")
             (request chaddr))
    "cm-client-class"
    "cpe-client-class")
  "<none>")
```

この場合、関数ifの評価が失敗した場合、クライアントクラスルックアップID式の値は"<none>"になります。もちろん、代わりにクライアントクラスの名前だったかもしれません。

データタイプの変換

関数が特定のデータ型の引数を必要とする場合、そのデータ型に値を変換しようとします。このエラーが発生する場合が多いため、関数全体が失敗することがあります。データ型変換は、to-string 、 to-blob、to-sintおよびto-uint関数によっても実行されます。関数が特定のデータ型の引数を必要とするたびに、外部から利用できる関数の内部バージョンを呼び出します。

また、as-string、 as-blob、as-sint、およびas-uint変換関数もあり、値のデータは目的のデータ型として再ラベル付けされます。次の表に、両方の関数セットの変換マトリックスが表示されます。

to-stringとas-stringの違いに注意してください。たとえば、BLOB形式のデータがあるとします。このデータは、要求パケットからデータを取得する関数評価 (**要求** get **オプション**) の結果、または blob データをサブ文字列で処理した結果として使用される場合があります。このデータが BLOB 型であっても、実際に ASCII 文字列データを表す場合は、文字列として使用することをお勧めします。変換には as-string と to-string の 2 つの選択肢があります。どちらを選ぶべきでしょうか? データが ASCII バイトで構成されており、そのデータ型を文字列としてそのまま認識し、基本的にリセットする場合は、as-string関数を使用します。つまり、BLOB のバイトを文字列として使用します。BLOB 00:01 は文字列に変換できず、試してみるとエラーがスローされます。blob 68:65:6c:6c:6f は、as-string で文字列に変換して "hello" を生成します。一方、ASCIIデータである可能性もない可能性もある一連のバイトがある場合で、データをBLOBの文字列形式で表すには、 to-string を使用する必要があります。たとえば、to-string は最初が 0 次が 1 から成る 2 バイトの BLOB を文字列 "00:01" に変換します。

表 *42 :* データ型変換行列

機能	文字列	BLOB	符号付き整数	符号なし整数
as-blob	失敗することはできません。ASCII文字にBLOB バイトとして再ラベルを付けます。	—	失敗することはできません。は、整数の4 バイトから 4 バイトのBLOBを生成します。	失敗しません。4 バイトの整数 から 4 バイトの BLOB を生成します。
as-sint	通常は役に立ちません。は、1 バイト、2 バイト、3バイト、または 4 バイトの文字列を BLOB に変換し、それを符号付き整数にパックします。	通常は役に立ちません。は 1 バイト、2バイト、3バイト、または 4 バイトの BLOB のみを変換します。	—	失敗することはできません。より大きな符号なし整数が正符号付き整数に収まる場合は、負の符号付き整数に変換されます。
as-string	—	文字列バイト (印刷可能な文字の場合) として再ラベル付けする	4 バイトの BLOB に変換し、それを BLOB として処理します (いくつかの特殊な整数を除いて失敗します)。	4 バイトの BLOB に変換し、BLOB として処理します(いくつかの特殊な整数を除いて失敗します)。

機能	文字列	BLOB	符号付き整数	符号なし整数
as-uint	通常は役に立ちません。1 バイト、2 バイト、3 バイト、または 4 バイトの文字列を blob に変換し、次に符号付き整数に変換します。	通常は有用ではありません。1、2、3、4 バイトの BLOB のみを変換します。	失敗することはできません。は符号なし整数に変換され、負の符号付き整数は大きな符号なし整数になります。	—
to-blob	"01:02:03" の形式である必要があります。	—	失敗することはできません。は、整数の 4 バイトから 4 バイトの BLOB を生成します。	失敗しません。4 バイトの整数 から 4 バイトの BLOB を生成します。
to-sint	nまたは–nの形式でなければなりません。	1 バイト、2 バイト、3 バイト、または 4 バイトの BLOB のみ。	—	大きすぎて符号付き整数に収まらない場合にのみ変換します。
to-string	—	失敗しません	失敗しません	失敗しません
to-uint	形式n である必要があります。	1、2、3、4 バイトの BLOB のみ。	非負のみ。	—

式の関数

以下のセクションでは、式関数をリストします。式はかっこで囲む必要があります。

+、-、*、/、%

構文：

(+ arg1 ... argn)

(– arg1 ... argn)

(* arg1 ... argn)

(/ arg1 ... argn)

(% arg1 arg2)

説明：

符号付き整数または式の算術演算は、符号付き整数に変換できます。符号付き整数に変換できない (かつ null でない) 引数は、エラーを返します。null に評価される引数は無視されます (た

だし、–および/の最初の引数はnullに評価できません)。これらの関数は常に符号付き整数を返します(オーバーフローとアンダーフローは現在捕捉されないことに注意してください)。

- +引数を合計します。引数がない場合、結果は0になります。
- –単一の引数の値を否定するか、または複数の引数の場合は、残りの値を最初の引数から連続して減算します。たとえば、(– 3 4 5)は-6になります。
- *引数の値の積を取ります。引数がない場合、結果は1になります。
- /連続して最初の引数を他のすべての引数で除算します。例えば、(/ 100 4 5)は5になります。最初の引数以外の引数が0の場合は、エラーが返されます。
- %は、最初の引数の結果の残りを2番目の引数で除算した剰余を決定する剰余算術演算子です。例えば、(% 12 7) は5 (12 / 7 = 1 * 7 + 5) となります。

例：

(+ 1 2 3 4) は10を返します

(- 10 5 2) は3を返します

(* 3 4 5) は60を返します

(/ 20 2 5) は2を返します

(/ 20 0) はエラーを返します

(% 12 7) は5 (12/7=1*7+5) を返します

---

and

構文：

(and arg1 ... argn)

説明：

引数を左から右の順に評価します。引数がnullと評価された場合、引数の評価を停止し、nullを返します。それ以外の場合は、最後の引数argnの値を返します。

例：

(and "hello" "world") は "world" を返します

(and (request option 82 1) (request option 82 2)) は、オプション82サブオプション1とサブオプション2の両方が要求に存在する場合は、オプション82サブオプション2を返し、それ以外の場合はnullを返します。

---

as-blob

構文：

(as-blob expr)

説明：

exprを BLOB として扱います。exprが文字列に評価された場合、その文字列を構成するバイトは返される BLOB のバイトになります。exprが BLOB に評価される場合、その BLOB は変更されずに返されます。exprがいずれかの種類の整数に評価された場合、整数のバイトを含む 4 バイトの BLOB が返されます。

例：

(as-blob "hello world") は、blob の 68:65:6c:6c:6c:6f:20:77:6f:72:6c:64 を返します

---

as-sint

構文：

(as-sint expr)

説明：

exprを符号付き整数として扱います。exprが 4 バイト以下の文字列または BLOB に評価された場合、関数はそれらのバイトから構築された符号付き整数を返します (4 バイトより長い場合はエラーを返します)。exprが符号付き整数に評価された場合、値は変更されずに返されます。符号なし整数の場合、同じビット値を持つ符号付き整数を返します。

例：

(as-sint ff:ff:ff:ff) は -1 を返します

(as-sint 2147483648) はエラーを返します

---

as-string

構文：

(as-string expr)

説明：

exprを文字列として扱います。exprが文字列に評価された場合、その文字列を返します。exprが BLOB に評価された場合、出力できない ASCII 値でない限り、BLOB 内のバイトから構築された文字列を返します。exprが整数に評価された場合、その値は単一文字の ASCII 値であると見なされ、それがエラーを返す印字出来ない文字列でない限り、その 1 文字から成る文字列が返されます。

例：

(as-string 97) は "a" を返します

(as-string 68:65:6c:6c:6f:20:77:6f:72:6c:64) は "hello world" を返します

(as-string 0) はエラーを返します

as-uint

構文：

(as-uint expr)

説明：

exprを整数として扱います。exprが 4 バイト以下の文字列または BLOB に評価された場合、それらのバイトから構築された符号なし整数を返します。4 バイトより長い場合は、エラーを返します。結果が符号なし整数の場合は、引数をそのまま返します。符号付き整数の場合、同じビット値を持つ符号なし整数を返します。

例：

(as-uint-2147483648) は、符号なし整数 2147483648を返します

(as-uint-1) は、符号なし整数 4294967295を返します

(as-uintff:ff:ff:ff) は、符号なし整数 4294967295を返します

ash

構文：

(ash expr shift)

(lshift expr shift)

説明：

shift 量によってビットがシフトされた整数または BLOB を返します。exprは、整数、BLOB、または文字列に評価できます。exprが文字列に評価された場合、この関数は文字列を符号付き整数に変換しようとします。両方とも失敗した場合は、エラーを返します。shiftは、符号付き整数に変換可能なものに評価する必要があります。shiftが正の値の場合、シフトは左になります。負の値を指定すると、シフトは右になります。exprの結果が符号付き整数の場合、右シフトは符号拡張を伴います。exprの結果が符号なし整数またはBLOBになる場合、右シフトは最上位ビットで 0 ビットシフトします。

例：

(ash00:01:001) は、ブロブ 00:02:00 を返します

(lshift00:01:00-1) は、ブロブ 00:00:80 を返します

(ash11) は、符号なし整数 2 を返します

bit

構文：

(bit-and arg1 arg2)

(bit-andc1 arg1 arg2)

(bit-andc2 arg1 arg2)

(bit-eqv arg1 arg2)

(bit-or arg1 arg2)

(bit-orc1 arg1 arg2)

(bit-orc2 arg1 arg2)

(bit-xor arg1 arg2)

説明：

2 つの引数に対するビット単位のブール演算の結果を返します。結果のデータ型は、両方の引数がいずれかの種類の整数を返す場合は符号付き整数になります。arg1引数とarg2引数は、2 つの整数、2 つの同じ長さの BLOB、または 1 つの整数と 1 つの長さ 4 の blob に評価される必要があります。いずれかの引数が文字列に評価された場合、関数は文字列を符号付き整数に変換し、失敗した場合はBLOBに変換しようとします。この変換後、結果は上記の条件に一致する必要があります。これらの条件が満たされない場合は、エラーを返します。

演算c1とc2、それぞれ第 1 および第 2 引数が、演算の前に補完されることを示します。

例：

(bit-and 00:20 00:ff) は、00:20 を返します

(bit-or 00:20 00:ff) は、00:ff を返します

(bit-xor 00:20 00:ff) は、00:df を返します

(bit-andc1 00:20 00:ff) は、00:df を返します

---

bit-not

構文：

(bit-not expr)

説明：

exprのビットごとの補数である値を返します。式は、型またはBLOB のいずれかの整数に評価する必要があります。文字列に評価される場合、関数は文字列を符号付き整数に変換しようとします。それが失敗した場合は、BLOBに対して、失敗した場合はエラーを返します。結果のデータ型は、exprとその後の変換を評価した結果と同じです。

例：

(bit-not ff:ff) は、00:00を返します

(bit-not 1) は 4294967295を返します

(bit-not "hello world") は、エラーを返します

---

byte

構文：

(byte arg1)

説明：

1 バイトの BLOB の作成を容易にします。データ型に応じて、この BLOB を返します。

- sint,uint—整数の下位バイトを返します。
- blob—BLOB の最後のバイトを返します。
- string—文字列の最後のバイトを返します。

例：

(byte 150) は、96 の BLOBを返します

(byte 0x96) は、96 の BLOBを返します

---

comment

構文：

(comment comment expr1 ... exprn)

説明：

最初の引数は評価されず、引数が 1 つしかない場合は null を返します。引数が複数ある場合は、引数 expr1からexprnを評価し、 exprnの値を返します。

例：

(comment "this is a comment that won’t get lost" (request option 82 1))

---

concat

構文：

(concat arg1 ... argn)

説明：

引数の値を文字列または BLOB に連結します (null 引数は無視)。最初の引数 (arg1) は、文字列または BLOB に評価する必要があります。評価が整数の場合、関数はそれを BLOB に変換します。arg1のデータ型 (任意の変換後) は、結果のデータ型を決定します。この関数は、後続のすべての引数を結果のデータ型に変換し、この変換が失敗した場合はエラーを返します。

例：

(concat "hello" "world") は、"helloworld" を返します

(concat -1 "world") はエラーを返します

(concat -1 00:01:02) は、blob の ff:ff:ff:ff:00:01:02 を返します

---

datatype

構文：

(datatype expr)

説明：

式の結果のデータ型を返します (expr) 式がエラーなしで評価された場合、データ型を文字列として返します。

- "未設定" (内部、null と見なされます)
- "null"
- "uint"
- "sint"
- "string"
- "blob"

---

dotimes

構文：

(dotimes (var count-expr [result-expr]) exp1 ... expn)

説明：

最初にゼロに設定された単一のローカル整数変数varを持つ環境を作成し、expnを通じてexp1を評価します。次に、varを 1 ずつインクリメントし、count-exprより小さい場合は、expnを通じてexp1を再度評価します。varがcount-expr以上の場合、関数はresult-exprを評価し、dotimes全体の結果として返します。result-exprがない場合、関数は null を返します。

varはローカル変数を定義し、アルファベットの名前でなければなりません。count-expr は、整数に評価するか、1 に変換可能でなければなりません。exp1からexpnは、任意のデータ型に評価できる式です。result-exprはオプションであり、表示される場合は任意のデータ型に評価できます。関数がcount-exprを評価すると、varはバインドされず、count-exprに出現することはで

きません。あるいは、varはresult-exprの評価にバインドされ、count-exprの値を持ちます。result-exprを省略すると、この関数は null を返します。

（注） exp1のvarの値をexpnを通じて変更する場合は、無限ループを簡単に作成できるので注意してください (例を参照)。

例：

(let (x y) (setq x 01:02:03) (dotimes (i (length x)) (setq y (concat (substring x i 1) y)))) は 03:02:01 を返します

(dotimes (i 10) (setq i 1)) は無限ループとなります!

---

environmentdictionary

構文：

(environmentdictionary {get | put val | delete} attr)

説明：

DHCP 拡張環境ディクショナリ属性値を取得、配置、または削除します。valは属性の値で、attrは属性名です。両方とも、初期データ型に関係なく文字列に変換されます。初期環境ディクショナリは変更できませんが、シャドウすることができます (最初のディクショナリ内の何かを再定義することはできますが、それを削除すると、元の初期値が残っています)。get キーワードは "get" のオプションではありません。また、これらの例では、初期環境ディクショナリが使用され、式を「設定」するために使用できる一方で、この関数は、すべての環境ディクショナリを介して拡張機能と通信するためにも使用できます。要求と応答のペア。

例：

nrcmd> dhcp setinitial-environment-dictionary=first=one, second=2

(environmentdictionary get "first") は "one" を返します

(environmentdictionary get "second") は "2" を返します(文字列の 2 です)

(environmentdictionary put "two" "second") は "second" を返します

(environmentdictionary delete "first") は null を返します

---

equal, equali

構文：

(equal expr1 expr2 expr3)

(equali expr1 expr2 expr3)

説明：

このequal関数は、expr1とexpr2を評価した結果の等価性を評価します。等しい場合は、次の値が返されます。

1. 指定されている場合はexpr3の値を返します。
2. expr2の値 (および可能な文字列変換後のデータ型) は、expr2が null でない限り、それ以外の値です。
3. 文字列 "*T*" (null を返すと、比較が失敗したことを誤って示すため)。

expr1とexpr2が等しくない場合、この関数は null を返します。

引数には任意のデータ型を指定できます。異なる場合、関数はこれらを比較する前に文字列に変換します (これは失敗できません)。文字列変換は、同等の(to-string..)を使用して無効にすることができます。したがって、blob 61:62 は "ab" 文字列と等しくありません。また、1 バイトの BLOB 01 はリテラル整数 1 と等しくないことに注意してください (どちらも文字列に変換され、"01" と "1" の文字列は等しくありません)。

関数equaliはequal関数と同じですが、比較が文字列に対する比較の場合 (文字列引数が使用されたか、引数が文字列に変換されたため)、大文字と小文字を区別しない比較が使用されます。

例：

(equal (request option "dhcp-class-identifier") "docsis") は、オプションの値 dhcp-class-identifier が "docsis" と同じ文字列である場合、文字列 "docsis" を返します

(equali "abc" "ABC") は "ABC" を返します

(equal "abc" "def") は null を返します

(equal "ab" (as-string 61:62)) "this is true") は "this is true" を返します

(equal "ab" 61:62 "this is not true") は null を返します

(equal 01:02:03 01:02:03) は 01:02:03 を返します

(equal (as-blob "ab") 61:62) は 61:62 を返します

(equal 1 (to-blob 1)) は null を返します

(equal (null) (request option 20)) は、パケットにオプション 20 がない場合、"*T*" を返します

---

error

構文：

(error)

説明：

error 関数の評価の上に try 関数がない限り、式の評価全体が失敗する"回復なし"エラーを返します。

---

if

構文：

(if cond [then else])

説明：

if-then-elseの意味で条件式を評価します。condが null 以外の値に評価された場合、then引数を評価した結果を返します。それ以外の場合は else引数を評価した結果を返します。then および elseは、オプションの引数です。then引数とelse引数を省略すると、cond引数を評価した結果が返されます。else引数を省略し、condが null に評価された場合、この関数は null を返します。3 つの引数のいずれにもデータ型に制限はありません。

例：

```
(if (equali
      (substring (request option "dhcp-class-identifier") 0 6)
      "docsis"
  (request option 82 1))
```

いずれの場合も、dhcp クラス識別子の最初の 6 文字が "docsis" である場合は、オプション 82 のサブオプション 1 を返します。それ以外の場合は null を返します。

---

ip-string

構文：

(ip-string blob)

説明：

4 バイトの IP アドレスBLOBの文字列表現を"a.b.c.d"の形式で返します。単一の引数BLOBは、BLOB に評価するか、または 1 つに変換可能である必要があります。BLOB が 4 バイトを超える場合、この関数は最初の4つのバイトのみを使用してIPアドレス文字列を作成します。BLOB のバイト数が少ない場合、関数は IP アドレス文字列を作成するときに右端のバイトをゼロと見なします。

例：

(ip-string 01:02:03:04) は "1.2.3.4" を返します

(ip-string -1) は "255.255.255.255" を返します

(ip-string (as-blob "hello world")) は "104.101.108.108" を返します

---

ip6-string

構文：

(ip6-string blob)

説明：

16 バイトの IPv6 アドレスBLOBの文字列表現を"a:b:c:d:e:f:g:h"の形式で返します。引数 blob は、blob に評価されるか、blob に変換可能である必要があります。BLOB が 16 バイトを超える場合、この関数は最初の 16 バイトのみを使用して IPv6 アドレス文字列を作成します。BLOB のバイト数が少ない場合、関数は IPv6 文字列を作成するときに右端のバイトをゼロと見なします。

（注） IPv6 アドレスを文字列として表す方法は複数あるため、IPv6 アドレスの文字列形式を比較すると、結果が不整合になる可能性があります。IPv6 アドレスを BLOB 値と比較するのが最善であり、アドレスの表現にあいまいさはありません。文字列形式の IPv6 アドレスが既にある場合は、to-ip6を参照してください。

例：

(ip6-string (as-blob "hello world")) は "6865:6c6c:6f20:776f:726c:6400::" を返します

---

is-string

構文：

(is-string expr)

説明：

exprの評価結果が文字列であるか、文字列として使用できる場合は、exprの値を返します。つまり、as-string がエラーを返さない場合、is-string は expr の値を返します。

例：

(is-string 01:02:03:04) は null を返します

(is-string "hello world") は "hello world" を返します

(is-string 68:65:6c:6c:6f:20:77:6f:72:6c:64) は blob 68:65:6c:6c:6f:20:77:6f:72:6c:64 を返します

---

length

構文：

(length expr)

説明：

値が expr の値の長さ（バイト単位）である整数値を返します。引数exprは任意のデータ型に評価できます。整数は常に長さ 4 を持ちます。文字列の長さには、文字列を終了する可能性のあるゼロバイトは含まれません。

例：

(length 1) は 4 を返します

(length 01:02:03) は 3 を返します

(length "hello world") は 11 を返します

---

let

構文：

(let (var1..varn) expr1 ..expn)

説明：

null 値に初期化されるローカル変数var1からvarnを持つ環境を作成します (setq関数を使用して他の値を指定できます)。ローカル変数が null に初期化されると、関数は式expr1からexprnを順番に評価します。その後、最後の式exprnの値を返します。この関数の利点は、値を一度計算し、ローカル変数に代入してから、その値を再計算せずに他の式で再利用できることです。変数では大文字と小文字が区別されます。

例：

```
(let (x)
  (setq x (substring (request option "dhcp-class-identifier") 0 6))
  (or (if (equali x "docsis") "client-class-1")
      (if (equali x "something else") "client-class-2")))
```

---

log

構文：

(log severity expr)

説明：

exprを文字列に変換した結果をログに記録します。severityとexprは文字列でなければならず、評価が 1 でない場合は 1 に変換されます。severityは null にすることもできます。文字列の場合、次のいずれかの値を持つ必要があります。

- "debug"
- "severity" (severityが null の場合のデフォルト)
- "info"
- "warning"
- "error"

（注） ログ記録はサーバーリソースを大量に消費するため、式に入れるlog関数評価の数を制限します。「error」の重大度がログに記録された場合でも、ログ関数はエラーを返しません。これは、ログメッセージにエラーを示すタグのみを付けます。関数評価の一部としてエラーを返すerror関数を参照してください。

mask-blob

構文：

(mask-blob mask-size length)

説明：

lengthの blob 長さで、BLOB の上位ビットから始まる長さmask-sizeのマスクを含む BLOB を返します。mask-sizeは、整数に評価される式、または変換可能な式です。同様にlengthはmask-sizeより小さくすることはできませんが、0 または正の値を指定する必要があるという点以外は、固定の制限はありません。mask-sizeが 0 より小さい場合は、BLOB の右端から計算されたマスク長を示します。

例：

(mask-blob 1 4) は 80:00:00:00 を生成します

(mask-blob 4 2) は f0:00 を生成します

(mask-blob 31 4) は ff:ff:ff:fe を生成します

(mask-blob -1 4) は 00:00:00:01 を生成します

mask-int

構文：

(mask-int mask-size)

説明：

整数の上位ビットから始まる mask-size のマスクを含む整数を返します。mask-size は整数に評価されるか、または整数に変換される式である必要があります。mask-size が 0 より小さい場合は、整数の右端から計算されたマスク長を示します。

例：

(mask-int 1) は 0x80000000 を生成します

(mask-int 4) は 0xf0000000 を生成します

(mask-int 31) は 0xfffffffe を生成します

(mask-int -1) は 0x00000001 を生成します

---

not

構文：

(not expr)

説明：

exprは、文字列、BLOB、または整数に評価できる式です。その評価の結果が NULL でない場合は、null が返されます。その評価の結果が null の場合、null 以外の値が返されます。exprの値が null の場合に返される null 以外の値は、2 回の呼び出しで同じままであるとは保証されません。

例：

(not "hello world") は null を返します

---

null

構文：

(null [expr1 ... exprn])

説明：

null を返し、その引数を評価しません。

---

or, pick-first-value

構文：

(or arg1... argn)

(pick-first-value arg1... argn)

説明：

引数を順番に評価します。引数の評価が null 以外の値を返す場合、その値が返されます。1 つの引数がnull以外の値を返した後、他の引数は評価されません。それ以外の場合は、最後の引数argnの値を返します。データ型は同じである必要はありません。

例：

```
(or
  (request option 82 1)
  (request option 82 2)
  01:02:03:04)
```

はオプション 82 のサブオプション 1 の値を返し、それが存在しない場合はサブオプション 2 の値を返し、存在しない場合は 01:02:03:04 を返します。

---

parse

構文：

(parse expr1 expr2)

説明：

expr2 で指定されたデータ型として解析された文字列 expr1 を解析した BLOB 結果を返します。expr1 が文字列でない場合は、文字列に変換されます。expr2 は、Cisco Prime Network Registrar でサポートされる AT_* data types（文字列またはその数値）のいずれかである必要があります（オプションの検証タイプ （548 ページ） を参照してください）。

この機能は、Cisco Prime Network Registrar 11.0 で導入されました。

例：

(parse 1234 "AT_INT") は d2:04:00:00 を返します。

(parse "cisco.com" "AT_DNSNAME") は、05:63:69:73:63:67:03:63:6f:6d:00 を返します。

---

progn, return-last

構文：

(progn arg ... argn)

(return-last arg ... argn)

説明：

引数を順番に評価し、最後の引数argnの値を返します。

例：

```
(progn
  (log (null) "I was here")
  (request option 82 1))

(return-last
  (log (null) "I was here")
  (request option 82 1))
```

---

regex

構文：

(regex expr1 expr2 var1... varn)

(regex expr1 expr2)

説明：

指定した target-string (expr2) で正規表現パターン (expr1) と一致するサブ文字列を検索し、指定された変数var1、var2、varnに設定します。つまり、指定されたターゲット文字列 (expr2) で正規表現パターン (expr1) で一致する最初のサブ文字列は、var1 に設定され、2 番目のサブ文字列はvar2に設定されます。変数を指定するときは、let関数の前に置く必要があります。この関数は変数なしで使用することもできますが、この場合、正規表現パターン (expr1) で最初に一致するサブ文字列を、指定されたターゲット文字列 (expr2) で返します。

正規表現パターンの一致は文字列に対してのみ機能するので、パターン (expr1) とターゲット文字列 (expr2) の両方とも文字列である必要があります。そうでない場合、以下の例で使用されるように as-string 関数を使用する必要があります。

例：

(regex "[H][a-z]+" "Hello World") は "Hello" を返します

```
(let (x y z)
  (regex "[H][a-z]+" "Hello Hi World" x y z))
```

は x="Hello"、y="Hi"、z=null を設定し、"Hello"を返します

必要に応じて、let 内の regex の後に追加の式を配置して、x と y を操作できます。

---

request

構文：

(request [get | get-blob] [relay [number]] packetfield)

説明：

DHCPv4 packetfield の有効な値は次のとおりです。

op (blob 1)

htype (blob 1)

hlen (blob 1)

hops (blob 1)

xid (uint)

secs (uint)

flags (uint)

ciaddr (blob 4)

yiaddr (blob 4)

siaddr (blob 4)

giaddr (blob 4)

chaddr (blob hlen)

sname (string)

file (string)

request packetfield関数は、request パケットから指定されたフィールドの値を返します。DHCP request パケットには、オプション領域のオプションと同様に名前付きフィールドが含まれます。この形式の要求関数は、requestパケットから特定の名前付きフィールドを取得するために使用されます。relayキーワードは、request option関数に記述されています。

RFC 2131 で定義されている packetfield の値は、上記のとおりです。要求できるpacketfieldの値がいくつかありますが、未加工の DHCP パケットでは正確にこれらの方法で表示されません。これらはパケットに現れるデータを取り、よく使用される方法で結合します。これらの説明では、想定されるパケットの内容は次のとおりです。

hlen = 1 htype = 6 chaddr = 01:02:03:04:05:06

macaddress-string(string) - MAC アドレスをhlen、 htype、 chaddr形式で返します (たとえば、"1,6,01:02:03:04:05:06")

macaddress-blob(blob) - hlen:htype:chaddr形式の MAC アドレスを返します (たとえば、01:06:01:02:03:04:05:06)

macaddress-clientid(blob) - Microsoft htypeのMAC アドレスから作成されたクライアント ID を返します。

DHCPv6 packetfieldの有効な値は次のとおりです。

msg-type (uint)

msg-type-name (string)

xid (uint)

relay-count (uint)

hop-count (uint)

link-address (blob 16)

peer-address (blob 16)

DHCPv6 のmsg-typeパケット フィールドは、現在のリレーまたはクライアント メッセージの種類を示し、値を持ちます。

1=SOLICIT、2=ADVERTISE、3=REQUEST、4=CONFIRM、5=RENEW、6=REBIND、8=RELEASE、9=-DECLINE、11=INFORMATION-REQUEST、12=RELAY-FORWARD

msg-type-nameパケットフィールドは、メッセージタイプ名の文字列を返します。SOLICIT のように、文字列の値は常に大文字です。

xidは 24 ビット クライアント トランザクション IDで、relay-countは要求内のリレー メッセージの数です。

DHCPv4 パケットから DHCPv6 パケット フィールドが要求されると、エラーが返されます。その逆も同様です。

例：

(request get ciaddr) は存在する場合は ciaddr を返し、それ以外の場合は null を返します

(request ciaddr) は次と同等です (request get ciaddr)

(request giaddr) は、0 以外の場合は giaddr を返し、それ以外の場合は null を返します。

---

request dump

構文：

(request dump)

説明：

現在の要求パケットをログファイルにダンプします。すべての式の評価がdumpキーワードをサポートしているわけではないため、未サポートの場合は無視されます。

---

request option

構文：

(request [get | get-blob] option-request)

ここで option-request は次のとおりです。

1. IPv6 -relay [n] 用のオプションのリレー メッセージ セレクタ
2. 1 つ以上のオプション句 (複数のオプションが IPv6 でのみサポートされています) - option name | id [vendor name | enterprise-id name | id] [instance n]
3. 0 個以上のサブオプション句が続く - name | id [vendor name | enterprise-id name | id] [instance n]
4. オプションの句が続く - [instance-count | count | index n]

説明：

パケットからオプションの値を返します。キーワードは次のとおりです。

- get- 省略した場合は省略可能。
- get-blob- オプションバイトに直接アクセスできるデータを BLOB として返します。
- relay—IPv6 パケットにのみ適用され、それ以外の場合はエラーを返します。クライアント オプションの代わりにリレー オプションを要求します。nは、クライアントに最も近い n 番目のリレー エージェントを示します。省略すると、0 (クライアントに最も近いリレー エージェント) が想定されます。
- option—オプション (およびサブオプション) は、整数または文字列に評価される id または name引数で指定します。これらのいずれかに評価されない場合、関数は変換を行わないため、エラーを返します。名前指定子の有効な文字列値は、拡張機能に使用されるものと同じです。

- enterprise-id- オプションまたはサブオプションの後で、指定された enterprise-id を持つオプションまたはサブオプションのインスタンスを選択します。エンタープライズ ID は、整数または文字列に評価する必要があるidまたはname引数として指定できます。
- vendor- オプションまたはサブオプションの後で、オプションのデータをデコードするためにベンダーのカスタム・オプション定義を使用することを要求します。DHCPv6 オプションには適用されません。指定されたベンダ文字列に定義が存在しない場合、エラーは発行されず、オプションの標準定義が使用されます (なしの場合は BLOB と見なされます)。
- instance- 直前のオプションまたはサブオプションのn番目のインスタンスを選択します。インスタンスは 0 から始まります。(インスタンスとインスタンスカウントは、単一のリクエスト関数で一緒に使用することはできません。
- instance-count- 前のオプションまたはサブオプションのインスタンス数を返し、通常は、そのすべてのインスタンスをループ処理するために使用されます。オプションまたはサブオプションが存在しない場合は 0 を返します。
- index- 複数の値 (つまり、アドレスの配列または整数値) を含むオプションでn番目の値を選択します。インデックスは 0 から始まります。たとえば、index 0は最初の値を返し、index 1は 2 番目の値を返します。
- count- 前のオプションの関連するデータ項目の数を返し、通常はindexキーワードと共に使用して、オプションまたはサブオプションのすべてのデータ値をループします。

サブタブ (サブオプション) 指定子に定義されている唯一の文字列値サブオプション名は、リレーエージェント情報オプション (82) 用であり、復号化された DHCP パケット データ項目（551 ページ）セクションのDHCPv4 **および** BOOTP **オプション**の表にリストされています。

このrequest option関数は、要求されたオプションに応じて、データ型を持つ値を返します。これは、テーブル内のデータ型がrequest関数によって返されるデータ型にどのように対応するかを示しています。

表 *43* : *request*関数によって返されるデータ型

オプションデータ型	返されるデータ型
blob	blob
IP アドレス	4 バイトの BLOB
string	string
8 ビットの符号なし整数	uint
16 ビットの符号なし整数	uint
32 ビットの符号なし整数	uint
integer	sint
バイト値ブール型	sint=1 が true の場合は true、false の場合は null

例：

(request option 82) は relay-agent-info オプションを BLOB として返します。

(request option 82 1) は circuit-id (1) サブオプションだけを返します。

(request option 82 "circuit-id") は、 (request option 82 1) と同等です

(request option "domain-name- servers") は domain-name-servers オプションから最初の IP アドレスを返します

(request option 6 index 0) は、(request option 6 count) と同等で、IP アドレスの数を返します。

(request get-blob option "dhcp-class-identifier") は、文字列ではなく、BLOB として値を返します

(request option "IA-NA" instance 2 option "IAADDR" instance 3) は、IA-NA オプションの 3 番目のインスタンス、および IA-NA オプションにカプセル化された IAADDR オプションの 4 番目のインスタンスを返します

(request get-blob option "vendor-opts" enterprise-id 1234) はenterprise-id 1234 のオプション データの BLOB を返します

(request option "vendor-opts" enterprise-id 1234 3) は、要求されたベンダーオプションデータからサブオプション 3 を返します

DHCPv6 オプション 16 ベンダー クラス (長さ区切りフィールドを含む):

DHCPv6 メッセージのデータ:

```
00:10:00:11:00:00:00:7b:00:04:01:02:03:04:00:05:68:65:6c:6c:6f
^   ^ ^   ^ ^         ^ ^               ^ ^                  ^
|   | |   | |         | +--- field 0 ---+ +-- field 1 -------+
|   | |   | |         |
|   | |   | +---------+ enterprise-id 123(10)
|   | +---+ length 17
+---+ Option 16 Vendor-Class
```

(request option 16 enterprise-id 123) -> タイプ: blob 値: '01:02:03:04'

(request option 16 enterprise-id 456) -> タイプ: 値の設定解除: 'null'

(request get-blob option 16 enterprise-id 123) -> タイプ: blob 値: '00:00:7b:00:04:01:02:03:04:00:05:68:65:6c:6c:6c:6f'

(request option 16 enterprise-id 123 index 0) -> タイプ: blob 値: '01:02:03:04'

(request option 16 enterprise-id 123 index 1) -> タイプ: blob 値: '68:65:6c:6c:6f'

（注） DHCPv6 Option 15、User-Classは、同じように動作します。

DHCPv6 Option 17 Vendor Opts (サブオプションが含まれています):

DHCPv6 メッセージ内のデータ：

```
00:11:00:12:00:00:01:c8:00:01:00:04:0a:0b:0c:0d:00:05:00:02:01:02
^   ^ ^   ^ ^         ^ ^                     ^ ^              ^
|   | |   | |         | +---- suboption 1 ----+ +- suboption 5 -+
|   | |   | |         |
```

```
|   | |   | +----------+ enterprise-id 456(10),1c8(16)
|   | +---+ length 18
+---+ Option 17 Vendor-Opts
```

(request option 17 enterprise-id 456) -> タイプ: blob 値: '00:00:01:c8:00:01:00:04:0a:0b:0c:0d:00:05:00:02:01:02'

(request option 17 enterprise-id 0x1c8) -> タイプ: blob 値: '00:00:c8:00:01:00:00:04:0a:0b:0c:0d:00:00:05:00:02:01:02'

(request option 17 enterprise-id 123) -> タイプ: 値の設定解除: 'null'

(request option 17 enterprise-id 456 index 0) -> タイプ: blob 値: '00:00:c8:00:01:00:00:04:0a:0b:0c:0d:00:00:05:00:02:01:02'

(request option 17 enterprise-id 456 1) -> タイプ: blob 値: '0a:0b:0c:0d'

(request option 17 enterprise-id 456 2) -> タイプ: 値の設定解除: 'null'

(request option 17 enterprise-id 456 5) -> タイプ: blob 値: '01:02'

---

requestdictionary

構文：

(requestdictionary{get | put val | delete} attr)

説明：

DHCP 拡張要求ディクショナリ属性値を取得、配置、または削除します。valは属性の値で、attrは属性名です。両方とも、初期データ型に関係なく文字列に変換されます。get キーワードは、"get" のオプションではありません。

---

response

構文：

(response [get | get-blob] [relay [number]] packetfield)

説明：

応答パケットから指定されたpackefieldの値を返します。説明と有効な値は、request packetfield関数の説明と同じです。

---

response dump

構文：

(response dump)

説明：

現在の応答パケットをログファイルにダンプします。すべての式の評価がdumpキーワードをサポートしているわけではないため、未サポートの場合は無視されます。

---

response option

構文：

(response [get | get-blob] option-request)

ここで option-request は次のとおりです。

1. IPv6 -relay [n] 用のオプションのリレー メッセージ セレクタ
2. 1 つ以上のオプション句 (複数のオプションが IPv6 でのみサポートされています) - option name | id [vendor name | enterprise-id name | id] [instance n]
3. 0 個以上のサブオプション句が続く - name | id [vendor name | enterprise-id name | id] [instance n]
4. オプションの句が続く - [instance-count | count | index n]

説明：

パケットからオプションの値を返します。キーワードは、request関数のキーワードと同じです。

---

responsedictionary

構文：

(responsedictionary{get | put val | delete} attr)

説明：

DHCP 拡張応答ディクショナリ属性値を取得、配置、または削除します。valは属性の値で、attrは属性名です。両方とも、初期データ型に関係なく文字列に変換されます。get キーワードは、"get" のオプションではありません。

---

search

構文：

(search arg1 arg2 fromend)

説明：

arg1のバイト シーケンスと完全に一致するバイトのサブシーケンスのためのarg2の値を構成するバイトを検索します。見つかった場合、サブシーケンスが開始するarg2の要素のインデックスを返します(fromend引数を "true" またはその他の任意の null 以外の値に設定しない限り)。それ以外の場合は null を返します。(arg1が null の場合は 0 を返し、arg2が null の場合は null を返

します。この関数は、両方の引数に対して暗黙のas-blob変換を行います。したがって、文字列と BLOB の実際のバイト シーケンスを比較し、sints と uints は比較の目的で 4 バイトの BLOB になります。

null 以外の fromend引数は、一番右の一致するサブシーケンスの左端の要素の index を返します。

例：

(search "test" "this is a test") は 10 を返します

(search "test" "this test test test" "true") は 15 を返します

---

setq

構文：

(setqvar expr)

説明：

let関数内でのみ有効です。varは、外側のlet関数で定義されたvar1からvarnのローカル変数のいずれかでなければなりません。

例：

例についてはlet 関数を参照してください。

---

starts-with

構文：

(starts-with expr prefix-expr)

説明：

prefix-expr の値が expr の先頭と一致する場合、exprの値を返します。prefix-expr が expr より長い場合は null を返します。この関数は、prefix-expr が expr (文字列または BLOB) と同じデータ型に変換できない場合、または expr が整数に評価された場合にエラーを返します。

例：

(starts-with "abcdefghijklmnop" "abc") は "abcdefghijklmnop" を返します

(starts-with "abcdefgji" "bcd") は null を返します

(starts-with 01:02:03:04:05:06 01:02:03) は 01:02:03:04:05:06 を返します

(starts-with "abcd" (as-string 61:62)) は "abcd" を返します

(starts-with "abcd" 61:62) は null を返します

(starts-with "abcd" (to-string 61:62)) は null を返します

substring

構文：

(substring expr offset len)

説明：

オフセットから始まる式exprのlenバイトを返します。exprは文字列またはBLOB です。整数の場合は、BLOB に変換されます。結果は文字列または BLOB、またはいずれかの引数が null と評価される場合は null になります。条件：

- offsetが長さlenより大きい場合、結果は null になります。
- offset + lenはexprの終わりを超えるデータで、関数は残りのデータをexprで返します。
- offsetが 0 より小さい場合、オフセットはデータの末尾から取得されます (最後の文字は、最初の文字を参照する –0=0 なので、インデックス -1 です)。
- これはデータの先頭を越えてデータを参照し、オフセットはゼロと見なされます。

例：

(substring "abcdefg" 1 6) は "bcdefg" を返します

(substring 01:02:03:04:05:06 3 2) は 04:05 を返します

synthesize-host-name

構文：

(synthesize-host-name method namestem)

説明：

構成されたメソッド(指定されていない場合) または指定されたmethodとnamestemに基づいてホスト名を生成します。

method 引数の有効なメソッドは、DHCPv4 要求または DHCPv6 要求が処理されているかどうかによって異なります。DHCPv4 の場合、有効なメソッドは、defaultまたは v4-synthetic-name-generator の列挙値の 1 つ:address、client-id、またはhashed-client-idです。DHCPv6 の場合、有効なメソッドは、defaultまたは v6-synthetic-name-generatorの列挙値の 1 つ:duid、hashed-duid、cablelabs-device-id、またはcablelabs-cm-mac-addrです。これらの列挙メソッドの詳細については、DHCPv4 と DHCPv6 での合成名の生成 （306 ページ）を参照してください。

namestem引数は、DNS 更新構成のsynthetic-name-stem値を指定します(DNS 更新設定の作成 （317 ページ）を参照してください)。

例：

(synthesize-host-name) は "dhcp-rhfxxi5pkjp6o" を返します。

(synthesize-host-name "duid" "test") は "test-00030001010203040506" を返します

(synthesize-host-name "client-id" "test") は "test-00030001010203040506" を返します

---

to-blob

構文：

(to-blob expr)

説明：

式を BLOB に変換します。条件：

- exprは文字列に評価され、 "nn:nn:nn" 形式である必要があります。この関数は、文字列を BLOB に変換した結果である BLOB を返します。関数が文字列を BLOB に変換できない場合は、エラーを返します。
- exprは、その BLOB を返す、BLOB に評価されます。
- exprは整数に評価され、ネットワーク順で整数のバイトを表す 4 バイトの BLOB を返します。（データタイプの変換 （395 ページ） を参照）。

例：

(to-blob 1) は 00:00:00:01 を返します

(to-blob "01:02") は 01:02 を返します

(to-blob 02:03) は 02:03 を返します

---

to-ip、to-ip6

構文：

(to-ip expr)

(to-ip6 expr)

説明：

式を文字列、BLOB、または整数として IP アドレスに変換します。条件：

- 文字列は、IPv4 の場合はドット付き 10 進法 IP アドレス形式、または IPv6 の場合はコロン形式の形式でなければなりません。文字列を解析して IP アドレスに変換することによって決定された BLOB IP アドレスを返します。
- 結果は BLOB で、(to-ip ..) の最初の 4 バイトと (to-ip6 ..) の最初の 16 バイトを返します。blob が to-ip の場合は 4 バイト未満、または to-ip6 の場合は 16 バイト未満の場合、引数 BLOB のバイト数は 0 バイトで高次バイトに埋め込まれます。
- 結果は整数で、(いずれかのタイプの)整数をブロブに変換します。整数と BLOB はネットワークの順序で並べ替えるため、順序の変更は必要ありません。

---

to-lower

構文：

(to-lower expr)

説明：

文字列を受け取り、小文字の文字列を生成します。client-lookup-id 属性を使用して、クライアント指定子を計算して、(LDAP ではなく) CNRDB ローカル ストア内の client-entry を検索する場合、結果の文字列は小文字である必要があります。この関数を使用すると、client-lookup-id の結果を小文字の文字列に簡単に作成できます。client-lookup-id を使用して LDAP にアクセスする場合、この機能を使用する場合と使用しない場合があります。

---

to-sint

構文：

(to-sint expr)

説明：

式を符号付き整数に変換します。

exprが文字列に評価される場合、符号付き整数に変換できる形式である必要があります。条件：

- exprが 1 ~ 4 バイトの BLOB に評価される場合、関数はそれを符号付き整数として返します。
- exprが 4 バイトを超える長さの BLOB に評価される場合、エラーを返します。
- exprが符号なし整数に評価される場合、符号なし整数の値が最大の正符号付き整数より大きい場合を除き、同じ値の符号付き整数を返します。
- exprが符号付き整数に評価される場合、その値を返します。

例：

(to-sint "1") は 1 を返します

(to-sint -1) は -1 を返します

(to-sint 00:02) は 2 を返します

(to-sint "00:02") はエラーを返します

(to-sint "4294967295") は 2147483647 を返します

---

to-string

構文：

(to-string expr)

説明：

式を文字列に変換します。exprが文字列に評価された場合は、その文字列を返します。BLOBまたは整数の場合は、その印字可能な表記を返します。すべての値が印字可能な表記であるため、expr自体がエラーなしで評価された場合、エラーは返されません。

例：

(to-string "hello world") は "hello world" を返します

(to-string -1) は "-1" を返します

(to-string 02:04:06) は "02:04:06" を返します

---

to-uint

構文：

(to-uint expr)

説明：

式を符号なし整数に変換します。条件:

- exprが文字列に評価される場合、符号なし整数に変換できる形式である必要があります。
- exprが 1 ~ 4 バイトの BLOB に評価される場合、符号なし整数として返されます。
- exprが 4 バイトよりも長い blob に評価される場合、エラーを返します。
- exprが符号付き整数に評価される場合、符号付き整数の値が 0 未満でない限り、同じ値の符号なし整数を返します。
- exprが符号なし整数に評価される場合、関数はその値を返します。

例：

(to-uint "1") は 1 を返します

(to-uint 00:02) は 2 を返します

(to-uint "4294967295") は 4294967295を返します

(to-uint "00:02") はエラーを返します

(to-uint -1) はエラーを返します

---

translate

構文：

(translate expr search replace)

説明：

文字列またはBLOBのシーケンスに回避する式を引数として受け取り、searchに表示されるさまざまな文字またはバイトをreplaceの対応する値 (同じ位置) に置き換えます。条件：

- exprが文字列またはBLOBである場合、値はそのまま残され、それ以外の場合は強制的に文字列になります。処理後にexprが文字列である場合、search と replace は文字列である必要があります。
- expr が BLOB である場合、search と replace の両方が BLOB である必要があります。
- replace が search より短い場合、replaceに対応するバイトまたは文字がない search 内のバイトまたは文字は出力からドロップされます。
- replace が表示されない場合、search のバイトまたは文字はすべて expr から削除されます。

例：

(translate "Hello apple and eve" "abcdef" "123456") は "H5llo 1ppl5 1n4 5v5" を返します

(translate "a&b$c%d" "%$&") は "abcd" を返します

---

try

構文：

(try expr failure-expr)

説明：

評価中にエラーが検出されなかった場合、exprを評価し、その評価の結果を返します。exprの評価中にエラーが発生した場合は、次の手順を実行します。

- failure-expr があり、エラーなしで評価された場合、try関数の結果としてその評価の結果を返します。
- failure-expr があり、関数が failure-expr を評価中にエラーが発生した場合、 エラーを返します。
- failure-expr がない場合、try は null を返します。

例：

(try (try (expr) (complex-failure-expr)) "string-constant") は外側の try がエラーを返さないことを保証します（"string-constant" の評価は失敗できないため）

(try (error) 01:02:03) は常に 01:02:03 を返します

(try 1 01:02:03) は常に 1 を返します

(try (request option 82) "failure") は "failure"を返しません。（request option 82）は、パケットに option-82 がなく、エラーを返さない場合に null になるためです。

(try (request option "junk") "failure") は "junk" が有効な option-name ではないため、"failure" を返します。

---

unparse

構文：

(unparse expr1 expr2 [expr3])

説明：

expr2 で指定されたデータ型として BLOB expr1 を解析した結果の文字列を返します。expr3 で指定されたとおりに変更されることがあります。expr1 が BLOB でない場合は、BLOB に変換されます。expr2 は、Cisco Prime Network Registrar でサポートされる AT_* data types（文字列またはその数値）のいずれかである必要があります（オプションの検証タイプ（548 ページ）を参照してください）。expr3 はオプションで、値は「none」、「alternate」、または「feature」で、動作は expr2 に依存します。たとえば、AT_BOOL タイプの場合、「feature」は「enabled」または「disabled」を返し、「alternate」は「on」または「off」を返し、「none」（または no expr3）は「true」または「false」のいずれかを返します。

この機能は、Cisco Prime Network Registrar 11.0 で導入されました。

例：

(unparse 00 "AT_BOOL" "feature") は disabled を返します。

(unparse 05:63:69:73:63:67:03:63:6f:6d:00 "AT_DNSNAME") は 「cisco.com」を返します。

---

validate-host-name

構文：

(validate-host-name hostname)

説明：

hostname 文字列を受け取り、検証済みのhostnameを返します。これは、入力 hostname と同じか、次のように変更できます。

- ハイフンに割り当てられたスペースと下線付き文字。
- 無効なhostname文字を削除。有効な文字は a ～ z、A ～ Z、0 ～ 9、およびハイフンです。
- Null ラベルを削除（「..」が「.」に変更される）。
- hostname の各ラベルは 63 文字に切り捨てられます。

例：

(validate-host-name "a b c d e f") は "a-b-c-d-e-f" を返します

(validate-host-name "_a_b_c_d_e_f_") は "a-b-c-d-e-f" を返します

(validate-host-name "abcdef") は "abcdef" を返します

(validate-host-name "a&b*c#d@!e()f") は "abcdef" を返します

オプションに対して式を使用する

Cisco Prime Network Registrar 11.0 以降では、式を使用して、オプションに値を返すことができます（DHCPv4 および DHCPv6）。

オプションに式を使用する場合は、次の点に注意してください。

- オプションインスタンスには、固定値または式を指定できますが、両方使用することはできません（ただし、式は固定値を返すことは可能）。
- 式であるオプションインスタンスは、そのオプションがクライアント要求の応答に追加されるたびに評価されます。
- 式であるオプションインスタンスは、リースクエリ（unitary、bulk、active）では評価されません（返されません）。これは、式を評価するためのコンテキストが使用できないためです。
- オプションの式は、次のいずれかを返す必要があります。
  - Null 値：この場合、オプションは応答に追加されません。
  - <none> の値 （大文字と小文字を区別しない）：この場合、オプションは応答に追加されません。
  - BLOB 値：この場合、値はこのオプションとして返されます。これは完全なオプションデータである必要があります。ベンダーオプション（DHCPv4 オプション 125 や DHCPv6 オプション 17 など）の場合は、最初の 4 バイトに企業 ID を含める必要があります。
  - 文字列値：この場合、値はオプションの定義に基づいて解析され、解析された値が返されます。解析が失敗した場合、オプションは応答に追加されません。

  結果に関係なく式を評価した後、サーバーはオプションの他のインスタンスのポリシー階層の検索を続行しないことに注意してください。
- 式追跡設定がオプション式に適用されます。
- オプションは予測できない順序で応答に追加されるため、式であるオプションそして、応答ディクショナリの他のポイントの値として使用するオプションは、予測できない結果が生じる場合があるため、推奨しません。
- オプション値のラウンドロビンは、式であるオプションで使用されます。式の結果の値はラウンドロビンされます。
- NULL 値によって、オプションが応答に追加されないため、式であるオプションは、長さが 0 のオプション値を生成できません。

（注） DHCPv4 オプション、dhcp-lease-time (51)、dhcp-renewal-time (58) および dhcp-rebinding-time (59) は、式ではサポートされていません。これらは値で設定する必要があります。式で設定されている場合、DHCP サーバーはこのオプションを無視します。

CLI の場合、**ポリシー**のヘルプには、オプションインスタンスを式として設定する方法の詳細が含まれています。

式を使用して、サブスクライバーにリースされるIPアドレスを制限する

これらの例では、クライアントを制限する、制限しないもの、および構成制限を超えて、クライアントクラスの制限超過に割り当てる必要があるものを設定します。クライアントの3つのクラスのそれぞれに、それぞれスコープと選択タグがあります。これらの例では、次のCisco Primeネットワーク レジストラー設定環境を想定しています(これは実際の環境とは異なり、図のためだけに使用されます)。

- Client-classes—制限、制限なし、および制限超過。
- Scopes10.0.1.0 (プライマリ)、10.0.2.0、10.0.3.0(セカンダリ)、サブネットの名前。
- Selection tags—制限タグ、制限なしタグ、および制限超過タグ。スコープは、それらが表すアドレス プールの名前が付けられます。選択タグは、範囲に割り当てられ、10.0.1.0 は制限タグ、10.0.2.0 は無制限タグ、10.0.3.0 は制限を超えるタグを取得します。

関連項目

制限事例 1: DOCSIS ケーブル モデム （427 ページ）

制限事例 2: 拡張 DOCSIS ケーブル モデム （428 ページ）

制限事例 3: 非同期転送モードでの DSL （429 ページ）

制限事例 1: DOCSIS ケーブル モデム

テストは、デバイスが DOCSIS ケーブル モデムと見なされるかどうかを判断し、各ケーブル モデムの背後にあるカスタマーデバイスの数を制限することです。クライアントクラスの制限 ID は、リレー エージェント情報オプションのremote-idサブオプションに含まれるケーブル モデムの MAC アドレスです。

サーバー上のクライアント・クラス・ルックアップ ID属性の式は、次のとおりです。

```
// Expression to set client-class to no-limit or limit based on remote-id
(if (equal (request option "relay-agent-info" "remote-id")
           (request chaddr))
```

```
  "no-limit"
  "limit")
```

上記の式は、relay-agent-infoオプションのremote-idサブオプション (2) の内容がパケットのchaddrと同じである場合、クライアントクラスは制限なしであることを示しています。

制限クライアントクラスの制限id式は次のとおりです。

```
(request option "relay-agent-info" "remote-id")
```

この式は、次の手順で使用します。

ステップ 1 クライアント クラスを定義します。

ステップ 2 スコープ、範囲、およびタグを定義し、それらがプライマリまたはセカンダリの場合に定義します。各スコープのホスト範囲は、すべてのホスト番号が同じである場合よりも、誤読される可能性が低いことを確認します。

ステップ 3 制限数を定義します。これは、デフォルトのポリシーに入ることができます。リクエストに制限 ID が表示されない場合、カウントはチェックされません。

ステップ 4 次の目的で、式ファイル cclookup1.txt に式を追加します。

```
// Expression to set limitation count based on remote-id
(if (equal (request option "relay-agent-info" "remote-id")
           (request chaddr))
  "no-limit"
  "limit")
```

ステップ 5 サーバー レベルでクライアント クラスの検索 ID属性を設定する場合は、式ファイルを参照してください。

ステップ 6 クライアントの制限 ID に対する別の式を cclimit1.txt ファイルに追加します。

```
// Expression to set limitation ID based on remote-id
(request option "relay-agent-info" "remote-id")
```

ステップ 7 クライアントクラスの制限 id属性を設定する際は、この式ファイルを参照してください。

ステップ 8 サーバーをリロードします。

以前に使用されていない構成に対してこれを行うと、最初の 2 つの DHCP クライアントに共通のremote-idオプション 82 サブオプション値が設定されます。同じ値を持つ 3 番目のクライアントは、クライアント クラスの制限超過に入ります。サブスクライバが制限なしクライアント クラスに持つことができるデバイスの数には制限はありません。MACアドレスがremote-idサブオプションの値と等しいデバイスは、制限の目的で無視され、制限 ID が設定されていない制限なしクライアント クラスに入ります。

制限事例 2: 拡張 DOCSIS ケーブル モデム

この例は、制限事例 1: DOCSIS ケーブル モデム （427 ページ）で説明されている例の拡張です。後者の例では、デフォルトポリシーに対して制限数が2つ定義されているため、すべてのケーブル モデムがクライアント デバイスを 2 つ超えるだけで済みます。この例では、制限タ

グ選択タグを使用するスコープとは異なる数のデバイスに IP アドレスを付与できるように、特定のケーブル モデムを設定しています。

この場合、クライアントクラスデータベースで、2 つ以上のアドレスを持つケーブルモデムを明示的に設定する必要があります。この場合、Cisco Prime Network レジストラーまたは LDAP データベースでケーブル モデムのクライアント エントリを検索できるように、サーバー全体でのクライアント クラス処理を有効にする必要があります。ケーブル モデムが見つからならない場合、デバイスの数は2に制限されます。この検出では、ケーブルモデムに設定されたポリシーの制限数が使用されます。

この例では、5つのデバイスを許可する 5 つの追加ポリシーが必要です。

ステップ 1 サーバー全体でクライアントクラスの処理を有効にします。

ステップ 2 5つのデバイスの制限数を持つ 5 つのポリシーを作成します。

ステップ 3 前の例と同様に、式を使用して、制限クライアントクラスの制限 ID を設定します。制限 ID を cclimit2.txt ファイルに、ルックアップ ID を cclookup2.txt ファイルに入れます。

```
cclimit2.txt file:
// Expression to set limitation ID
(request option "relay-agent-info" "remote-id")

cclookup2.txt file:
// Expression to set client-class lookup ID
(concat "1,6," (to-string (request option "relay-agent-info" "remote-id")))
```

ステップ 4 適切な属性を設定する際には、これらのファイルを参照してください。

ステップ 5 いくつかのケーブル モデム クライアントを定義し、5つのポリシーを適用します。

ステップ 6 サーバーをリロードします。

制限事例 3: 非同期転送モードでの DSL

この例では、式を使用して、非同期転送モード(ATM)ルーティングブリッジカプセル化(RBE)を使用してサービス プロバイダへの加入者のデジタル加入者線 (DSL) アクセスを構成する方法を示します。サービス プロバイダは、DSL サブスクライバーを構成する ATM RBE を使用するようになっています。Cisco IOS Release 12.2(2)T よりルーテッド ブリッジ カプセル化機能の DHCP オプション 82 サポートされるようになり、サービス プロバイダは DHCP を使用して IP アドレスを割り当てられるようになったほか、オプション 82 を使用してセキュリティおよび IP アドレス割り当てポリシーを実装できるようになりました。

このシナリオでは、DSL サブスクライバは Cisco 7401ASR ルータの個々の ATM サブインターフェイスとして識別されます。各顧客はルータに独自のサブインターフェイスを持ち、各サブインターフェイスには独自の仮想チャネル識別子(VCI)と仮想パス識別子(VPI)があり、ATM スイッチを通過する ATM セルの次の宛先を識別します。7401ASR ルータは、Cisco 7206 ゲートウェイ ルータにルーティングします。

ステップ 1 IOS を使用して、ルータの DHCP サーバーとインターフェイスを設定します。これは典型的な IOS 設定です:

```
Router#ip dhcp-server 170.16.1.2
Router#interface Loopback0
Loopback0(config)#ip address 11.1.1.129 255.255.255.192
Loopback0(config)#exit
Router#interface ATM4/0
ATM4/0(config)#no ip address
ATM4/0(config)#exit
Router#interface ATM4/0.1 point-to-point
ATM4/0.1(config)#ip unnumbered Loopback0
ATM4/0.1(config)#ip helper-address 170.16.1.2
ATM4/0.1(config)#atm route-bridged ip
ATM4/0.1(config)#pvc 88/800
ATM4/0.1(config)#encapsulation aal5snap
ATM4/0.1(config)#exit
Router#interface Ethernet5/1
Ethernet5/1(config)#ip address 170.16.1.1 255.255.0.0
Ethernet5/1(config)#exit
Router#router eigrp 100
eigrp(config)#network 11.0.0.0
eigrp(config)#network 170.16.0.0
eigrp(config)#exit
```

ステップ 2 IOS で、システムが Cisco IOS DHCP サーバーに転送される BOOTREQUEST メッセージに DHCP オプション 82 データを挿入できるようにします。

```
Router#ip dhcp relay information option
```

ステップ 3 IOS で、オプション 82 remote-idサブオプション(2)を使用して DHCP サーバーに送信される DHCP リレーエージェントのループバック インターフェイスの IP アドレスを指定します。

```
Router#rbe nasip Loopback0
```

ステップ 4 Cisco Prime Network レジストラーで、サーバー全体でのクライアント クラスの処理を有効にします。

ステップ 5 1つのデバイスの制限数を持つ 1 つのポリシーを作成します。

ステップ 6 パケットを適切なクライアントクラスに配置します。すべてのパケットは、クライアントクラスの制限内にあるべきです。値limitのみを含むルックアップ・ファイルを作成し、クライアント・クラスのルックアップ ID を設定します。cclookup3.txt ファイルで次の操作を行います。

```
// Sets client-class to limit
"limit"
```

ステップ 7 式を使用して、制限されたパケットに正しい制限IDがあることを確認します。ファイルに式を入れ、そのファイルを参照して制限 ID を設定します。サブストリング関数は、オプション 82 サブオプション 2 (remote-id) データ・フィールドのバイト 10 から 12 を抽出することによって VPI/VCI を取得します。cclimit3.txt ファイルで次の手順を実行します。

```
// Sets limitation ID
(substring (request option 82 2) 9 3)
```

ステップ 8 サーバーをリロードします。

デバッグ式

式に問題がある場合は、サーバー起動時に DHCP ログ ファイルを調べます。すべての式は、関数の入れ子を明確にするような形で印刷され、意図を確認するのに役立ちます。特に、ログファイルに出力された式をコピーして、エディタに貼り付けることができます。各行の先頭から文字を削除すると、結果の式が正しく入力されます (読み取りや変更が非常に簡単になります)。関数と引数のequalデータ型変換に特に注意してください。引数が同じデータ型でない場合、to-string関数と同様のコードを使用して文字列に変換されます。

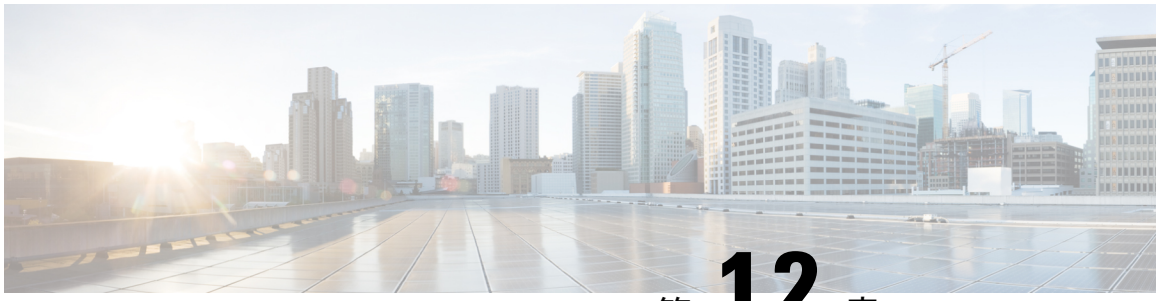
DHCP サーバーの式トレース レベル属性を使用して、式のさまざまなデバッグ レベルを設定できます。実行されたすべての式は、属性によって設定された次数までトレースされます。最高のトレース レベルは 10 です。レベルを少なくとも 2 に設定すると、失敗した式はレベル 10 で再試行されます。

式トレース・レベルのトレース・レベルは次のとおりです (数値を使用)。

- 0— トレースなし
- 1— 失敗、(tryによって保護されたものを含む)
- 2— 失敗の再試行の合計 (再試行のトレース レベル = 6)
- 3— 関数呼び出しと戻り値
- 4— 関数の引数が評価される
- 5— 関数の引数を印刷する
- 6—データ型変換(すべて)

構成に問題がある式をトレースするために、式構成トレース・レベル属性も存在し、1 から 10 までの任意のレベルに設定できます。レベルを 2 以上に設定すると、構成されていない式はレベル 6 に設定して再試行されます。番号付けのギャップは、将来のレベルの追加に対応するためです。式構成トレース・レベルのトレース・レベルは次のとおりです (number 値を使用)。

- 0— 追加のトレースなし
- 1— 追加のトレースなし
- 2— 失敗の再試行（デフォルト）
- 3— 関数定義
- 4— 関数の引数
- 5— 変数の検索とリテラルの詳細
- 6—すべて



第 12 章

拡張ポイントの使用

拡張は、Cisco Prime Network Registrar が DHCP 要求をどのように処理し、応答するかに影響を与えたり、通常はユーザー インターフェイスを使って行うことができない DHCP サーバーの動作を変更したりするように記述することができます。この章では、DHCPv4 および DHCPv6 の拡張を添付できる拡張ポイントについて説明します。

拡張機能の使用

Tcl または C/C++ で記述できる拡張機能、機能を使用して、Cisco プライムネットワーク レジストラー DHCP サーバーの動作を変更およびカスタマイズできます。

DHCP サーバーで使用する拡張機能を作成するには、次の手順に従います。

1. 実行するタスクを決定します。どの DHCP パケット プロセスを変更しますか?
2. 使用するアプローチを決定します。パケットプロセスを変更する方法を教えてください。
3. 拡張機能をアタッチする拡張ポイントを決定します。
4. 言語 (Tcl または C/C++) を選択します。
5. 拡張機能を書き込む (また、コンパイルとリンクも可能です)。
6. DHCP サーバー構成に拡張機能を追加します。
7. 拡張ポイントに拡張子をアタッチします。
8. DHCP サーバーをリロードして、拡張を認識します。
9. 結果をテストしてデバッグします。

（注） Cisco Primeネットワーク レジストラーをアップグレードする際は、すべての DHCP C/C++ 拡張(dex エクステンション)を再コンパイルすることをお勧めします。

関連項目

拡張機能の作成、編集、および添付 （434 ページ）
タスクの決定 （435 ページ）
アプローチの決定 （436 ページ）
拡張言語の選択 （436 ページ）

拡張機能の作成、編集、および添付

拡張機能を作成、編集、および添付できます。

拡張機能ポイントごとに複数の拡張機能を関連付けることができます。各拡張機能は、添付ファイルの作成時に使用されたシーケンス番号で指定された順序で実行されます。Web UI では、拡張が [DHCP 拡張ポイントのリスト（List DHCP Extension Points）] ページに拡張ポイントごとに表示される順序。CLI では、シーケンス番号の値をコマンドと共にdhcp attachExtension使用します。

拡張ポイントごとの複数の拡張機能の詳細については、「」を参照してください複数の拡張機能に関する考慮事項 （439 ページ）。

ローカル アドバンスド Web UI

拡張機能を作成して添付するには、次の操作を行います。

ステップ 1 メニューからDeployExtensions[DHCP]サブメニューの下で[DHCP拡張のリスト/追加]ページを開きます。

ステップ 2 アイコンをAdd Extensionsクリックして、[DHCP サーバー拡張の追加] ダイアログ ボックスを開きます。

ステップ 3 拡張機能を作成した後、このページの 1 つ以上の拡張ポイントに添付できます。拡張機能をアタッチできる拡張ポイントを表示するには、[DHCP 拡張の一覧/追加] ページでDHCP Extension Pointsタブをクリックします。

ステップ 4 各拡張ポイントに複数の拡張子をアタッチする場合は、矢印キーをクリックしてエントリを並べ替えることで、その拡張子が処理される順序を変更できます。拡張を削除するには、[削除（Delete）]アイコンをクリックします。

CLI コマンド

このコマンドextensionを使用するには、次の構文が必要です。

```
nrcmd> extension name create language extension-file entry-point
```

エントリ ポイントは、拡張子ファイル内のエントリ ポイントの名前です。また、DHCP サーバーがファイルをロードするたびに、初期エントリ ポイントに対してオプションのinit-entry属性インイット・エントリー（468ページ）値を設定することもできます(を参照)。この関数は、このモジュールにバインドされている任意の拡張ポイントから呼び出すことができます。拡張機能を 一extension list覧表示することもできます。

拡張機能をアタッチしてデタッチするには、dhcp attachExtension次dhcp detachExtensionの構文が必要な DHCP サーバーを使用します。

```
nrcmd> dhcp attachExtension extension-point extension-name [sequence-number]
nrcmd> dhcp detachExtension extension-point [sequence-number]
```

シーケンス番号は、拡張ポイントごとに複数の拡張をアタッチする場合に適用され、シーケンスの順序は 1 から 32 まで増加します。省略した場合、デフォルトは 1 になります。

現在登録されている拡張機能を表示するには、dhcp listExtensionsコマンドを使用します。

関連項目

拡張機能の使用 （433 ページ）

タスクの決定

拡張を適用するタスクは、通常、環境のニーズを満たすように、DHCPサーバー処理の変更です。要求の受信からクライアントへの応答まで、これらのDHCPサーバーの各処理ポイントで拡張機能を適用できます。

1. パケットを受信してデコードします。
2. クライアント クラスを検索、変更、および処理します。
3. 応答の種類を作成します。
4. サブネット (DHCPv6 の場合はリンク) を決定します。
5. 既存のリースを検索します。
6. リース要求をシリアル化します。
7. クライアントのリース受け入れ可否を決定します。
8. 応答パケットを収集し、エンコードします。
9. パケットの安定したストレージを更新します。
10. パケットを返します。

これらの手順の詳細な一覧 (各ステップで使用する拡張ポイント) が拡張を使用した DHCP 要求処理 （446 ページ）に表示されます。

たとえば、BOOTP 構成を使用する異常なルーティング ハブがある場合があります。このデバイスは、イーサネット・ハードウェア・タイプ (1) およびMACアドレスを指定したBOOTP要求をchaddrフィールドに出します。その後、同じ MAC アドレスを持つ別の BOOTP 要求を送信しますが、ハードウェアタイプはトークンリング (6) です。2 つの異なるハードウェアの種類を指定すると、DHCPサーバーは2つのIPアドレスをデバイスに割り当てます。通常、DHCP サーバーは、ハードウェア タイプ 1 の MAC アドレスとタイプ 6 の MAC アドレスを区別し、

異なるデバイスと見なします。この場合、DHCP サーバーが同じデバイスに 2 つの異なるアドレスを渡すことを防ぐ拡張機能を作成できます。

アプローチの決定

多くの場合、単一の問題に対して多くのソリューションが使用できます。書き込む拡張子の種類を選択する場合は、まず入力DHCPパケットを書き換えることを検討する必要があります。DHCP サーバーの内部処理を知る必要がないため、これは良いアプローチです。

でタスクの決定 （435ページ）説明する問題については、次のいずれかの方法で拡張機能を記述して解決できます。

- トークン リング (6) ハードウェア タイプ パケットをドロップします。
- パケットをイーサネット パケットに変更し、終了時に再度スイッチを戻します。

2 番目の方法では、より複雑な拡張が必要ですが、DHCP クライアントは DHCP サーバーからの応答を使用できます。2 番目の方法では、パケットの書き換えがpost-packet-encode行われます(このポストパケットエンコード （482ページ）場合は、拡張ポイントを使用します)。他の方法では、他の拡張と拡張ポイントが必要です。

拡張言語の選択

Tcl または C/C++ で拡張機能を記述できます。DHCP サーバーに関する限り、各言語の機能は似ていますが、アプリケーション プログラミング インターフェイス (API) は言語設計の 2 つの非常に異なるアプローチをサポートするために若干異なります。

- Tcl— Tcl でのスクリプトは C/C++ でのスクリプトよりもやや簡単ですが、解釈され、シングルスレッドで、より多くのリソースが必要になる場合があります。ただし、C/C++ よりも深刻なバグが発生する可能性が低く、サーバー障害の可能性も低くなります。Cisco プライムネットワーク レジストラーは、Tcl バージョン 8.6 を現在サポートしています。
- C/C++：この言語では、外部プロセスとの通信を含む、可能な限り最大のパフォーマンスと柔軟性を実現できます。ただし、C/C++ API は Tcl API よりも複雑です。また、C/C++ では、拡張機能のバグが原因でサーバー障害が発生する可能性も高くなります。

言語に依存しない API

以下の概念は、Tcl または C/C++ で拡張機能を記述するかどうかに依存しません。

関連項目

ルーチン署名 （437 ページ）

Dictionaries （437 ページ）

ディクショナリでのユーティリティ メソッド （438 ページ）

設定エラー （438 ページ）

ルーチン署名

ファイル内に、複数の拡張関数を含めることができるルーチンとして、拡張機能を定義する必要があります。次に、1 つ以上の DHCP サーバー拡張ポイントに拡張機能を接続します。DHCP サーバーは、その拡張ポイントに到達すると、拡張機能が定義するルーチンを呼び出します。ルーチンは成功または失敗を返します。拡張エラー時にパケットをドロップするように DHCP サーバーを構成できます。

構成された各拡張機能に異なるエントリ ポイントを指定することで、1 つのファイル (Tcl ソース ファイル、C/C++ .dll または .so ファイル) を DHCP サーバーに対して複数の拡張子として構成できます。

サーバーは、少なくとも 3 つの引数 (要求、応答、および環境) の 3 つのディクショナリを使用して、すべてのルーチン エントリ ポイントを呼び出します。各ディクショナリには、キーと値のペアである、多くのデータ項目が含まれています。

- この拡張機能は、特定のデータ項目のディクショナリに対して get メソッドを実行することで、DHCP サーバーからデータ項目を取得できます。
- この拡張機能は、同じ名前付きデータ項目の多くについて、put 操作または remove 操作を実行してデータ項目を変更できます。

すべての拡張ポイントですべての辞書を使用することはできませんが、すべてのルーチンの呼び出しシーケンスは、すべての拡張ポイントで同じです。特定の拡張ポイントに存在しないディクショナリを参照しようとすると、拡張機能でエラーが発生します。（拡張ディクショナリ （460 ページ） を参照。）

Dictionaries

要求、応答、およびサーバーのデータには、ディクショナリ インターフェイスを介してアクセスします。拡張ポイントには、要求、応答、環境という 3 種類のディクショナリが含まれています。

- Request — DHCP 要求に関連付けられた情報と、要求自体に含まれるdictionaryすべての情報。データは、文字列、整数、IP アドレス、および BLOB 値です。
- Response DHCP クライアントに返す DHCP 応答パケットの生成に関連dictionaryする情報。データは、文字列、整数、IP アドレス、および blob の値です。
- Environment — DHCP サーバーと拡張の間で渡dictionaryされる情報。

辞書の説明については、を参照してください拡張ディクショナリ （460 ページ）。

環境ディクショナリを使用して、異なる拡張ポイントにアタッチされた拡張機能間で通信することもできます。拡張が構成されている最初の拡張ポイントが検出されると、DHCP サーバーは環境ディクショナリを作成します。環境ディクショナリは、DHCP サーバーが許容される

データ項目の名前を修正しない唯一のディクショナリです。環境ディクショナリを使用して、文字列値のデータ項目を挿入できます。

DHCP クライアントの要求と応答の間の制御フロー内のすべての拡張ポイント (変更の原因にlease-state-change応じて、 を除くすべての拡張ポイント) は、同じ環境ディクショナリを共有します。したがって、拡張は何らかの条件が存在することを判断し、環境辞書にセンチネルを置いて、後続の拡張が同じ条件を決定するのを避けることができるようにすることができます。

前の例では、post-packet-decode拡張ポイントの拡張は、特定の製造元のデバイス、BOOTP、およびトークン リングから、特定の製造元のデバイスからは、パケットが対象となっていたと判断し、トークン リングからイーサネットにハードウェアの種類を書き換えます。また、環境ディクショナリに sentinel を配置し、post-packet-encode拡張ポイントの非常に単純な拡張で、ハードウェアの種類をトークン リングに書き換えます。

ディクショナリでのユーティリティ メソッド

各ディクショナリには、拡張のトレース レベルをリセットし、出力ファイルに値を記録できるユーティリティ メソッドが関連付けられます。

設定エラー

拡張機能は、さまざまな理由で失敗する可能性があります。次に例を示します。

- サーバーはファイルを見つけることができません。
- エントリ ポイントまたはinit-entryエントリ ポイントは、ファイルに表示されません。
- 拡張機能自体は、呼び出しからinit-entryエラーを返すことができます。

それ自体では、拡張エラーは致命的ではなく、DHCPサーバーの起動を妨げません。ただし、任意の拡張ポイントで失敗した拡張機能を構成した場合、サーバーは起動しません。したがって、構成プロセスをデバッグするには、拡張init-entryポイントにアタッチせずに、その時点で拡張機能インイット・エントリー （468 ページ）を構成できます(を参照)。このプロセスが正常に完了したら、拡張機能を拡張ポイントにアタッチできます。

外部サーバーとの通信

外部サーバーまたはデータベースと通信する拡張機能を作成して、クライアント クラスに影響を与えたり、着信DHCPクライアント要求を検証したりできます。このような拡張機能を記述することは複雑な作業であり、かなりのスキルとデバッグの専門知識が必要です。このような拡張機能はマルチスレッド化する必要があり、DHCPサーバーのパフォーマンスが許容レベルに維持される場合は、外部サーバーと非常に迅速に通信する必要があります。

パフォーマンスの低下は、要求を処理しているスレッドを停止拡張機能が原因で発生する可能性があります。拡張機能が外部サーバーと通信している間、スレッドが停止します。この対話に 50～100 ミリ秒以上かかる場合、サーバーのパフォーマンスに大きく影響します。この拡張機能を展開する特定の環境では、この影響を受ける場合と影響を与えない場合があります。

外部サーバーとの通信を同期化する (つまり、外部サーバーとの通信のために着信 DHCP クライアント要求処理が停止する) ことを回避する 1 つの方法は、DHCP クライアント要求の処理中にこの通信を実行しないようにすることです。. これは明らかに聞こえるし、それはまた、それの顔に、不可能に聞こえる。ただし、DHCPクライアントサーバープロトコルの性質上、外部サーバーへのアクセスを DHCP クライアント要求処理から切り離す方法があります。

このボトルネックを回避するには、拡張機能の一部としてキャッシュ メカニズムを使用します。サーバーが要求に対して拡張機能を呼び出すときは、クライアントデータのキャッシュをチェックし(マルチスレッドの問題を回避するために適切なロックを使用して) します。クライアントが次の場合:

- キャッシュ内(および有効期限がない)では、キャッシュ内のデータに応じて、要求を受け入れるか拒否するかを拡張機能に依頼します。
- キャッシュ内に存在しない場合は、拡張キューに外部サーバーへの要求をキューに入れ(できればUDP経由で)、DHCPクライアント要求をドロップします。クライアントが要求を再送信する時点で、データはキャッシュ内に格納されます。

このキャッシング・メカニズムでは、拡張機能に受信側スレッド(init-entry拡張ポイントで開始および停止) が必要です。このスレッドは、ソケットを読み取り、応答でキャッシュを更新します。このスレッド(または別のスレッド)もタイムアウトし、キャッシュから古い項目を削除する必要があります。ただし、単一スレッドを使用する場合は、より大きな受信ソケット・バッファー・サイズの設定が必要になる場合があります。

これらの方法は、DHCPサーバーの負荷が高く、外部サーバーの速度が十分でない場合にのみ必要です。しかし、この状況は実際にはあまりにも一般的であることが判明しました。また、外部サーバーに到達できない場合(接続タイムアウトが秒ではなく分数の場合)に何が起こるかを考慮してください。

拡張機能の認識

DHCPサーバーは、最初に起動時または再ロード時に自身を構成する場合にのみ、拡張機能を認識します。拡張機能または拡張機能の構成は、一般的に変更できます。ただし、サーバーをリロードまたは再起動するまでは、変更は無効です。DHCPサーバーの再読み込みを忘れることは、拡張機能のデバッグ中に頻繁に発生するエラーの原因になることがあります。

Cisco Prime Network レジストラーでリロードが必要な理由は、エクステンションを事前にロードし、サーバー設定時に準備することで、処理への影響を最小限に抑えるためです。この方法は実稼働モードでは便利ですが、拡張機能をデバッグするときには、ある程度の不満が生じる可能性があります。

複数の拡張機能に関する考慮事項

任意の拡張ポイントで複数の拡張機能を登録できます。DHCPサーバーは、処理を再開する前に、拡張ポイントに接続されているすべての拡張機能を実行します。

- 拡張機能が明示的にデータ項目を設定しない限り、拡張機能は明示的にデータ項目を設定しないでください。たとえば、(表 31-25 の表 31-5 のドロップ環境ディクショナリデータ

項目について説明されているように)、拡張機能は、ほとんどの拡張ポイントでクライアント・パケットのドロップを要求できます。

サーバーは、ドロップ・セットが False の拡張ポイントで登録された最初の拡張を呼び出します。1 つ以上の拡張機能を True または False に設定できます。すべての拡張機能が明示的にドロップを True または False に設定した場合、サーバーは最後に実行された拡張機能が要求した任意のアクションを実行します。

これは望ましい動作ではない場合があります。したがって、このデータ項目の場合、パケットをドロップする場合にのみ、拡張機能がドロップを True に設定する方が良いでしょう。このようにすれば、すべての拡張機能がこの規則で再生された場合、いずれかのエクステンションが要求した場合にパケットがドロップされます。

- 別の拡張機能がパケットの破棄を望む場合、その処理を行う必要がなくなる可能性があるため、ドロップが True の場合は、拡張機能がすぐに返される場合があります。
- 後の拡張ポイントで使用する項目を格納するために環境ディクショナリを使用する場合、それらのデータ項目名は、その拡張機能に固有の接頭辞またはサフィックスを使用する必要があります。これにより、データ項目名の競合が発生する可能性が低くなります。
- 少なくとも 1 つの環境ディクショナリ データ項目、リース (DHCPv4 の場合) またはクライアント (DHCPv6) を使用してデータを格納するために使用できるユーザー定義データ(表 48 : 一般的な環境ディクショナリ データ項目を参照) には、特別な注意が必要です。

  これらの拡張機能が互いの値を保持し、認識するために特別な注意を払っていない限り、このデータ項目を複数の拡張機能を使用するのは困難な場合があります。したがって、複数の拡張機能がこのデータ項目を使用できると想定することはできません。

- 拡張機能を最初に実行するか、必要に応じて最後に実行するかを指定する必要があります。たとえば、サーバーが最初に特定のパケットをドロップする拡張機能を実行する必要があります。なぜなら、これはサーバーの処理の負荷を軽減するためです (ドロップが true の場合、残りの拡張が直ちに戻ると仮定します)。

TCL 拡張

Tcl で拡張機能を記述する場合は、Tcl API、エラーとブール変数の処理方法、および Tcl 拡張機能の初期化方法を理解する必要があります。Cisco Prime Network Registrar は TCL バージョン 8.6 を使用します。

（注） 単一の TCL インタープリタが DHCP サーバーによって使用されます。これはパフォーマンスに重大な影響を与える可能性があります。TCL 拡張機能は、高性能なマルチスレッド DEX 拡張機能や非常にシンプルで高速な操作に変更する前のより複雑なロジックのプロトタイピングに最適です。

関連項目

TCL アプリケーション プログラム インターフェイス

すべての Tcl 拡張は、同じルーチンシグネチャを持っています。

```
proc yourentry { request response environ } { # your-code }
```

辞書のデータ項目を操作するには、これらの引数をコマンドとして扱う必要があります。したがって、入力パケットのgiaddrを取得するには、次の書き込みを行います。

```
set my_giaddr [ $request get giaddr ]
```

これにより、パケット内のgiaddrの文字列値にmy_giaddr Tcl 変数が設定されます。たとえば、10.10.1.5 または 0.0.0.0 などです。

次の Tcl ステートメントを使用して、入力パケットのgiaddrを書き換えることができました。

```
$request put giaddr "1.2.3.4"
```

複数の拡張ポイントに対して1つのルーチンエントリを構成し、サーバーが呼び出す拡張ポイントに応じてその動作を変更するには、DHCPサーバーは、環境ディクショナリの拡張ポイントextension-pointの ASCII 名をキーの下に渡します。

Tcl 拡張機能の例については、Cisco Prime Network Registrar ディレクトリ /opt/nwreg2/local/examples/dhcp/tcl（デフォルト）を参照してください。

TCL エラーの処理

次の場合は、Tcl エラーが発生します。

- 使用できない辞書を参照します。
- 使用できないディクショナリ データ項目を参照します。
- 無効なデータ項目 (たとえば、無効な IP アドレス) に対して put 操作を要求します。

このような場合、ステートメントを catch error ステートメントで囲む場合を除き、拡張は直ちに失敗します。

```
catch { $request put giaddr "1.2.3.a" } error
```

TCL エラーの処理

次の場合は、Tcl エラーが発生します。

- 使用できない辞書を参照します。
- 使用できないディクショナリ データ項目を参照します。
- 無効なデータ項目 (たとえば、無効な IP アドレス) に対して put 操作を要求します。

このような場合、ステートメントを catch error ステートメントで囲む場合を除き、拡張は直ちに失敗します。

```
catch { $request put giaddr "1.2.3.a" } error
```

Tcl 拡張機能の構成

Tcl 拡張を構成するには、それを書き込み、次の拡張ディレクトリに配置します。

/var/nwreg2/local/extensions/dhcp/tcl

DHCP サーバーは、起動時に拡張を構成すると、Tcl ソース ファイルをインタプリタに読み込みます。ソースファイル内の Tcl インタプリタがファイルをロードできない場合、構文エラーは拡張に失敗します。通常、DHCP サーバーは、エラーを見つけるために Tcl からログ ファイルにエラー トレースバックを生成します。

TCL でのブール変数の処理

環境ディクショナリでは、ブール変数は文字列値で、値はtrueORになります。 falseDHCP サーバーは、値をtrueまたはfalseに設定する拡張を要求します。ただし、要求ディクショナリまたは応答ディクショナリでは、ブール値は 1 バイトの数値形式であり、true1falseです0。C/C++ 拡張機能の方が効率的ですが、この方法では Tcl API が少し複雑になります。

TCL での init-entry 拡張ポイント

Tcl拡張はinit-entry拡張ポイントをサポートインイット・エントリー（468ページ）しています (を参照してください) 、 init-argsパラメータでコマンド ラインに渡すarguments引数は、 key に関連付けられた環境辞書に表示されます。

単一のTCLインタープリタがDHCPサーバーによって使用されます。これにより、情報フローの問題が回避され、クライアント要求のフォローに使用できる情報を保存するためのグローバル変数を使用できますが、パフォーマンスに重大な影響を与えます。

すべての TCL 拡張は TCL インタープリタを共有していることに注意してください。Tcl 拡張が、グローバル変数を初期化したり、プロシージャを定義したりする場合は、これらが他の Tcl 拡張グローバル変数またはプロシージャ名と矛盾していないことを確認してください。

C/C++ 拡張

すべての DHCP C/C++dex拡張は、DHCP 拡張の略で拡張です。

関連項目

C/C++ API

C/C++ API entryinit-entryのルーチンと ルーチンの両方のルーチン署名は次のとおりです。

```
typedef int (DEXAPI * DexEntryPointFunction)(
int iExtensionPoint,
dex_AttributeDictionary_t* pRequest,
dex_AttributeDictionary_t* pResponse,
dex_EnvironmentDictionary_t* pEnviron );
```

3 つの構造体へのポインターと共に、拡張ポイントの整数値は、各ルーチンのパラメーターの 1 つです。

C/C++ API は、共有ライブラリを Cisco Prime Network レジストラー DHCP サーバー ファイルとリンクする必要がないように、特に構築されています。拡張機能を構成するときに、ルーチンのエントリを構成します。要求ディクショナリ、応答ディクショナリ、および環境ディクショナリに対して実行する操作に必要なコールバック情報は、拡張ルーチンに渡される 3 つのディクショナリ パラメータを構成する構造体に含まれています。

DHCP サーバーは、すべてのバイナリ情報をネットワーク順に返しますが、実行アーキテクチャに対して正しく配置されるとは限りません。

C/C++ でのタイプの使用

型を使用する多くの C/C++ ルーチンが使用できますgetByType()。これらのルーチンは、パフォーマンスに影響を受けやすい環境で使用するように設計されています。これらのルーチンの背後にある理由は、拡張機能がinit-entry、たとえば、ポイントで、型へのポインターを取得し、その後 C/C++ API のルーチンを呼び出すときに、文字列値の名前の代わりにポインターを使用する可能性があります。この方法で型を使用すると、実行の拡張処理フローから 1 つのハッシュ テーブルルックアップが削除され、拡張機能のパフォーマンスが (少なくともわずかに) 向上する必要があります。

C/C++ 拡張機能のビルド

ディレクトリ /opt/nwreg2/local/examples/dhcp/dex には、サンプルの C/C++ 拡張コードと、サンプル拡張機能を構築するために設計された短いメイクファイルが含まれています。独自の拡張子を作成するには、このファイルを変更する必要があります。このセクションには、Visual C++ および GNU C++ のセクションがあります。コメント行を移動するだけで、ご使用の環境に合わせてファイルを構成できます。

拡張機能はインクルードファイル dex.h を参照する必要があります。このファイルには、プログラムが C/C++ API を使用するために必要な情報が含まれています。

.so ファイル（すべての dex 拡張は共有ライブラリ）を作成したら、/var/nwreg2/local/extensions/dhcp/dex ディレクトリに移動する必要があります。その後、それらを設定できます。

C/C++ でのスレッドセーフな拡張の使用

DHCP サーバーはマルチスレッドなので、その DHCP サーバー用に記述された C/C++ 拡張機能はスレッド セーフである必要があります。複数のスレッド 、および場合によっては複数のプロセッサは、同じエントリ ポイントでこれらの拡張機能を同時に呼び出すことができる必要があります。Cisco Prime Network レジストラー用の C/C++ 拡張機能を設計する前に、マルチスレッド環境用のコードを記述した経験が豊富である必要があります。

注意 C/C++ 拡張機能はすべてスレッド セーフである必要があります。そうしないと、DHCP サーバーは正しく動作せず、診断が非常に困難な方法で失敗します。これらの拡張機能が使用するすべてのライブラリおよびライブラリ ルーチンもスレッド セーフである必要があります。

いくつかのオペレーティング システムでは、使用するランタイム関数がスレッド セーフであることを確認する必要があります。各関数のマニュアルを確認します。いくつかのオペレーティングシステムでは、スレッドセーフな特別なバージョンが提供されています(多くの場合、関数名_r)。

スレッドセーフでない呼び出しを行うスレッドがある場合、そのスレッドは、その呼び出しの安全なバージョンまたはロックされたバージョンを構成するスレッドに影響を与えます。これにより、メモリの破損、サーバー障害などが発生する可能性があります。

これらの問題の原因が明らかになることはめったにないので、これらの問題を診断することは非常に困難です。サーバー障害を引き起こすには、非常に高いサーバー負荷または多数のプロセスを持つマルチプロセッサマシンが必要です。数日間の実行時間が必要な場合があります。多くの場合、拡張実装の問題は、一定期間の重い負荷が続くまで現れなくなることがあります。

ランタイムまたはサードパーティのライブラリによっては、スレッドセーフでない呼び出しを行う可能性があるため、検出できない外部ファイルが (UNIXnm上で) リンクされている場合は、実行可能ファイルを確認してください。

ライブラリーのルーチンが、次の表に示す _r 接尾部を持たないルーチンを呼び出す場合、ライブラリーはスレッド・セーフではなく、使用できません。これらのライブラリ ルーチンのスレッド セーフ バージョンへのインターフェイスは、オペレーティング システムによって異なる場合があります。

asctime_r	getgrid_r	getnetent_r	getrPCbynumber_r	lgamma_r
ctermid_r	getgrnam_r	getprotobyname_r	getrPCent_r	localtime_r
ctime_r	gethostbyaddr_r	getprotobynumber_r	getservbyname_r	nis_sperror_r
fgetgrent_r	gethostbyname_r	getprotoent_r	getservbyport_r	rand_r
fgetpwent_r	gethostent_r	getpwnam_r	getservent_r	readdir_r
fgetspent_r	getlogin_r	getpwent_r	getspent_r	strtok_r
gamma_r	getnetbyaddr_r	getpwuid_r	getspnam_r	tmpnam_r
getgrent_r	getnetbyname_r	getrPCbyname_r	gmtime_r	ttyname_r

C/C++ 拡張の設定

サーバーの実行中は .dll ファイルと .so ファイルがアクティブであるため、上書きすることはお勧めできません。サーバーを停止した後、.dll ファイルと .so ファイルを新しいバージョンで上書きできます。

C/C++ 拡張のデバッグ

C/C++ 共有ライブラリは DHCP サーバーと同じアドレス空間で実行され、DHCP サーバー内の情報へのポインタを受け取るため、C/C++ 拡張機能のバグによって DHCP サーバーのメモリが非常に簡単に破損し、サーバーの障害が発生する可能性があります。このため、C/C++ 拡張機能の作成とテストを行う場合は、細心の注意を払います。多くの場合、Tcl 拡張機能を持つ拡張機能へのアプローチを試して、パフォーマンスを向上させるために C/C++ で拡張機能をコーディングする必要があります。

関連項目

C/C++ における DHCP サーバー メモリへのポインター （445 ページ）
C/C++ での init-entry エントリ ポイント （446 ページ）

C/C++ における DHCP サーバー メモリへのポインター

C/C++ 拡張インターフェイス ルーチンは、次の 2 つの形式で DHCP サーバー メモリにポインターを返します。

- 一連のバイトへの char* ポインタ。

- abytes_tと呼ばれる構造体へのポインタで、関連付けられた長さ (dex.h で定義される) を持つ一連のバイトへのポインタを提供します。

どちらの場合も、DHCP サーバー メモリへのポインタは有効ですが、拡張ポイントで拡張機能が実行されます。また、この要求を処理するシリーズの残りの拡張ポイントにも有効です。したがって、post-packet-decode拡張ポイントで返されるabytes_tポインターは、post-send-packet拡張ポイントで有効です。

ポインターは、環境ディクショナリーに入れられた情報が有効である限り、有効です。ただし、1 つの例外があります。1 つの C/C++ ルーチンであるgetType、 は、型を参照するabytes_tへのポインターを返します。これらのポインターは、拡張機能の有効期間全体を通じて有効です。通常、サーバーは、拡張ポイントでこのルーチンをinit-entry呼び出し、共有ライブラリの静的データの型を定義するabytes_t構造体へのポインターを保存します。返getTypeされるabytes_t構造体へのポインターは、初期化のinit-entry呼び出しから初期化解除の呼び出しまで有効です。

C/C++ での init-entry エントリ ポイント

DHCP サーバーは、init-entry拡張機能を構成するときにインイット・エントリー （468 ページ）1 回 (を参照)、拡張機能を構成解除するときに 1 回、拡張ポイントを呼び出します。dex.h ファイルは、構成および構成解除の DEX_UNINITIALIZE DEX_INITIALIZE呼び出しの拡張ポイントとして渡される 2 つの拡張ポイント値を定義します。拡張ポイント データ項目の環境ディクショナリ値は、initialize各呼uninitializeび出しの中またはまたは呼び出しの中にあります。

拡張ポイントをinit-entryinitialize呼び出すときに、環境ディクショナリ データpersistent項目に値trueが含まれている場合は、呼び出しから戻る前に、いつでもuninitialize環境ディクショナリ ポインタを保存して使用できます。このようにして、バックグラウンドスレッドは、環境ディクショナリ ポインタを使用して、サーバー ログ ファイルにメッセージを記録できます。一度に 1 つのスレッドがディクショナリへの呼び出しを処理するように、ディクショナリへのすべてのアクセスをインターロックする必要があることに注意してください。保存されたディクショナリ ポインタは、uninitialize拡張が呼び出しから戻ったときに使用できます。このようにして、バックグラウンド スレッドは、終了時にメッセージをログに記録できます。

拡張を使用した DHCP 要求処理

Cisco プライムネットワーク レジストラー DHCP サーバーには、独自のエクステンションをアタッチできる拡張ポイントがあります。制御の処理フロー内で、それらを使用する場所を示すわかりやすい名前が付きます。

拡張ポイントは DHCP クライアントからの入力要求の処理に関連しているため、DHCP サーバーが要求をどのように処理するかを理解しておくと役立ちます。要求処理は、次の 3 つの一般的なステージで行われます。

1. 初期要求処理 (表 44 : 拡張機能を使用した初期要求処理を参照)
2. DHCPv4 または DHCPv6 処理 (表 45 : 拡張機能を使用した DHCPv4 または DHCPv6 要求処理を参照)
3. 最終応答処理 (表 46 : 拡張機能を使用した最終応答処理を参照)

表 *44 :* 拡張機能を使用した初期要求処理

クライアント要求処理ステージ	使用する拡張ポイント
1. DHCP クライアントからパケットを受信します。	pre-packet-decode
2. パケットをデコードします。	post-packet-decode
3. クライアント クラスを決定します。	
4. クライアント クラスを変更します。	post-class-lookup
5. クライアント クラスを処理し、クライアントを検索します。	pre-client-lookup post-client-lookup
6. 要求から応答コンテナーを作成します。	

表 *45 :* 拡張機能を使用した *DHCPv4* または *DHCPv6* 要求処理

クライアント要求処理ステージ	使用する拡張ポイント
1. DHCPv4 で、このクライアントに既に関連付けられているリースがあれば、そのクライアントのリースを探すか、または新しいリースを見つけます。	
2. このクライアントに関連付けられているすべての要求をシリアル化します(要求がシリアル化キューの先頭に到達すると処理が続行されます)。	
3. DHCPv6 では、クライアント要求を処理し、必要に応じてリースを生成します。サーバーは、バインディングごとに使用できるプレフィックスごとに、少なくとも 1 つの優先リースをクライアントに提供しようとします。 リースを生成し、クライアント要求に対してリース状態を変更できますが、予約済みのリースに対しては生成できません。	generate-lease(DHCPv6lease-state-changeでは両方で複数の呼び出しが可能)
4. リースがこのクライアントに対して (まだ) 許容できるかどうかを判断します(DHCPv6 では複数回発生する可能性があります)。	check-lease-acceptable
5. 必要に応じて DNS 更新操作を開始します (DHCPv6 で複数回発生する可能性があります)。	

表 *46 :* 拡張機能を使用した最終応答処理

クライアント応答処理ステージ	使用する拡張ポイント
1. 応答パケットに含めるすべてのデータを収集します。	
2. リース データベースに書き込みます。	
3. エンコード用の応答パケットを準備します。	pre-packet-encode
4. クライアントに送信する応答パケットをエンコードします。	post-packet-encode
5. パケットをクライアントに送信します。	post-send-packet
6. クライアントと要求のすべてのコンテキストを解放します。	environment-destructor

これらの手順と拡張機能を使用するその他の機会については、次のセクションで説明します。拡張ポイントは に示boldされています。

関連項目

DHCPv6 拡張の有効化 （449 ページ）
パケットの受信 （449 ページ）
パケットのデコード （449 ページ）
クライアントクラスの決定 （449 ページ）
クライアントクラスの変更 （450 ページ）
クライアントクラスの処理 （450 ページ）
応答コンテナの作成 （451 ページ）
ネットワークとリンクの決定 （451 ページ）
リースの検索 （451 ページ）
リース要求のシリアル化 （453 ページ）
リースの受け入れの決定 （453 ページ）
DHCPv6 リース （454 ページ）
応答パケット データの収集 （456 ページ）
応答パケットの符号化 （457 ページ）
安定ストレージの更新 （457 ページ）
パケットの送信 （457 ページ）
DNS 応答の処理 （457 ページ）
リース状態変更のトレース （458 ページ）

有効なリースクエリ通知の制御 （458 ページ）

DHCPv6 拡張の有効化

デフォルトでは、拡張はDHCPv4 のみをサポートすると想定されます。DHCPv6 拡張を記述するには、次のinit-entry必要がある拡張ポイントを実装する必要があります。

1. dhcpサポート環境データ項目v4を (DHCPv4 の場合のみ、プリセット値)、(DHCPv6v6の場合v4,v6のみ)、または (DHCPv4 および DHCPv6 の場合) に設定します。このデータ項目は、拡張機能がサポートする内容をサーバーに示します。
2. 拡張拡張APIバージョン環境データ項目を2に設定します。(拡張拡張 api バージョンが2に設定されていない場合、dhcpサポート データ項目は無視されます。

パケット形式、DHCP プロトコル、および内部サーバー データの違いにより、DHCPv4 と DHCPv6 用の拡張を個別に作成する必要がある場合があります。ただし、両方の種類の拡張機能の基本は非常に同じです。

サーバーは、処理時に基本的に同じ場所でこれらの拡張ポイントを呼び出しますが、クライアントごとに複数のリース要求が発生する可能性があるため、一部の DHCPv6 拡張ポイントを複数回呼び出すことができます。

パケットの受信

DHCP サーバーは、ポート 67 の DHCPv4 パケットをポート 547 (DHCP 入力ポート) で受信し、それらを処理用にキューに入れます。UDP 入力キューをできるだけ早く空にしようと試み、空きスレッドが処理可能になるとすぐに、受信したすべての要求を内部リストに保持し、処理を行います。このキューの長さは設定でき、設定された最大長を超えてはなりません。

パケットのデコード

フリースレッドが使用可能な場合、DHCP サーバーは、入力要求を処理するタスクを割り当てます。最初に行われる操作は、入力パケットをデコードして、それが有効な DHCP クライアント パケットかどうかを判断することです。このデコード プロセスの一部として、DHCP サーバーは、すべてのオプションが有効かどうかを確認します。つまり、オプションの長さが要求パケットの全体的なコンテキストで意味を持つかどうかを確認します。また、DHCP 要求パケット内のすべてのデータもチェックしますが、この段階ではパケット内のデータに対しては何も処理を行いません。

入力パケットpre-packet-decodeを書き換える場合は、拡張ポイントを使用します。DHCP サーバーがこの拡張ポイントを通過した後、パケットからのすべての情報を複数の内部データ構造体に格納し、後続の処理をより効率的にします。

クライアントクラスの決定

クライアント クラスルックアップ IDで式を構成すると、この段階で DHCP サーバーが式を評価します(式の説明式の使用方法 （389ページ）については、「」を参照)。式の結果は、<null>

または文字列に変換された値です。文字列の値は、クライアント クラス名または <none> のいずれかである必要があります。<none> の場合、サーバーは、クライアント クラスルックアップIDが構成されていない場合と同じ方法でパケットの処理を続行します。<null>応答の場合、またはクライアント クラスルックアップ IDを評価するエラーの場合、サーバーはエラー メッセージをログに記録し、パケットを破棄します (post-class-lookup拡張ポイントで構成された拡張がパケットをドロップしないようにサーバーに指示しない限り)。クライアントクラスを設定するプロセスの一部として、DHCPサーバーはそのクライアントクラスに対して設定された制限 IDを評価し、要求と共に保存します。

クライアントクラスの変更

DHCP サーバーは、クライアントクラスルックアップ IDを評価し、クライアント クラスを設定した後、拡張ポイントにアタッチpost-class-lookupされた任意の拡張を呼び出します。この拡張機能を使用して、クライアント クラスが要求に関連付けられるデータ (制限 idなど) を変更できます。また、クライアント クラスルックアップ IDの評価によってパケットがドロップされた場合も、拡張機能は学習します。この拡張は、パケットをドロップする必要があるかどうかを調べますが、サーバーにパケットをドロップしないようにサーバーに指示します。

また、post-class-lookup拡張ポイントで実行されている拡張機能は、要求に対して新しいクライアント クラスを設定し、現在のクライアント クラスではなくそのクライアント クラスのデータを使用できます。これは、クライアントクラスを設定する唯一の拡張ポイントで、実際にそのクライアントクラスを要求に使用します。

クライアントクラスの処理

クライアント クラス処理を有効にした場合、DHCP サーバーはこの段階で処理を実行します。

拡張ポイントpre-client-lookupを使用して、検索を妨げたり、既存のデータを上書きするデータを提供したりして、クライアントが参照するように影響を与えます。DHCPサーバーは、拡張ポイントpre-client-lookupを通過した後、クライアントをローカル データベースまたは LDAP データベース (構成されている場合) で検索します (拡張機能が特に禁止しない限り)。

サーバーは、クライアントを参照した後、クライアントエントリのデータを使用して、追加の内部データ構造を入力します。DHCP サーバーは、指定されたクライアント クラス エントリのデータを使用して、クライアントエントリで指定されていないデータを完成させます。DHCP サーバーは、追加の処理のために内部データ構造のさまざまな場所に格納されているすべてのデータを取得すると、次の拡張ポイントを実行します。

拡張ポイントpost-client-lookupを使用して、クライアント クラスの処理から入力された内部サーバー データ構造を調べるなど、クライアント クラスの参照プロセスの操作を確認します。また、拡張ポイントを使用して、DHCPサーバーが追加の処理を行う前にデータを変更することもできます。

応答コンテナの作成

この段階では、DHCPサーバーは要求の種類を決定し、入力に基づいて適切な応答コンテナーを構築します。たとえば、要求が DHCPDISCOVER である場合、サーバーは DHCPOFFER 応答を作成して処理を実行します。入力要求が BOOTP 要求の場合、サーバーは BOOTP 応答を作成して応答処理を実行します。

DHCPv6 の場合、サーバーは要求に応じて、アドバタイズ パケットまたは REPLY パケットを作成します。

ネットワークとリンクの決定

DHCPサーバーは、すべての要求の発信元のサブネットを特定し、IPアドレスを含む一連のアドレスプール、スコープ、プレフィックス、またはリンクにマッピングする必要があります。

DHCPv4 の場合、DHCP サーバー内部はネットワークの概念であり、この場合は LAN セグメントまたは物理ネットワークを指します。DHCPサーバーでは、すべてのスコープまたはプレフィックスが 1 つのネットワークに属します。

スコープまたはプレフィックスの中には、ネットワーク番号とサブネットマスクが同じであるため、同じネットワーク上でグループ化されているものもあります。その他のグループは、主スコープまたはプレフィックス ポインターを通じて関連付けられているため、グループ化されます。

Cisco Prime Network レジストラー DHCP サーバーは、次の順序で DHCP クライアント要求を処理するために使用するネットワークを決定します。

1. ソース・アドレスを判別する場合は、giaddrか、giaddrがゼロの場合は、要求が到着したインターフェースのアドレスを判別します。
2. このアドレスを使用して、このアドレスと同じサブネット上にあるサーバーで構成されたスコープまたはプレフィックスを検索します。サーバーがスコープまたはプレフィックスを見つけられない場合は、要求を削除します。
3. スコープまたはプレフィックスを見つけた後、そのネットワークを使用して以降の処理を行います。

DHCPv6処理については、リンクとプレフィックスの決定（156ページ）を参照してください。

リースの検索

DHCPv4の場合、DHCPサーバーがネットワークを確立すると、ネットワーク レベルで保持されているハッシュ テーブルが検索され、ネットワークが既にクライアント IDを認識しているかどうかを確認できます。このコンテキストでは、このクライアントが以前にこのネットワークでオファーまたはリースを受け取り、その時点以降、別のクライアントにリースが提供されなかったりリースされたりしていないことを意味します。したがって、現在のリースまたは使用可能な期限切れのリースがネットワーク レベルのハッシュテーブルに表示されます。DHCPサーバーは、リースを検出した場合、次の手順に進みます。

DHCP サーバーがリースを検出せず、これが BOOTP または DHCPDISCOVER 要求である場合、サーバーはネットワーク内のスコープまたはプレフィックスから予約済みリースを検索します。

予約済みリースが見つかった場合、サーバーはスコープまたはプレフィックスとリースの両方が受け入れられるかどうかを確認します。予約済みリースと、それを含むスコープまたはプレフィックスに関して、以下の条件を満たす必要があります。

- リースは使用可能である必要があります (別の DHCP クライアントにはリースされません)。
- スコープまたはプレフィックスは、要求の種類 (BOOTP または DHCP) をサポートする必要があります。
- スコープまたはプレフィックスは、非アクティブ化された状態であってはなりません。
- リースは非アクティブ化された状態であってはなりません。
- 選択タグには、クライアント選択基準をすべて含める必要があり、クライアント選択基準から除外されるものは含まれていなければなりません。
- スコープまたはプレフィックスは、更新専用の状態にすることはできません。

予約済みのリースが許容される場合、サーバーは次の手順に進みます。このクライアントの既存のリースまたは予約済みリースが見つからなかった場合、サーバーはこのクライアントに使用可能な IP アドレスを見つけようとします。

DHCPサーバーが使用する一般的なプロセスは、このネットワークに関連付けられたすべてのスコープまたはプレフィックスをラウンドロビン順にスキャンし、クライアントに対して許容可能なスコープと使用可能なアドレスを探します。有効なスコープまたはプレフィックスには、次の特性があります。

- クライアントに選択基準が関連付けられている場合、選択タグにはクライアント包含基準がすべて含まれている必要があります。
- クライアントに選択基準の除外が関連付けられている場合、選択タグにはクライアント除外基準が含まれていなければなりません。
- スコープまたはプレフィックスがクライアント要求タイプをサポートする必要がある- クライアント要求が DHCPREQUEST である場合は、DHCP のスコープまたはプレフィックスを有効にする必要があります。同様に、要求が BOOTP 要求である場合は、BOOTP と動的 BOOTP のスコープまたはプレフィックスを有効にする必要があります。
- 更新のみの状態にすることはできません。
- 非アクティブ状態にすることはできません。
- 使用可能なアドレスが必要です。

サーバーが許容範囲またはプレフィックスを見つけられない場合、メッセージをログに記録してパケットを廃棄します。

DHCPv6処理については、リンクとプレフィックスの決定（156ページ）を参照してください。

リース要求のシリアル化

1 つのクライアントとリースに対して複数の DHCP 要求を同時に処理できるため、DHCPv4 要求をリース レベルでシリアル化する必要があります。サーバーは、リースでキューに登録し、キューイングの順序で処理します。

DHCPv6 の場合、サーバーはクライアント (リンク単位) でシリアル化され、リースではシリアル化されません。

リースの受け入れの決定

DHCPv4 の場合、DHCP サーバーは、クライアントに対してリースが (まだ) 受け入れられるかどうかを判断します。初回クライアントの新規取得リースの場合は、許容されます。ただし、サーバーが既存のリースの更新を処理する場合、サーバーがリースを許可してから受け入れ可能な条件が変更されている可能性があるため、その受け入れ可能性を再度確認する必要があります。

クライアントの現在のリースとは異なる予約がある場合、サーバーは最初に予約済みリースが許容できるかどうかを判断します。リリースの受け入れ基準は次のとおりです。

- 予約済みリースが使用可能である必要があります。
- 予約済みリースは非アクティブ状態にしないでください。
- スコープまたはプレフィックスは非アクティブ状態にしないでください。
- 要求が BOOTP の場合、スコープまたはプレフィックスは BOOTP をサポートする必要があります。
- 要求が DHCP の場合、スコープまたはプレフィックスが DHCP をサポートしている必要があります。
- クライアントに選択基準がある場合、選択タグにはクライアントの包含条件がすべて含まれている必要があります。
- クライアントに選択基準の除外がある場合、選択タグにはクライアントの除外基準が含まれていなければなりません。
- このリースに以前関連付けられているクライアントが現在のクライアントではない場合、スコープまたはプレフィックスは更新専用の状態であってはなりません。

予約済みリースがこれらの基準をすべて満たしている場合、DHCP サーバーは現在のリースを受け入れられないと見なします。このクライアントに予約されたリースがない場合、または予約済みリースが受け入れ可能な条件を満たしていない場合、DHCP サーバーは現在のリースを受け入れ可能な状態で調べます。

受け入れ可能な基準は次のとおりです。

- リースは非アクティブ状態にしないでください。
- スコープまたはプレフィックスは非アクティブ状態にしないでください。
- 要求が BOOTP の場合、スコープまたはプレフィックスは BOOTP をサポートする必要があります。要求が DHCP の場合、スコープまたはプレフィックスが DHCP をサポートしている必要があります。

- クライアントがこのリースの予約を持っておらず、要求が BOOTP である場合、スコープまたはプレフィックスは動的 BOOTP をサポートする必要があります。
- クライアントがこのリースの予約を持っていない場合、他のクライアントもできません。
- クライアントに選択基準がある場合、選択タグにはクライアントの包含条件がすべて含まれている必要があります。
- クライアントに選択基準の除外がある場合、選択タグにはクライアントの除外基準が含まれていなければなりません。
- このリースに以前関連付けられているクライアントが現在のクライアントではない場合、スコープまたはプレフィックスは更新専用の状態であってはなりません。

ヒント DHCP サーバーの処理のこの時点で、拡張ポイントをcheck-lease-acceptable使用できます。これを使用して、受け入れ性テストの結果を変更できます。これは細心の注意を払って行うだけです。

リースが受け入れられないと判断した場合、DHCP サーバーは、現在処理されている特定の DHCP 要求に応じて、異なるアクションを実行します。

- DHCPDISCOVERDHCP サーバーは現在のリースを解放し、このクライアントに対して別の許容可能なリースを取得しようとします。
- DHCPREQUEST:リースが無効であるため、DHCP サーバーはDHCP クライアントにNACKを送信SELECTINGします。クライアントは、すぐにディスカバー要求を発行して新しい DHCPOFFER を取得する必要があります。
- DHCPRENEW()DHCPDHCPREBINDサーバーは、DHCP クライアントを強制的に INIT フェーズに入れようとする NACK を DHCP クライアントに送信します(DHCP クライアントが DHCPDISCOVER 要求を強制的に発行するように試みます)。クライアントが実際に要求を発行するまで、リースは有効です。
- BOOTP：DHCP サーバーは、現在のリースを解放し、このクライアントに対して受け入れ可能な別のリースを取得しようとします。

注意 延長点には細心check-lease-acceptableの注意を払ってください。拡張ポイントが返す答えが、DHCPDISCOVER 要求または動的 BOOTP 要求で実行された使用可能なリースの検索での受け入れ可能なチェックと一致しない場合、無限のサーバー ループが発生する可能性があります(即時または次の DHCPDISCOVER または BOOTP のいずれか要求)。この場合、サーバーは新しく使用可能なリースを取得し、それが受け入れられないと判断し、新しく使用可能なリースを取得し、連続ループで許容できないリースを判断します。

DHCPv6 リース

DHCP サーバーは、クライアントのIA_NA、IA_TA、およびIA_PDオプションをスキャンして、IPv6 リース要求をDHCPv6 バインディング （233 ページ）処理します (を参照)。これらのオプションごとに、サーバーはクライアントが明示的に要求するリースを考慮します。クライアントとバインディング (IA オプションおよび IAID) にリースがすでに存在する場合、サーバー

は、リースがまだ受け入れられるかどうかを判別します。クライアントがクライアントに対してまだ存在しないリースの場合、サーバーは次の場合にクライアントにそのリースを与えようとします。

- 別のクライアントまたはバインディングがリースをまだ使用していません。
- リースのプレフィックスには、割り当てアルゴリズム属性にクライアント要求フラグが設定されています。
- リースは使用でき、使用できるプレフィックス (を参照してくださいDHCPv6 プレフィックスのユーザービリティ （455 ページ）)。

次に、サーバーは、クライアントが予約を使用していること、およびクライアントが、リンク上の各使用できるプレフィックスに対して、優先する有効期間がゼロ以外の、有効なリースを持っていることを確認しようとします。したがって、サーバーはこれらの各バインディングを次のように処理します。

1. プレフィックス割り当てアルゴリズム属性で予約フラグが設定されている場合は、バインディングにクライアント予約(まだ使用されていない)を追加します。サーバーは、予約に対して適切なタイプの最初のバインディングを使用します。つまり、IA_NAバインディングにアドレス リースを使用し、IA_PD バインドのプレフィックス リースを使用します。
2. クライアントが使用できるプレフィックスごとに優先されるゼロ以外の有効期間を持つリースがない場合、サーバーはクライアントにリースを割り当てようとします。プレフィックス割り当てアルゴリズムフラグは、サーバーがリースを割り当てる方法を制御します。

関連項目

DHCPv6 プレフィックスのユーザービリティ （455 ページ）

DHCPv6 リースのユーザービリティ （455 ページ）

DHCPv6 リースの割り当て （456 ページ）

DHCPv6 プレフィックスのユーザービリティ

使用できるプレフィックス:

- 非アクティブ化されません。
- 期限切れではありません。
- バインディング・タイプのリースを許可します。
- クライアントの選択基準 (存在する場合) に一致します。
- クライアント選択除外条件 (存在する場合) に一致しません。

DHCPv6 リースのユーザービリティ

使用できるリースは次のとおりです。

- 使用不可でないこと。
- 失効していない。

- 非アクティブ化されていません。
- 別のクライアント用に予約されていません。
- すべての更新を阻害したり、再起動時に更新を禁止したりしない.
- 更新された場合は更新可能 (IA_TAリースは更新可能ではありません)。
- 有効な有効期間がゼロ以外の場合は、リーアスブルです。

DHCPv6 リースの割り当て

サーバーは、プレフィックスに新しいリースを割り当てる必要がある場合、プレフィックスgenerate-lease拡張フラグがアロケーション アルゴリズム属性に設定されている場合、拡張ポイントで登録されている拡張を呼び出します。（リースの生成 （477ページ） を参照）。拡張機能は、割り当てるアドレス(IA_NAまたはIA_TAバインディング) またはプリフィックス(IA_PDバインディング) を指定するか、サーバーが通常の割り当てアルゴリズムを使用するように要求するか(割り当てアルゴリズムで有効になっている場合)、またはこのプレフィックスのリースの割り当てをスキップするようサーバーに要求します。サーバーが無効なアドレスまたはプレフィックスを指定した場合、または既に使用中の場合、サーバーは拡張を再度呼び出す可能性があります。

拡張が許可されていない場合、拡張機能が登録されていないか、拡張機能がサーバーの通常の割り当てアルゴリズムを要求する場合、サーバーはランダムに生成されたアドレスを割り当てるか、(プレフィックス割り振りアルゴリズム属性によって制御される)最初の最適な使用可能なプレフィックスを見つけてリースを作成します。

サーバーがリースを取得し、そのサーバーで受け入れ可能なDHCPv6 リースのユーザービリティ （455 ページ）チェックを行うと (を参照)、サーバーはcheck-lease-acceptableエクステンションポイントで登録されているエクステンションを呼び出して、エクステンションがリースの受け入れ可能を変更できるようにします。（check-lease-acceptable （480ページ） を参照）。通常、この拡張ポイントを使用して、許容できる結果を許容できない結果に変更します。ただし、サーバーでは許容できない結果を許容可能な結果に変更できますが、悪影響を及ぼす可能性があるため、この方法は推奨されません。リースが受け入れられない場合、サーバーは別のリースを割り当てようとする可能性があります。したがって、無限ループを避けるために注意してください。場合によっては、クライアントが取得check-lease-acceptablegenerate-leasegenerate-leaseするリースのフル コントロールに対して 、 および 拡張ポイントが必要になる場合があります。

サーバーは、各check-lease-acceptableリースの各クライアント要求の拡張ポイントを呼び出します。

応答パケット データの収集

この処理の段階では、DHCP サーバーはDHCP応答で返送するすべてのデータを収集し、応答を送信するアドレスとポートを決定します。拡張ポイントをpre-packet-encode使用して、応答でDHCP クライアントに返送されるデータを変更したり、DHCP応答を送信するアドレスを変更したりできます。（pre-packet-encode （481 ページ） を参照）。

⚠

注意 拡張ポイントでドロップされたパケットは、DHCP パケットでも BOOTP パケットでも、残りのリース時間の間は Cisco Prime Network レジストラーのリース状態データベースにリースされるアドレスを示します。 pre-packet-encode このため、パケットを早い時点でドロップすることをお勧めします。

応答パケットの符号化

この段階では、DHCP は応答データ構造内の情報をネットワーク パケットにエンコードします。この DHCP クライアントが DNS アクティビティを必要とする場合、DHCP サーバーは DHCP サーバーの DNS 処理サブシステムに対して DNS 作業要求をキューに入れます。この要求は、可能な限り実行されますが、通常はクライアントにパケットを送信する前には実行されません。（pre-packet-encode （481 ページ） を参照。）

安定ストレージの更新

この段階で、DHCP サーバーは、続行する前に、情報のディスク上のコピーが IP アドレスに関して最新の状態であることを確認します。DHCPv6 の場合、これには複数のリースが含まれる場合があります。

パケットの送信

DHCPpost-send-packet要求/応答サイクルポスト送信パケット （483 ページ）の重大な時間制約の外部で実行する処理については、拡張ポイント (を参照) を使用します。サーバーがパケットをクライアントに送信すると、この拡張ポイントがコールされます。

DNS 応答の処理

ここでは、DHCP サーバーが DNS に名前を追加する処理を簡単に示します。

1. :DHCP サーバーは、転送(A レコード)DNS 要求で使用する名前を作成します。 Builds up a name to use for the A record DHCPv6 の場合、これらは AAAA レコードです。DNS 名は、DHCP 要求のオプションから通常取り込まれるクライアント要求ホスト名およびクライアント ドメイン名データ項目、および DNS 更新設定 (ホスト名生成/v6 ホスト名生成式を含む) など、さまざまなソースから取得されます。
2. - この段階では、DNS 名更新要求の前提条件は、名前が存在しないことを示します。 Tries to add the name, asserting that none exists yet 成功した場合、DHCP サーバーは逆レコードの更新を続行します。
3. サーバーはホスト名を追加しようとし、ホストが存在し、送信されたレコードと同じ TXT レコード(DHCPv6 の DHCID レコード)を持っていることを主張します。 Tries to add the name, asserting that the server should supply it
   - これが成功した場合、サーバーは次の手順に進みます。

- 失敗した場合、サーバーは名前付け再試行が終了したかどうかをチェックします。
- 名前付けエントリが使い果たされなかった場合は、最初のステップに戻り、A レコードの名前を作成します。

DHCPv6 の場合、サーバーは TXT レコードの代わりに DHCID レコードを使用します。また、DHCPv6 クライアントは複数のリースを持つことができますが、転送ゾーンは同じか、または異なる可能性があります。

4. :DHCP サーバーは、リバース (PTR) レコードに関連付ける名前を認識したので、レコードの所有者であると見なすことができるため、前提条件なしでリバース レコードを更新できます。 Updates the reverse record 更新に失敗した場合、DHCP サーバーはエラーを記録します。

リース状態変更のトレース

サーバーは、リースlease-state-changeが状態を変更するたびに (および、状態が変更された場合にのみ) 拡張ポイントを呼び出します。既存の状態は、応答ディクショナリlease-stateデータ項目にあります。新しい状態は、 の環境ディクショナリnew-stateにあります。これはnew-state、既存の状態と等しくない(存在する場合、サーバーは拡張機能を呼び出しません)。サーバーはさまざまな場所で呼び出すため、この拡張機能は読み取り専用であり、ディクショナリ項目を変更しないようにする必要があります。この拡張ポイントは、リース状態の変更を追跡する場合にのみ使用します。

有効なリースクエリ通知の制御

サーバーは、dhcp リスナーのleasequery-send-all属性に基づいて、アクティブなリースクエリ通知用にリースがキューに入っているかどうかを判断します。この属性が有効な場合、DHCP サーバーは常にアクティブな leasequery クライアントに通知を送信します。無効にするか、または設定解除した場合、DHCP サーバーは、アクティブな leasequery クライアントで正確な状態を維持するために必要な通知のみを送信します。

顧客が書き込んだ拡張を使用してリースの送信を制御できるように (特定の状態の変更に関する場合など)、新しいデータ項目であるアクティブリースクエリコントロールが要求ディクショナリと応答ディクショナリの両方に追加されました。これらのデータ項目には、次の 3 つの値があります。

- 0 - 未指定(サーバーが通知を送信するかどうかを決定します)
- 1- 送信 (サーバーが通知を送信します)
- 2 -送信しない(サーバーは通知を送信しません)

アクティブリースクエリコントロールデータ項目は 0 として初期化され、未指定です。

（注） これらのデータ項目は書き込みおよび読み取りできますが、読み取られる値は、以前に書き込まれた値のみです。

これらのデータ項目は、書き込み後にDHCPサーバーに特定のアクションを強制実行させることができますが、前に書き込みせずに読み取ると、常に 0 (未指定) が返されます。これらのデータ項目では、処理中のリースに対する変更(存在する場合)に関するメッセージをアクティブな leasequery クライアントに送信するかどうかを決定する際に DHCP サーバーが行う選択を決定することはできません。したがって、これらのデータ項目は技術的には読み取り/書き込み可能ですが、読み取りでは以前に書き込んだ内容を判断することしかできません。

これらのデータ項目は、リースがアクティブなリースクエリ通知のためにキューに入れられたときと同様に、内部リース状態データベースにリースが書き込まれるときに検査されます (応答ディクショナリが最初に調べられますが、次に要求が返されます)。これは、チェック-リース許容およびリース状態変更拡張ポイントの後、パケットエンコード前の拡張ポイントより前に発生します。したがって、これらの属性に対してパケットエンコード前の拡張ポイントまたはそれ以降に行われた変更は無視されます。

リースがアクティブなリースクエリ通知のキューに入っているかどうかは、次のように決定されます。

応答のアクティブリースクエリ制御	要求のアクティブリースクエリコントロール	リースクエリ送信-すべて	操作
0：指定なし	0：指定なし	偽または未設定	条件付き (リースクエリ送信-すべて属性の説明を参照)
0：指定なし	0：指定なし	[はい（True）]	Sent
0：指定なし	1：送信する	Ignored	Sent
0：指定なし	2 - 送信しない	Ignored	送信されない
1：送信する	Ignored	Ignored	Sent
2 - 送信しない	Ignored	Ignored	送信されない

（注） 応答と要求のアクティブ・リース照会制御は、リース照会-send-all属性の検査の前に検査されます。

これらのディクショナリ データ項目のいずれかが未指定以外の値を持つ場合、その値は dhcp リスナのleasequery-send-all属性で設定されている値をオーバーライドします。

（注） アクティブな leasequery 情報の送信を制御するには、リース状態変更拡張ポイントでのみ実行される単一の拡張を書き込むことはできません。

リース状態の変更は、予期した場合には発生しない場合があります。たとえば、リースがリースされている場合、同じクライアントがディスカバー/オファー/リクエスト/ACKサイクルを通

過すると、リース状態変更拡張ポイントは呼び出されません。したがって、アクティブな leasequery クライアントへの情報の転送を絶対的に制御するには、要求処理でアクティブ・リースクエリ制御属性を初期化し、場合によって、それを変更するか、または、リース状態変更拡張点で応答ディクショナリ値で操作することによってオーバーライドする必要があります。

拡張ディクショナリ

すべての拡張は、3 つの引数を持つルーチンです。これらの引数は、要求ディクショナリ、応答ディクショナリ、および環境ディクショナリを表します。すべての辞書がすべての拡張に使用できるわけではありません。次の表は、拡張機能ポイントと、それらのポイントで使用できるディクショナリを示しています。

表 *47*: 拡張ポイントと関連する辞書

拡張ポイント	ディクショナリ
init-entry	環境
pre-packet-decode	要求、環境
post-packet-decode	要求、環境
pre-client-lookup	要求、環境
post-client-lookup	要求、環境
post-class-lookup	要求、環境
generate-lease	要求、応答、環境
lease-state-change	対応,環境
check-lease-acceptable	要求、応答、環境
pre-packet-encode	要求、応答、環境
post-packet-encode	要求、応答、環境
post-send-packet	要求、応答、環境
environment-destructor	環境

（注） サーバーが DHCPv6 再設定メッセージを送信すると、要求pre-packet-encodeなしで post-packet-encode、post-send-packetおよび 拡張ポイントを呼び出すことができます。

要求ディクショナリと応答ディクショナリの場合、このメソッドをisValid使用して、辞書が拡張ポイントで使用できるかどうかを調べることができます。

3 つのディクショナリはそれぞれ、名前と値のペアで構成されています。環境ディクショナリは、すべての拡張ポイントで使用でき、最も単純なディクショナリです。要求ディクショナリと応答ディクショナリは複雑で、データが入力されます。したがって、これらのディクショナリの 1 つに値を設定する場合は、データ型を値に一致させる必要があります。値の取得、書き込み、および削除にはディクショナリを使用できます。

関連項目

環境ディクショナリ （461 ページ）

要求ディクショナリと応答ディクショナリ （465 ページ）

環境ディクショナリ

環境ディクショナリは、すべての拡張ポイントで使用できます。厳密には、名前と値の両方が文字列である名前と値のペアのセットです。

DHCPサーバーは、環境ディクショナリを使用して、拡張機能のさまざまな点で異なる方法で拡張機能と通信します。一部の拡張ポイントでは、サーバーは、変更する拡張機能の情報を環境ディクショナリに配置します。その他の場合、拡張機能は、拡張機能の処理が完了した後、フローまたはデータを制御する環境ディクショナリ内の値を配置できます。

環境ディクショナリは、拡張機能が名前と値のペアを入れることができるという特徴で一意です。文書化されていない名前と値のペアを使用してもエラーは発生しませんが、サーバーはこれらを認識しません。これらの名前と値のペアは、拡張機能ポイント間でデータを通信する場合に役立ちます。

DHCP サーバーは、DHCP 要求が到着し、処理を通じてその要求にディクショナリが残ると、環境ディクショナリを作成します。したがって、post-packet-decode拡張ポイントで実行される拡張機能は、環境ディクショナリにデータを格納し、pre-packet-encode拡張ポイントで実行される拡張機能は、ディクショナリからそのデータを読み取ることができます。

（注） init-entry拡張ポイントには、固有の環境ディクショナリがあります。

関連項目

一般的な環境ディクショナリ データ項目 （461 ページ）

初期環境ディクショナリ （464 ページ）

一般的な環境ディクショナリ データ項目

次の表のデータ項目は、すべての拡張ポイントで環境ディクショナリで有効です。（各辞書のデータ項目に固有の環境辞書の各セクションを参照してください。）

データ項目は、入力、出力、またはその両方です。

- 入力：DHCP サーバーは値を設定し、それを拡張に入力します。
- 出力：値は DHCP サーバーに出力され、DHCP サーバーは DHCP サーバーに出力され、DHCP サーバーに対して動作します。1 つの拡張ポイントで複数の拡張機能が存在する可能性があるため、拡張ポイントで実行されている以前の拡張機能がこれを設定している可能性があるため、これは、その拡張ポイントで実行される後の拡張機能への「入力」になる可能性があります。テーブルが「入力」ではないことを示している場合、DHCPサーバーがその拡張ポイントで拡張を呼び出す前に明示的にこれを設定しなかったことを意味します。

表 *48*：一般的な環境ディクショナリ データ項目

環境データ項目	説明
drop（input[1]/出力）	拡張機能が終了したときに、ドロップ値が文字列trueと等しい場合、DHCPサーバーは入力パケットをドロップし、ログファイルにメッセージを記録します。最初はfalseに設定します。ほとんどの拡張ポイントで使用できますが、すべてではありません (generate-leaseなど)。 （注） 拡張機能ポイントごとに複数の拡張機能のdropを使用する方法の複数の拡張機能に関する考慮事項（439ページ）推奨事項については、「」を参照してください。
拡張子名(入力)	拡張機能が構成された名前。同じコードを複数の異なる拡張機能として、また複数の異なる拡張ポイントで構成できます。 これにより、構成方法に応じて、1 つのコードで異なる処理を実行できます。コードでは、この文字列を使用して、自身の名前を知る必要がある拡張名シーケンス文字列内で自分自身を見つけることもできます。
拡張子名シーケンス(入力)	この拡張ポイントに対して構成されている拡張機能を表すコンマ区切りの文字列を提供します。拡張機能は、その前と後に実行できる拡張機能を決定することができます。拡張名データ項目は、現在実行中の拡張機能を提供します。 たとえば、最初の拡張子としてtclfirst構成しdexscript、5 番目として構成した場合、拡張子の名前のシーケンスには"tclfirst,,,,dexscript"が含まれます。
拡張ポイント(入力)	拡張ポイントの名前。たとえば、post-packet-decode のようになります。
拡張シーケンス(入力)	拡張ポイントの拡張のシーケンス番号を表す文字列。

環境データ項目	説明
ジャダーオーバーライド(出力)	このデータpre-packet-decode項目は、 、 post-packet-decode、pre-client-lookup、post-client-lookupおよびpost-class-lookup拡張によって設定され、クライアントのネットワークの場所を決定する際に使用するIPv4 アドレスまたはスコープ名を指定できます (giaddr または受信インターフェースのアドレスの代わりに)。これはDHCPv4 要求に対してのみ使用されます(DHCPv6 では無視されます)。スコープ名を指定した場合、クライアントの場所を特定するためだけに使用され、クライアントがそのスコープからリースを取得する必要はありません。
リンク・アドレス・オーバーライド(出力)	このデータpre-packet-decode項目は、 、 post-packet-decode、pre-client-lookup、post-client-lookupおよびpost-class-lookup拡張子によって設定され、クライアントのネットワークロケーションを決定する際に使用するIPv6アドレスまたはプレフィックス名を指定できます(Relay-Forwのリンクアドレスまたは受信インターフェースのアドレスの代わりに)。これは DHCPv6 要求に対してのみ使用されます (DHCPv4 では無視されます)。プレフィックス名を指定した場合、クライアントの場所を決定するためだけに使用され、クライアントがそのプレフィックスからリースを取得する必要はありません。リンクとプレフィックスの決定 （156ページ）を参照してください。
ログドロップメッセージ(出力)	ドロップtrue,値が文字列と等しく、log-drop-message値が拡張が終了したときに文字列falseと等しい場合、DHCP サーバーは入力パケットをドロップしますが、ログ ファイルにメッセージを記録しません。 には適用されませんinit-entry。
IP によるリリース(出力)	これを有効にするには、**パケットデコード前、パケットデコード後、クライアント参照前、ポストクライアントルックアップ**、クラス**後参照**拡張ポイントで呼び出される拡張によって設定する必要があります。 これは、DHCPRELEASE 要求にのみ適用されます。にtrue設定すると、DHCPRELEASE 要求から派生したクライアント IDによってリースを取得できない場合、IP アドレスによってリースを解放するようにサーバーに指示します。
トレース・レベル(出力)	この番号を設定すると、この要求を処理するすべての拡張機能の拡張トレース・レベル・サーバー属性の現行設定が、その番号になります。

環境データ項目	説明
ユーザー定義データ(出力)	要求処理の前にリースと共に保存されたリースのユーザー定義データ属性を使用して設定します。このファイルは、のpre-packet-encode前にディスクに書き込むことができますが、使用することはできません。 null に設定すると、サーバーはリースからのユーザー定義データを無視します。NULL 文字列値を使用して以前の値を削除することはできません。応答にのみ適しています。 サーバーがユーザー定義データをリースに書き込む場合、読み取り専用クライアントユーザー定義データ応答ディクショナリー項目はその値を想定します。 （注） このデータ項目を拡張ポイントの複数の拡張機能で使用する場合は注意が必要です。複数の拡張機能に関する考慮事項 （439ページ） を参照してください。

[1] post-client-lookup と post-class-lookup 以外のすべて。drop は出力にすぎません。**クライアント参照後**および**クラス後参照**の場合、指定したクライアント クラスが存在する場合、サーバー セットはfalseに設定されます。**クライアントクラス**が存在しない場合はtrue (したがって、拡張の変更がfalseに**ドロップ**しない限り、サーバーはこのパケットの処理を続行しません)。

初期環境ディクショナリ

init-argsとinit-entry. または、環境ディクショナリから読み取る拡張機能の構成情報を指定できます。一連の属性と値のペアを持つDHCPプロパティの初期環境ディクショナリを設定でき、各組み合わせはすべての環境ディクショナリで使用できます。この機能を使用すると、さまざまな構成情報およびカスタマイズ情報を指定できます。任意の拡張は、init-argsまたはinit-entryメソッドで必要とされる静的データ領域に格納しなくても、このデータを環境ディクショナリから直接読み取ることができます。

初期環境ディクショナリーを使用して定義された値を、任意の環境ディクショナリーから読み取ることができます。初期環境ディクショナリに表示される任意の属性に対して新しい値を定義することもできます。これらの新しい値は、その環境ディクショナリ (通常は処理される要求パケットの存続時間) の有効期間で使用できます。ただし、他の環境ディクショナリの内容は変更されません。(別の要求に関連付けられている) 新しい環境ディクショナリは、DHCPサーバーの初期環境ディクショナリプロパティによって定義された属性と値のペアを参照します。

さらに、これらの初期環境ディクショナリ属性と値のペアは、環境ディクショナリの値の列挙には表示されません。これらは、環境ディクショナリーで現在定義されていない属性値を要求する場合にのみ使用できます。属性と値のペアは、実際には環境ディクショナリに表示されません。したがって、いずれかの属性に新しい値を定義すると、その新しい値は環境ディクショナリに表示されます。後で値を削除した場合、元の値は、要求する必要がある場合は再び使用可能になります。

要求ディクショナリと応答ディクショナリ

要求ディクショナリと応答ディクショナリには、アクセス可能な名前の固定セットがあります。ただし、すべての拡張ポイントからすべての名前にアクセスすることはできません。これらのディクショナリは、内部サーバーのデータ構造を拡張機能で読み取り/書き込みアクセス、場合によっては読み取り専用アクセスに使用できるようにします。各データ項目には、特定のデータ型があります。PUT 操作で正しいデータ型 (C/C++ 拡張の場合) を省略した場合、または DHCP サーバーが正しいデータ型 (Tcl 拡張の場合) に変換できない場合、拡張は失敗します。

要求ディクショナリは、要求の処理の開始時に使用できます。サーバーが応答を作成すると、要求ディクショナリと応答ディクショナリの両方が使用できるようになります。応答ディクショナリが使用可能になる前にアクセスするとエラーになります。

一般に、拡張機能を使用してサーバーの情報データを変更することはできません。ただし、拡張機能を使用して構成済みのデータを変更できる場合もありますが、その 1 つの要求に対してのみ処理を行う間のみです。

DHCP 拡張ディクショナリには、受信したクライアント要求 (要求ディクショナリ) と送信された応答 (応答ディクショナリ) で使用できるオプションとデータ項目の詳細が記載されています。

関連項目

復号化された DHCP パケット データ項目 （465 ページ）
パラメータ リスト オプションの使用 （466 ページ）

復号化された DHCP パケット データ項目

DHCP プロトコルは要求-応答 UDP ベースのプロトコルであり、したがって、DHCP サーバー操作の刺激は通常、クライアントからの DHCP 要求です。通常、そのクライアントに返される DHCP 応答が返されます。

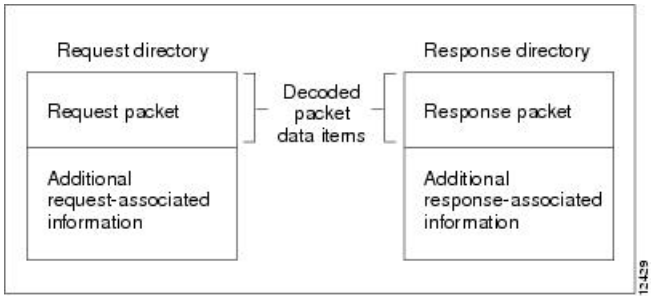
DHCP 拡張機能は、DHCP 要求の情報入力を、ほとんどの拡張ポイントで拡張に対して使用可能にし、pre-packet-encode拡張ポイントで利用可能な DHCP 要求への応答として送信される情報をpre-packet-encode （481 ページ）提供します (を参照してください)。

この DHCP パケット ベースの情報に加えて、DHCP 要求を処理するときに DHCP サーバーが使用する追加データがあります。このデータは、サーバーのアーキテクチャの一部として、DHCP 要求または DHCP 応答のいずれかに関連付けられます。このデータの多くは拡張でも利用できるようになっており、その多くは読み取りと書き込みの両方が可能です。

したがって、要求ディクショナリと応答ディクショナリには、各ディクショナリに 2 つのクラスのデータが含まれています。これらは、デコードされたパケットデータ項目だけでなく、他の要求または応答に関連付けられたデータ項目が含まれています。デコードされたパケットデータ項目は、DHCP 要求または DHCP 応答に直接含まれている、または DHCP からのデータ

項目です。デコードされたパケットデータ項目にアクセスすると、DHCP要求パケットとDHCP応答パケットを読み取り、場合によっては書き換えができます。次の図は、要求ディクショナリと応答ディクショナリの関係を示しています。

図 *16 :* 拡張機能の要求と応答の辞書



要求ディクショナリのデコードされたパケット データ項目を使用して、giaddr、 ciaddr、およびすべての着信DHCPオプションなどのDHCP要求パケットからの情報にアクセスできます。同様に、giaddrとciaddrを設定し、応答ディクショナリ内のデコードされたパケット データ項目にアクセスして、発信 DHCP 応答の DHCP オプションを追加および削除できます。

デコードされたパケットデータ項目によって提供されるパケット情報へのアクセスは、すべて利用できるわけではないことを認識することが重要です。その拡張ポイントで使用できる特定のデータ項目は、各拡張ポイントの説明に一覧表示されます。デコードされたパケットデータ項目は常にグループとしてアクセス可能であるため、グループとして一覧表示されます。

名前によって DHCP オプションにアクセスします。このオプションが存在しない場合、サーバーはそのオプションのデータを返しません。デコードされた要求またはデコードされた応答にオプションを配置すると、put 操作でデータを既存のデータに追加する場合を除き、デコードされた要求またはデコードされた応答に既に同じ名前を持つオプションが置き換えられます。

一部の DHCP オプションには、複数の値を指定できます。たとえば、ルーター オプションには、1つ以上のIPアドレスを関連付けることができます。これらの複数の値へのアクセスは、オプション名に対するインデックス付きの操作によって行われます。

ヒント　要求clearまたは応答ディクショナリの操作は、デコードされたパケットのすべてのオプションを削除します。

パラメータ リスト オプションの使用

DHCPサーバーが特別に処理するオプションdhcp-パラメータ要求リストは、次のいずれかの方法で処理します。

- 名前の下のバイトの複数値のオプションdhcp-パラメータ要求リスト。
- 名前の下の BLOB (バイトのシーケンス) オプションは、名前の下で- dhcp パラメーター要求リスト blobです。

いずれかの名前を使用してオプションを取得または設定できます。DHCP サーバーは、応答ディクショナリ内で、要求ディクショナリ内と異なる方法でdhcp パラメーター要求リスト(およびその-blobバリアント) を処理します。要求ディクショナリでこのオプションにアクセスすると、要求ディクショナリ内の別のDHCPオプションにすぎません。ただし、応答辞書では、特別な処理が行われます。

応答ディクショナリのdhcpパラメータ要求リストオプションを使用して、DHCPまたはBOOTPクライアントに返されるオプションの順序を制御できます。応答辞書にオプションを入れると、DHCP サーバーは既存のオプションを並べ替えて、オプションにリストされているオプションが最初に、リストに表示される順序になるようにします。その後、残りのオプションは、リスト内の最後のオプションの後に現在の順序で表示されます。DHCPサーバーはリストを保持し、リストを新しいものに置き換えるまで、そのリストを使用して、応答に入れる将来のオプションを並べ替えます。

拡張機能は、応答ディクショナリ内のdhcpパラメーター要求リストの取得操作を行う場合、オプションを検索するためにデコードされた応答パケットを検索しません。代わりに、DHCPサーバーは、デコードされた応答パケットに現在含まれているすべてのオプションのリストを含む 1 つを合成します。

拡張ポイントの説明

以下のセクションでは、各拡張ポイント、それらのアクション、およびデータ項目について説明します。すべての拡張ポイントについて、環境ディクショナリで トレースextension-pointレベルのデータ項目値を読み取り、設定できます。ほとんどの拡張ポイントでは、パケットをドロップするようにサーバーに指示することもできます。

関連項目

インイット・エントリー （468 ページ）
post-packet-decode （470 ページ）
事前パケットデコード （469 ページ）
ポストクラスルックアップ （473 ページ）
pre-client-lookup （474 ページ）
ポスト クライアント ルックアップ （476 ページ）
リースの生成 （477 ページ）
check-lease-acceptable （480 ページ）
リース状態の変更 （481 ページ）
pre-packet-encode （481 ページ）
ポスト パケット エンコード （482 ページ）
ポスト送信パケット （483 ページ）

環境デストラクタ （483 ページ）

インイット・エントリー

拡張init-entryポイントは、DHCP サーバーが拡張機能を構成または構成解除するときに呼び出す追加のポイントです。このエントリ ポイントは、拡張機能の他のエントリ ポイントと同じシグネチャを持ちますが、環境ディクショナリのみを使用できます。拡張機能init-entryinit-entrydhcpを CLI で設定するのではなく、既に構成済みの拡張機能に定義することで暗黙的に設定attachExtensionします。

（注） DHCPv6init-entryの拡張ポイントを有効にする (またはDHCPv4 の場合は拡張ポイントを無効にする) 拡張ポイントを指定する必要があります。

エントリ ポイントの名前をinit-entry持つ を構成するだけでなく、init-entryポイントを呼び出す前に、DHCPサーバーが環境ディクショナリの文字列引数の下に読み込む引数の文字列を構成することもできます。引数を使用すると、異なる初期化引数を指定してカスタマイズされた拡張を作成できるため、異なる動作を引き出すためにコードを変更する必要はありません。

（注） サーバーが拡張ポイントで拡張機能をinit-entry呼び出す順序は、リロードからリロード、リリースからリリースまで異なる場合があります。

注意 拡張は、uninitializeに呼び出されたときに、作成したスレッドを終了し、それ自体の後にクリーンアップしてから戻る必要があります。拡張が返されると、DHCPサーバーは、拡張機能をメモリからアンロードします。

init-entry の環境ディクショナリ

init-entryに固有の環境ディクショナリ データ項目については、次の表を参照してください。

表 *49 : init-entry* 環境ディクショナリ データ項目

環境ディクショナリ データ項目	説明
dhcp サポート(入力/出力)	サーバーが拡張機能の登録済み拡張ポイントを呼び出す必要があるDHCP のバージョンまたはバージョン。にはv4、、v6またはv4,v6、 を指定できます。

環境ディクショナリデータ項目	説明
終了状態(出力)	**リース状態変更**拡張ポイントに接続された拡張の場合、指定した場合、**リース状態変更**拡張ポイントは、リースの現在の状態がexiting-stateで指定された状態である場合にのみ呼び出されます。拡張は、指定された状態が終了した場合にのみ呼び出されます。指定されていない場合、拡張が**リース状態変更**拡張ポイントにアタッチされている場合、すべての状態変更に対して拡張が呼び出されます。指定した場合、終了状態は有効なリース状態(利用可能、提供済み、リース済み、期限切れ、使用不可、解放済、その他利用可能、保留状態、取り消し済み)を指定する必要があります。 **注**:厳密な状態遷移表はありません。フェールオーバー環境では、バインド更新メッセージを受信するサーバーは、特定の状態遷移を必要とせずに、パートナーが通知する通知に状態を設定します。
拡張拡張-拡張-バージョン(出力)	拡張機能に必要な拡張機能 API の最小バージョン。現在の2API バージョンに設定します。
イニット引数(入力)	既存の拡張ポイントにinit-argsを設定して、引数を構成します。これらの引数は、エントリ ポイントの設定呼び出しとinit-entry構成解除呼び出しの両方に対して存在します。 構成呼び出しの拡張ポイント名initializeは です。 uninitialize
サーバー dhcp サポート(入力)	サーバーは、このデータ項目を設定して、サーバーがサポートするように構成されていることを示します。DHCPv4v6サーバー DHCP v4,v6 サポート属性設定 (エキスパート属性の可視性が 3 の設定が必要) と、プレフィックスがコンフィグレーションされているかどうかに応じて、、または 、、または 、以下を指定できます。 • dhcp サポート=bothおよびプレフィックスが構成されていない場合、サーバー-dhcp サポートは にv4設定されます。 • dhcp サポート=bothで 1 つ以上のプレフィックスが設定されている場合、サーバー DHCPv4,v6サポートは に設定されます。 • dhcp サポート=v4の場合、サーバー dhcp サポートはv4に設定されます。 • Dhcpサポート= と1つ以上のプレフィックスが設定されている場合 v6 、サーバー dhcp サポートはに設定され v6 ます。
サーバー拡張-拡張-バージョン(入力)	サーバー拡張 API のバージョン。値は2です。

事前パケットデコード

pre-packet-decode で使用可能なディクショナリは、要求と環境です。

この拡張ポイントは、要求が到着したときにDHCPサーバーが最初に検出したポイントです。サーバーは、パケットを受信した後、(拡張ポイントで) パケットをデpost-packet-decodeコードする前に呼び出します。拡張機能は、この拡張ポイントを使用してパケットを検査し、サーバーがデコードする前にパケットを変更したり、サーバーにドロップを発生させることができます。

要求ディクショナリの 2 つの主要なデータ項目はpre-packet-decode、クライアント パケットとパケットです。これらは、受信したパケットを調べて、パケットを変更し、それを書き戻すために使用できます。

注意 要求ディクショナリクライアントパケットとパケットデータアイテムpre-packet-decodeは、要求ディクショナリを持つ任意の拡張ポイントで使用できます。ただし、パケットを変更したり、pre-packet-decode以外の拡張ポイントで置き換えたりすることは、予期しない副作用を引き起こす可能性があるため、直接変更したり、置き換えたりしないでください。たとえば、サーバーがパケットに対する変更を取得しない場合や、処理中にオプションデータが予期せず変更される場合があります。

getBytes をクライアントパケットまたはパケットで使用する拡張は、返されたバッファに書き込みによってパケットのバイト数を直接変更します。ただし、拡張子はputまたはputBytesを使用してパケットの長さを調整する必要があります (パケットが大きすぎる場合は操作が失敗する可能性があります)。DHCPv6 の場合、パケットのクライアント部分の長さを調整する場合、リレーされる場合は、パケット内のリレー メッセージ オプションの長さを更新する必要があります。

パケットの解析を処理して必要なものを見つけ、意図している場合はパケットを適切に変更する必要があります。

サーバーは受信パケットをまだデコードしていないため、ほとんどの要求ディクショナリ データ項目は使用できません(通常は、サーバーが受信パケットからパケットを入力するため)。したがって、この拡張ポイントはパケットから直接データを抽出する必要があります。また、拡張子は正しくフォーマットされていないパケットを処理する必要があります。

着信パケット詳細ログを有効にすると、サーバーはこの拡張ポイントで登録された拡張機能を呼び出した後に受信パケットをログに記録します。DHCP サーバーのデバッグ トレースが 3 以上の場合、少なくとも 1 つの拡張機能が登録されている場合、サーバーはこの拡張ポイントに登録されている拡張機能を呼び出す前にパケットもログに記録します。

注意 この拡張は、受信したパケットが何らかの方法で検証される前に、そのパケットにアクセスします。したがって、拡張は、完全または部分的に無効なDHCPパケットを処理するように記述する必要があります。

post-packet-decode

post-packet-decode で使用可能なディクショナリは、要求と環境です。

拡張の説明

この拡張ポイントは、入力パケットのデコードの直後に続き、パケット内のデータに対する処理の前に行われます。この時点での拡張機能の主なアクティビティは、入力パケットから情報を読み取り、それを使用して何かを行う操作です。たとえば、入力パケットを書き換えるために使用できます。

post-packet-decode拡張ポイントは、使用する最も簡単な拡張ポイントの 1 つです。入力 DHCP または BOOTP パケットの書き換えとしてサーバーの動作の変更を表現できる場合は、この拡張ポイントを使用する必要があります。パケットはデコードされたが、処理されていないため、副作用の数は非常に限られています。

post-packet-decodeデコードされた入力パケットを変更し、サーバーがすべての変更を認識できるようにする唯一の拡張ポイントです。拡張機能でパケットをドロップし、環境ディクショナリのドロップデータ項目を使用して、さらに処理を終了させることができます。

クライアント ID の上書き

クライアント識別子 (ID) をオーバーライドするには、クライアント クラスのオーバーライド-クライアント ID属性の式の値を設定するか、拡張ポイントでオーバーライドクライアント ID データ項目をpost-packet-decode使用します。拡張メソッドは、クライアントをサーバーとは異なる識別子にマップします。

オーバーライド クライアント ID を文字列として取得または配置できる拡張データ項目のバリエーションがあります。また、オーバーライドクライアント ID のデータ型は、読み取り専用のオーバーライドクライアント ID データ型データ項目を使用して要求することもできます。

オーバーライドクライアント ID またはそのオーバーライドクライアントID 文字列バリアントの書き込み方法と取得方法に基づいて、異なる値が返されます (いくつかの例については、次の表を参照してください)。

表 *50*：クライアント *ID* の上書きの書き込みと取得

操作	使用されるデータ項目	値を入れる	値の取得
put	override-client-id	01:02:03:04	
putBytes	override-client-id	01 02 03 04	
get	override-client-id		01:02:03:04 (ブロブ)
getBytes	override-client-id		01 02 03 04 (生バイト)
get [Bytes]	オーバーライドクライアント ID 文字列		01:02:03:04 (blob-as-string)
get [Bytes]	オーバーライドクライアント ID データ型		blob

表 *51 :* クライアント *ID* の上書きの書き込みと取得

操作	使用されるデータ項目	値を入れる	値の取得
put [Bytes]	オーバーライドクライアント ID 文字列	01:02:03:04 テスト	
get [Bytes]	オーバーライドクライアント ID 文字列		01:02:03:04 テスト (文字列)
get [Bytes]	override-client-id		30:31:3a:30:32:3a:30:33:3a:30:34:74:65:73:74 (「01:02:03:04 テスト」のブロブ)
get [Bytes]	オーバーライドクライアントIDデータ型		nstr

同等のクライアントオーバーライドクライアントIDデータ項目 (応答ディクショナリが有効な後の拡張ポイントで使用できます) は、読み取り専用ですが、同じように機能します。

（注） [v6-] オーバーライドクライアント ID式を使用する場合、クライアント IDによる leasequery 要求は、クライアントのリースに関する情報を正しく取得するために、オーバーライドクライアント ID属性を指定する必要があります。

注意 この拡張は、サーバーがパケット構文解析を行った後、検証が適用される前に呼び出されます。したがって、潜在的に無効なパケットを処理するために拡張機能を作成する必要があります。

post-packet-decode の環境ディクショナリ

パケット デコード後に固有の環境ディクショナリ データ項目については、次の表を参照してください。

表 *52*：ポスト パケット デコード環境ディクショナリ データ項目

環境ディクショナリ データ項目	説明
cnr-転送-dhcp 要求(入力)	これらのデータ項目はいずれも DHCPv4 専用です。拡張が戻るときにcnr-forward-dhcp-requestがtrueに設定されている場合、cnr-request-forward-address-listには、サーバーが要求を転送する IPv4 アドレス (およびオプションでポート番号) のリスト (コンマ区切り) を含める必要があります。転送されると、サーバーは要求を破棄します。コンマ区切りリストの各エントリは、ipv4-addressまたはipv4-address:ポート番号(ポート番号が指定されていない場合は、デフォルトの dhcp サーバーポートが使用されます) です。詳細については、DHCP 転送の設定 （26 ページ）を参照してください。
cnr 要求-転送アドレスリスト(出力)	

ポストクラスルックアップ

使用できる辞書post-class-lookupは、要求と環境です。

サーバーはこの拡張ポイントを呼び出す場合、クライアント クラスルックアップ IDが存在する場合に限ります。それ以外の場合は、post-packet-decodeに似ています。サーバーは、クライアントpost-class-lookupクラスルックアップ IDを評価し、このクライアントのクライアント クラスデータを設定した後、拡張ポイントを呼び出します。

この拡張ポイントへの入力時に、環境ディクショナリのドロップデータ項目がtrueまたはfalseに設定されます。この設定を変更して、パケットをドロップ(またはドロップしない)に変更すると、サーバーは変更を認識します。サーバーは、ログ ドロップ メッセージを調べ、ドロップをログに記録するかどうかを決定します。

post-class-lookup の環境ディクショナリ

クラスルックアップ後に固有の環境ディクショナリデータ項目については、次の表を参照してください。

表 *53*：ポストクラスルックアップ環境辞書データ項目

環境ディクショナリ データ項目	説明
client-class-name（出力）	以前のクライアントクラスに関係なく、パケットの名前付きクライアントクラスを設定します。この設定は、拡張ポイントの終了時にドロップ環境ディクショナリデータ項目の値がfalseの場合にのみ有効です。

pre-client-lookup

pre-client-lookup で使用可能なディクショナリは、要求と環境です。

この拡張ポイントは、DHCP サーバーに対してクライアント クラス処理を有効にしている場合にのみ使用できます。この拡張ポイントを使用すると、拡張機能で次のアクションの一部またはすべてを実行できます。

- クライアント クラスの処理中にサーバーが検索するクライアントを変更します。
- 個別のデータ項目を指定して、指定したクライアント・エントリーまたはクライアント・クラスから見つかったデータ項目をオーバーライドします。
- クライアントの検索を完全にスキップするようにサーバーに指示します。この場合、使用されるクライアントデータは、環境ディクショナリで提供される拡張機能のデータだけです。

要求ディクショナリは、この拡張ポイントで実行されている拡張機能の操作に関する決定を行うために使用できますが、環境ディクショナリはすべての操作を制御します。

pre-client-lookup の環境ディクショナリ

次の表の環境ディクショナリ データ項目は、クライアントおよびクライアント クラス pre-client-lookupで使用できるコントロール データ項目です。

表 55 : 事前クライアントルックアップ環境ディクショナリがデータ項目をオーバーライドする で環境ディクショナリ データ項目を設定すると、その値はクライアントルックアップ (内部データベースまたは LDAP のいずれか) から決定された値よりも優先されます。ディクショナリに何も追加しない場合、サーバーはクライアントルックアップで使用可能な内容を使用します。

表 *54* : 事前クライアントルックアップ環境辞書コントロールデータ項目

環境ディクショナリ データ項目	説明
クライアント指定子(入力/出力)	クライアント クラスの処理コードが CNRDB または LDAP でルックアップするクライアントの名前。この拡張ポイントで名前を変更すると、DHCP サーバーは指定したクライアントを検索します。
デフォルトクライアントクラス名(出力)	次の場合に、デフォルトクライアントクラス名オプションに関連付けられた値をクラス名として使用するようにサーバーに指示します。 • pre-client-lookupクライアント指定子のデータ項目がスクリプトで指定されていません。 • サーバーは特定のクライアントエントリを見つけることができなかった。 その後、default-client-class-nameデータ項目は、デフォルトクライアントに関連付けられたクラス名よりも優先されます。

環境ディクショナリ データ項目	説明
スキップクライアントルックアップ(入力/出力)	値は、サーバー構成によって決まります。にtrue設定すると、DHCPサーバーは、この拡張機能の終了時にすぐに実行される通常のクライアントルックアップをスキップします。 このクライアントを記述するために使用されるデータ項目は、環境ディクショナリに置かれるものだけです (下の表を参照)。

表 *55 :* 事前クライアントルックアップ環境ディクショナリがデータ項目をオーバーライドする

環境データ項目	説明
アクション(出力)	この文字列を数値に変換し、結果をアクションとして使用します。使用できる数値は0(なしの場合)、(1除外の場合) です。
認証終了(出力)	1970 年 1 月 1 日からの絶対時間 (秒単位)。クライアント認証の有効期限を示すために使用します。 クライアント認証の有効期限が切れると、DHCP サーバーはクライアント クラスではなく、クライアントの認証されていないクライアント クラス オプションの値を使用して、クライアント エントリに不足しているデータ項目を入力します。
クライアント クラス名(出力)	このデータ項目で指定されたクライアント クラスを使用して、クライアント エントリの不足している情報を入力します。指定された名前に対応するクライアント クラスがない場合、DHCP サーバーは警告をログに記録し、処理を続行します。 を指定noneすると、DHCP サーバーはクライアント エントリにクライアント クラス名が含まれていないかのように動作します。
ドメイン名(出力)	このドメイン名は、DNS 更新の構成で指定されたクライアント DNS 操作よりも優先して使用します。スコープまたはプレフィックスのドメインのプライマリ サーバーとして表示される DNS サーバーは、指定したドメインのプライマリ サーバーである必要があります。 クライアントまたはクライアント クラスのエントリでドメイン名が上書きされない場合、DHCP サーバーはスコープまたはプレフィックスのドメイン名を使用します。 クライアント エントリまたは拡張機能に という単語noneが含まれている場合、DHCP サーバーはスコープまたはプレフィックスのドメイン名を使用します。

環境データ項目	説明
ホスト名(出力)	入力パケットで指定されたホスト名オプション、またはクライアントまたはクライアントクラスエントリからのデータに優先して、クライアントに対してこれを使用します。 これをにnone設定すると、DHCP サーバーはクライアントまたはクライアント クラスのエントリからの情報を使用せず、クライアント要求の名前を使用します。
ポリシー名(出力)	このポリシーは、クライアント エントリに指定されたポリシーとして使用し、そのクライアントエントリで指定されたポリシーを上書きします。
selection-criteria（出力）	コンマ区切りの文字列のリストで、各文字列はクライアントの選択基準を指定します(この入力パケットに対して)。クライアントが使用するスコープまたはプレフィックスには、これらの選択タグがすべて含まれる必要があります。 このデータ項目を使用して、クライアントまたはクライアント クラスのエントリで指定された条件をオーバーライドします。この場合、DHCP サーバーは、ローカル データベースまたは LDAP データベースのいずれに格納されているかに関係なく、クライアント エントリの選択基準を使用しません。 このデータ項目を にnone設定すると、DHCP サーバーはパケットの選択タグを使用しません。 これを null 文字列に設定すると、DHCP サーバーは設定されていないものとして扱い、クライアントまたはクライアント クラスエントリの選択基準を使用します。
認証されていないクライアントクラス名(出力)	サーバーがクライアントを認証しない場合に使用するクライアント クラスの名前。認証されていないクライアント クラス名を指定せずに指定する場合は、このデータ項目の値として無効なクライアント クラス名を使用します。 値none またはクライアント クラス名以外の任意の名前を使用できます。DHCP サーバーは、クライアント クラスが存在しないエラーをログに記録します。

ポスト クライアント ルックアップ

post-client-lookup で使用可能なディクショナリは、要求と環境です。

この拡張ポイントを使用して、クライアント クラスの処理操作全体の結果を調べ、その結果に基づいてアクションを実行できます。結果の一部を書き換えたり、パケットをドロップしたりするために使用できます。post-client-lookupクライアントクラスの処理から、拡張ポイントで実行されている拡張から返されるパケットのホスト名をオーバーライドする場合は、要求ディクショナリ内のクライアントが要求したホスト名データ項目に hostname を設定します。これ

により、Cisco Prime Network レジストラーは、そのデータ項目で指定した文字列でパケットが入ってきたかのように、サーバーを検索します。

また、この拡張ポイントを使用して、環境ディクショナリにデータ項目を配置して、pre-packet-encode拡張ポイントで実行されている拡張の処理に影響を与pre-packet-encode （481 ページ）えることができます (を参照) 、 応答パケットに異なるオプションを読み込む場合や、その他のアクションを実行する可能性があります。

post-client-lookup の環境ディクショナリ

に固有の環境ディクショナリ データ項目については、次のpost-client-lookup表を参照してください。

表 *56* : ポスト クライアントルックアップ環境ディクショナリ データ項目

環境ディクショナリ データ項目	説明
クライアント指定子(入力)	クライアント クラスの処理が検索したクライアントの名前。
cnr-ldap クエリに失敗しました(入力)	DHCP サーバーは、クライアント参照後スクリプトが LDAP サーバー障害に対応できるように、LDAP サーバー障害からの回復を容易にするためにこの属性を設定します。 クライアントルックアップ後の DHCP サーバーは、LDAP サーバー true ・エラーが原因で LDAP 照会が失敗した場合にこのフラグを設定します。サーバーが LDAP サーバーから応答を受信した場合、次の 2 つの条件のいずれかが発生します。 • フラグを に設定falseします。 • cnr-ldap-query-failed 属性は、環境ディクショナリに表示されません。

リースの生成

generate-lease で使用可能なディクショナリは、要求、応答、および環境です。この拡張ポイントは、DHCPv6 でのみ使用できます。

この拡張ポイントを使用して、DHCPv6アドレスまたはプレフィックスを生成し、拡張機能でアドレスまたはプレフィックスを制御できます。拡張機能が生成されたアドレス値を返すとき、サーバーは、エクステンションがリースアクティビティを制御していると仮定して、返されるアドレスまたはプレフィックスに対する多くの制限を緩和します。これには、フェールオーバーの制約が含まれます (したがって、奇数アドレスはバックアップによって割り当てることができ、偶数アドレスはメインによって割り当てることができ、他に利用可能なデリゲートされたプレフィックスを割り当てることができます)。拡張機能は、アドレスまたはプレフィックスの委任領域を管理します。

アドレス割りgenerate-lease当てまたはプレフィックスの委任時に拡張機能を呼び出すことを許可するようにプレフィックスが構成されている場合にのみ、サーバーが呼び出します。サーバーがリースの生成拡張機能を呼び出すと、次のようになります。

- サーバーは、応答ディクショナリのプレフィックスコンテキストを、リースが作成されるプレフィックスに設定します。(DEX_PREFIXsetObjectとDEX_INITIALを使用して呼び出すと、このコンテキストに戻ります。
- サーバーがまだリースを作成していないため、リースコンテキストは存在しません。しかしながら、リース結合データ項目、特にリース結合タイプおよびリース結合iaidは利用できる。(DEX_LEASEsetObjectとDEX_INITIALを使用して呼び出すと、リースコンテキストは3つのコンテキスト(リース、バインディング、およびプリフィックス)を設定するため、このコンテキストに戻り、プレフィックスも設定します。
- サーバーは、スキップリース環境ディクショナリデータ項目を false に設定します。
- サーバーは、(読み取り専用の)環境ディクショナリデータ項目を、このリースを作成するためにエクステンションを呼び出した回数 (1 から始まる) に設定します。
- プレフィックスの委任では、次の環境ディクショナリ データ項目を使用できます。
  - prefix-length- プレフィックスの長さ (要求されたプレフィックス長またはデフォルトのプレフィックス長)。
  - default-prefix-length— デフォルトのプレフィックス長(ポリシーからの)
  - longest-prefix-length— 許容される最長プレフィックス(ポリシーから)
  - shortest-prefix-length— (ポリシーからの) 最短許容プレフィックス。

拡張機能が返されるときに、次のことができます。

- 生成されたアドレス環境ディクショナリデータ項目にアドレスを設定して、明示的なアドレス(ステートフルアドレス割り当て用)を要求します。クライアントのアドレスが使用できない場合(つまり、アドレスが既に使用中の場合)、またはプレフィックスに含まれていない場合、サーバーはこの拡張機能を再度呼び出す可能性があります。
- 生成されたプレフィックス環境ディクショナリデータ項目にプレフィックスを設定して、明示的なプレフィックス(プレフィックス委任の割り当て)を要求します。クライアントに対してプレフィックスが使用できない場合、またはプレフィックスに含まれていない場合、サーバーはこの拡張機能を再度呼び出す可能性があります。次の条件の場合、クライアントではプレフィックスを使用できません。
  - プレフィックスが既に使用されている場合
  - 既に委任されている短いプレフィックスに含まれている場合
  - それより長いプレフィックスが既にサーバーによって委任されている場合

  ポリシーで許可されているプレフィックスが短い場合、または長い場合、サーバーはプレフィックスを拒否しません。
  - スキップリース環境ディクショナリデータ項目をtrueに設定して、サーバーがこのプレフィックスのリースを割り当てないようにします。サーバーは次のプレフィックス(存在する場合) に進みます。
  - 上記のいずれかを設定しないことで、通常のアドレス割り当てまたはプレフィックスの委任を許可します。

サーバーは、各リースに対して最大 500 回の拡張ポイントを呼び出します (この制限は、サーバーがランダムにリースを生成するときに現在適用される制限と同じです)。サーバーは、使用できないアドレスまたはデリゲートされたプレフィックス (プレフィックスの範囲外または既に存在する) を提供する場合にのみ、拡張機能を複数回呼び出します。

（注） この拡張ポイントでパケットをドロップするようサーバーに要求することはできません。

generate-lease の環境ディクショナリ

リースの生成に固有の環境ディクショナリ データ項目については、次の表を参照してください。

表 *57*: リース環境ディクショナリ データ項目の生成

環境ディクショナリ データ項目	説明
試み (入力)	サーバーが単一のリースに対してこの内線番号を呼び出す回数。
デフォルトプレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する既定のプレフィックス長を指定します。デフォルトのプレフィックス長に設定します (ポリシー階層から)。
生成アドレス(出力)	サーバーがリースに使用する拡張機能をアドレス指定します。
生成されたプレフィックス(出力)	委任されたDHCPv6プレフィックスは、サーバーがリースに使用する拡張機能を必要とします。
プレフィックスの長さ(入力) に制限	生成されたプレフィックスをクライアントが要求した**プレフィックス長プレフィックス長のプレフィックス長**に制限するようにサーバーが拡張を要求している場合はtrueに設定します。それ以外**の場合は** false。クライアントがプレフィックス長を要求した場合、サーバーは最初に、その長さのデリゲートされたプレフィックスを取得しようとして、**リースの生成**エクステンションを呼び出します。サーバーは、クライアントの要求された長さを最短プレフィックス長と最長プレフィックス長の間に制限することに注意してください。
最長プレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する最長のプレフィックス長を指定します。(Expert モード)の最長プレフィックス長(ポリシー階層からの) に設定します (ポリシー階層から) - デフォルトはデフォルトのプレフィックス長(未設定の場合) に設定されます。
prefix-length（入力）	要求されたプレフィックス長または既定のプレフィックス長に設定します。

環境ディクショナリ データ項目	説明
最短プレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する最短のプレフィックス長を指定します。(expert モード) の最短プレフィックス長(ポリシー階層から) に設定します (ポリシー階層から) - デフォルトはデフォルトのプレフィックス長(未設定の場合) に設定されます。
スキップリース(出力)	拡張機能がサーバーにリースを生成させたくない場合は、trueに設定します。

check-lease-acceptable

使用できる辞書check-lease-acceptableは、要求、応答、および環境です。

この拡張ポイントは、現在のリースがこのクライアントに対して許容されるかどうかをサーバーが判断した直後に取得されます。この拡張機能を使用すると、その操作の結果を調べ、ルーチンが異なる結果を返すようにすることができます。リースの受け入れの決定（453ページ） を参照してください。

check-lease-acceptable の環境ディクショナリ

チェック-リース許容に固有の環境ディクショナリ データ項目については、次の表を参照してください。

表 *58*：チェックリース許容環境辞書データ項目

環境ディクショナリ データ項目	説明
受け入れ可能(入力)	このクライアントでリースが受け入れられるかどうかに応じて、DHCP サーバーが初期化する読み取り/書き込みデータ項目。この結果を読み取り、変更することができます。許容データ項目をtrueに設定すると、それが許容可能であることを示します。falseに設定すると、受け入れられません。
デフォルトプレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する既定のプレフィックス長を指定します。デフォルトのプレフィックス長に設定します (ポリシー階層から)。
最長プレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する最長のプレフィックス長を指定します。(Expert モード)の最長プレフィックス長(ポリシー階層からの) に設定します (ポリシー階層から) - デフォルトはデフォルトのプレフィックス長(未設定の場合) に設定されます。

環境ディクショナリ データ項目	説明
prefix-length（入力）	クライアントが 1 を指定した場合は、クライアントが要求したプレフィックス長を指定します。
最短プレフィックス長(入力)	デリゲートされたプレフィックスの割り当てに使用する最短のプレフィックス長を指定します。(expertモード) の最短プレフィックス長(ポリシー階層から) に設定します (ポリシー階層から) - デフォルトはデフォルトのプレフィックス長(未設定の場合) に設定されます。

リース状態の変更

使用できる辞書lease-state-changeは応答と環境です。

既存の状態は、リース状態応答ディクショナリ データ項目にあります。新しい状態は、環境ディクショナリ データ項目の新しい状態にあります。新しい状態が既存の状態と一致する場合、サーバーは拡張ポイントを呼び出しません。

この拡張ポイントは、主に読み取り専用の目的で使用しますが、他の拡張ポイントが後で取得できるように、環境ディクショナリにデータを配置できます。

またlease-state-change、リースの有効期限など、別の環境ディクショナリを持つことができます。

lease-state-change の環境ディクショナリ

リース状態の変更に固有の環境ディクショナリ データ項目については、次の表を参照してください。

表 *59* : リース状態変更環境ディクショナリ データ項目

環境ディクショナリデータ項目	説明
新しい開始時の状態(入力)	新しい状態の開始時刻。前の状態の開始時刻は、応答ディクショナリのリース開始状態情報データ項目にあります。
新しい状態(入力)	リースの変更後の状態。現在の状態は、応答ディクショナリのリース状態リース情報データ項目にあります。

pre-packet-encode

pre-packet-encode で使用可能なディクショナリは、要求、応答、および環境です。

（注） DHCPv6 再設定メッセージの場合、要求ディクショナリはありません(再構成はサーバーによって開始されたメッセージであるため)。したがって、有効になっている拡張機能は、応答msg タイプの ADVERTISE またはisValidREPLY を調べるか、要求で再設定メッセージが存在することを確認するために使用する必要があります。

（注） 一括およびアクティブなリースクエリ応答の場合、生成されるパケットごとに拡張を呼び出します。そのため、拡張がこれらの応答に対して応答ディクショナリを使用し、メモリ使用率が増加することがあります。一括およびアクティブなリースクエリ応答に対してこの拡張点で拡張が呼び出された場合は、応答ディクショナリを一切使用しないことをお勧めします。拡張により、要求（または環境ディクショナリ）内のメッセージタイプまたはその他の情報を確認することができます。DHCPv4 の場合、要求の dhcp-message-type が 16（一括）か 18（アクティブ）かを確認します。DHCPv6 の場合、msg-type が 14（統一と一括の両方で使用）または 240（アクティブ）かどうかを確認します。

ポスト パケット エンコード

post-packet-encode で使用可能なディクショナリは、要求、応答、および環境です。

（注） DHCPv6 再設定メッセージの場合、要求ディクショナリはありません(再構成はサーバーによって開始されたメッセージであるため)。したがって、有効になっている拡張機能は、応答 msg タイプを調べて、またはisValid要求に対して応答メッセージタイプをチェックして、要求ディクショナリが存在することを確認します。

サーバーは、パケットをエンコードした後、クライアントに送信する前に、この拡張ポイントを呼び出します。これにより、サーバーはパケットをクライアントに送信する前にパケットを検査して変更するか、拡張機能によってサーバーがパケットをドロップする可能性があります(ただし、サーバは内部データとディスク上のデータに変更を加えた可能性があります。六色)。

クライアント パケットおよびパケットデータ項目が、 の要求ディクショナリで説明されているような動作で応答ディクショナリに事前パケットデコード（469 ページ）追加されました。この拡張ポイントは、応答クライアントパケットまたはパケットデータ項目を要求できる唯一のポイントであることに注意してください。また、サーバーはパケットに対して行われた変更を処理しません。サーバは、変更されたパケットをクライアントに送信するだけです。

発信パケット詳細ログを有効にすると、サーバーはこの拡張ポイントで登録された拡張機能を呼び出した後にパケットをログに記録します。DHCP サーバーのデバッグ トレースが X>=3 で設定されている場合、サーバーは、この拡張ポイントに登録されている拡張機能を呼び出す前に、少なくとも 1 つの拡張機能が登録されている場合にのみ、パケットをログに記録します。

（注） 一括およびアクティブなリースクエリ応答の場合、生成されるパケットごとに拡張を呼び出します。そのため、拡張がこれらの応答に対して応答ディクショナリを使用し、メモリ使用率が増加することがあります。一括およびアクティブなリースクエリ応答に対してこの拡張点で拡張が呼び出された場合は、応答ディクショナリを一切使用しないことをお勧めします。拡張により、要求（または環境ディクショナリ）内のメッセージタイプまたはその他の情報を確認することができます。DHCPv4 の場合、要求の dhcp-message-type が 16（一括）か 18（アクティブ）かを確認します。DHCPv6 の場合、msg-type が 14（統一と一括の両方で使用）または 240（アクティブ）かどうかを確認します。

ポスト送信パケット

DHCPpost-send-packet要求/応答サイクルの重大な時間制約の外部で実行する処理には、拡張ポイントを使用します。サーバーは、クライアントにパケットを送信した後、この拡張ポイントを呼び出します。

（注） DHCPv6 再設定メッセージの場合、要求ディクショナリはありません(再構成はサーバーによって開始されたメッセージであるため)。したがって、有効になっている拡張機能は、応答 msg タイプを調べて、またはisValid要求に対して応答メッセージタイプをチェックして、要求ディクショナリが存在することを確認します。

（注） 一括およびアクティブなリースクエリ応答の場合、生成されるパケットごとに拡張を呼び出します。そのため、拡張がこれらの応答に対して応答ディクショナリを使用し、メモリ使用率が増加することがあります。一括およびアクティブなリースクエリ応答に対してこの拡張点で拡張が呼び出された場合は、応答ディクショナリを一切使用しないことをお勧めします。拡張により、要求（または環境ディクショナリ）内のメッセージタイプまたはその他の情報を確認することができます。DHCPv4 の場合、要求の dhcp-message-type が 16（一括）か 18（アクティブ）かを確認します。DHCPv6 の場合、msg-type が 14（統一と一括の両方で使用）または 240（アクティブ）かどうかを確認します。

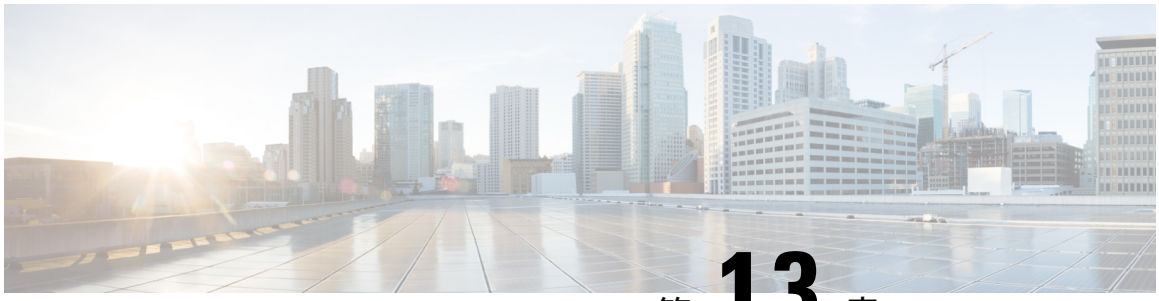
環境デストラクタ

拡張environment-destructorポイントを使用すると、拡張は、そのエクステンションが保持している可能性のあるコンテキストをクリーンアップできます。この拡張ポイントで使用できる唯一のディクショナリは環境です。

環境ディクショナリは、単一のクライアント要求に対して呼び出されるすべての拡張ポイントで使用できます。一部の拡張機能では、単一のクライアント要求のために呼び出される複数の拡張ポイント間のコンテキスト情報を維持する必要があり、サーバーが処理中に複数の場所で

要求をドロップする可能性があるため、拡張機能を確実に解放できないその要求に対して作成した可能性のあるコンテキスト。環境デストラクター拡張ポイントにより、何らかの理由で要求の処理が完了したときに、このコンテキストを確実に削除できるようになりました。

（注） サーバーは、他の接続ポイントでenvironment-destructor各拡張機能を呼び出さなかった場合でも、拡張ポイントに接続されているすべての拡張機能を呼び出します。



第 13 章

DHCPサーバーステータスダッシュボード

Web ユーザー インターフェイス(Web UI)の Cisco Prime Network レジストラーサーバー ステータス ダッシュボードには、トラッキングと診断に役立つグラフ、チャート、テーブルを使用して、システム ステータスのグラフィカル ビューが表示されます。これらのダッシュボード要素は、システム情報を整理および統合された方法で伝達するように設計されており、次の項目が含まれます。

- 重要なプロトコルサーバーおよびその他のメトリック
- アラームとアラート
- データベース インベントリ
- サーバーの正常性の傾向

ダッシュボードは、ダッシュボードを表示するシステムがその目的専用であり、プロトコル サーバーを実行しているシステムとは異なる場合があるトラブルシューティングのデスクコンテキストで使用するのが最適です。ダッシュボード システムは、プロトコル サーバーを実行しているシステムをブラウザでポイントする必要があります。

ダッシュボードインジケーターは、予想される通常の使用パターンからの逸脱を考慮して解釈する必要があります。異常なスパイクやアクティビティの低下に気付いた場合は、ネットワーク上で通信障害や停電が発生して調査する必要があります。

ダッシュボードを開く

ダッシュボード機能は、地域クラスターでも使用できます。既定では、システム メトリック チャートが提供されます。さまざまなクラスタのサーバー固有の(DHCP、DNS、およびCDNS)

チャートを表示できます。これは、[チャートの選択（Chart Selections）] ページで構成できます。

Web UI でダッシュボードを開くには、[操作（Operate）] メニューから [ダッシュボード（Dashboard）] を選択します。

表示タイプ

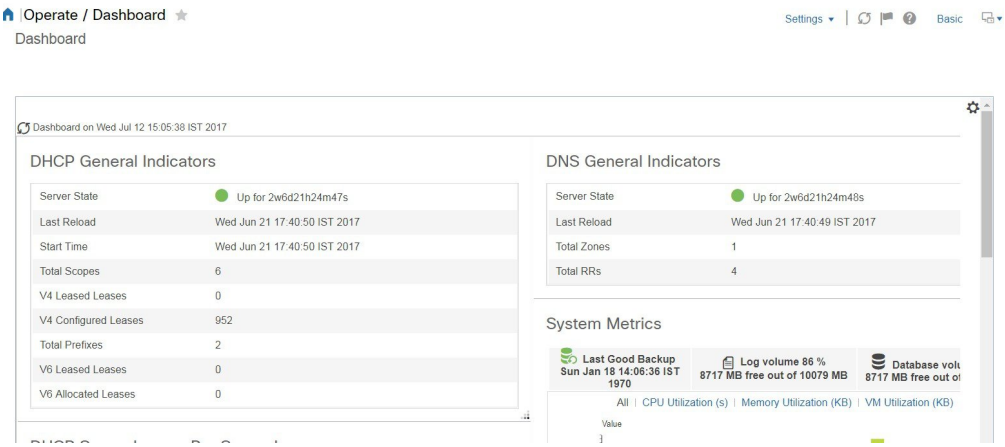
割り当てられた管理者ロールを使用して DHCP および DNS 権限を持っている場合、ダッシュボードのプリセット表示は次の表で構成されます(例については、次の表を参照してください)。

- **システム メトリック**- の「システム メトリックCisco プライムネットワーク レジストラー 11.0 管理ガイド」セクションを参照してください。
- DHCP **一般インジケータ**DHCP 一般指標 （501 ページ）- を参照してください。
- DNS **一般インジケータ**: の『Cisco PrimeNetwork Registrar 11.0 権限のあるキャッシュ DNS ユーザーガイド』の「DNS 一般インジケータ」の項を参照してください。

ヒント

これらは、プリセットの選択です。選択できる他のダッシュボード要素については、「含めるダッシュボード要素の選択 （493 ページ）」を参照してください。ダッシュボードには、セッション間での選択が保持されます。

図 *17:* プリセットのダッシュボード要素



各ダッシュボード要素は、最初は、要素に応じて、テーブルまたは特定のパネルチャートとして表示されます。

- **表**-テーブル （488 ページ）を参照。
- **折れ線グラフ** - 折れ線グラフ （488 ページ） を参照。
- **面グラフ** - 面グラフ （489 ページ） を参照。

一般ステータス インジケータ

上の図のサーバー状態の説明の緑色のインジケータに注意してください。これは、情報を提供するサーバーが正常に機能していることを示します。黄色のインジケータは、サーバーの動作が最適でないことを示します。赤いインジケータは、サーバーがダウンしていることを示します。これらのインジケーターは、通常の Web UI の [サーバーの管理（Manage Servers）] ページのサーバーの状態と同じです。

アラートレベルのグラフィックインジケータ

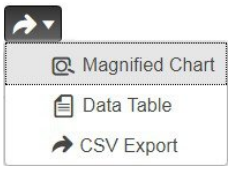
グラフ化された線とグラフの積み上げ領域は、標準の色と視覚的なコーディングに従って、主要な診断インジケータを一目ですぐに判断できます。グラフは、次の色とテキストのインジケーターを使用します。

- High alerts or warnings — 線または赤の領域(ハッチングされたテクスチャ付き)。
- All other indicators — 線や様々な他の色の領域でデータ要素を区別。グラフでは、緑や黄色は使用しません。

グラフの拡大と変換

別のウィンドウでグラフを拡大するには、パネルグラフの下部にある**グラフリンク**アイコンをクリックし、次に「拡大グラフ」オプションをクリックします(下の図を参照)。拡大表示モードでは、最初に表示されるグラフの種類から別のグラフの種類を選択できます(その他のチャート タイプ （490 ページ）を参照)。

図 *18*：拡大グラフ



（注） 拡大されたグラフの自動更新はオフになっています。最新のデータを取得するには、ページの左上にある [ダッシュボード（Dashboard）] の横にある [**更新**（Refresh）] アイコンをクリックします。

グラフを表に変換するには、「表としてグラフを表示する」を参照してください。表をグラフィック・グラフ形式に変換することはできません。

凡例

各グラフには、既定で色分けされた凡例が含まれています。

テーブル

テーブルとして表示されるダッシュボード要素には、行と列にデータが表示されます。以下のダッシュボード要素は、あらかじめ設定されており、テーブルで構成されます(または含める)。

- DHCP DNS の更新
- DHCP アドレスの現在の使用率
- DHCP の一般的なインジケータ
- DNS一般インジケータ
- DNS 一般インジケータのキャッシュ

（注） エキスパートモードでテーブルを表示すると、追加のデータが表示されることがあります。

折れ線グラフ

折れ線グラフとしてレンダリングされるダッシュボード要素には、x 軸と y 軸に対してプロットされた 1 つまたは複数の線を含めることができます。次の表では、3 種類の折れ線グラフについて説明します。

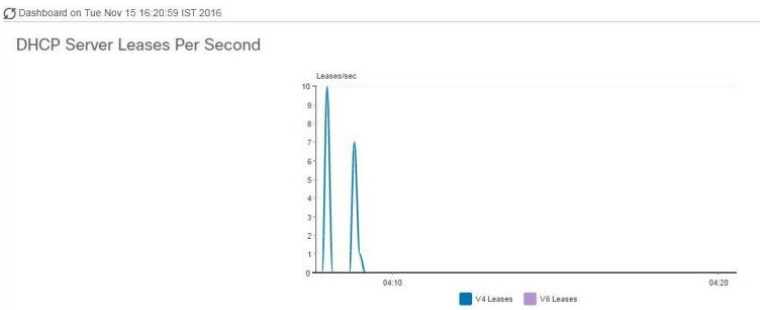
表 *60*：折れ線グラフのタイプ

折れ線グラフの種類	説明	表示されるダッシュボード要素
生データ折れ線グラフ	生データに対してプロットされた線。	• Java 仮想マシン (JVM) メモリー使用率(エキスパート・モードのみ) • DHCP バッファ容量 • DHCP フェールオーバー ステータス(2つのグラフ) • DNS ネットワーク エラー • DNS 関連サーバー のエラー
デルタ折れ線グラフ	2つの連続した生データの差に対してプロットされた線。	• DNS インバウンド ゾーン転送 • DNS アウトバウンド ゾーン転送

折れ線グラフの種類	説明	表示されるダッシュボード要素
レート折れ線グラフ	2つの連続した生データの差に対してプロットされた線は、それらの間のサンプル時間で割った。	• DHCP サーバー要求アクティビティ (下の画像を参照) • DHCP サーバー応答アクティビティ • DHCP 応答遅延 • DNS クエリー応答 • DNS 転送エラー

ヒント デルタまたはレートデータを示すグラフの生データを取得するには、エキスパートモードに入り、必要なチャートに移動します。パネルチャートの下にある [**チャートリンク**（Chart Link）] アイコンをクリックしてから [**データテーブル**（Data Table）] をクリックします。生データテーブルは、グラフデータテーブルの下にあります。

図 *19* : 折れ線グラフの例



面グラフ

面グラフとしてレンダリングされるダッシュボード要素は、複数の関連するメトリックを傾向グラフとしてプロットしますが、一方が積み上げ、最高点が累積値を表すようにします。値は、コントラストの色で個別にシェーディングされます。(面グラフとして図 19 : 折れ線グラフの例 （489ページ）に表示されるDHCPサーバー要求アクティビティ チャートの例については、次の図を参照してください)。

図 *20 :* 面グラフの例

これらは、凡例にリストされている順序で積み重ねられ、スタックの下部に左端の凡例項目、スタックの一番上に右端の凡例項目が表示されます。面グラフに事前に設定されているダッシュボード要素は次のとおりです。

- DHCP バッファ容量
- DHCP フェールオーバーステータス
- DHCP 応答遅延
- 1 秒あたりの DHCP サーバーのリース数
- DHCP サーバー要求アクティビティ
- DHCP サーバーの応答アクティビティ
- DNS 受信ゾーン転送
- DNS ネットワーク エラー
- DNS 送信ゾーン転送
- 1 秒あたりの DNS クエリ
- DNS 関連サーバー エラー

その他のチャート タイプ

選択できるその他のグラフの種類は次のとおりです。

- Line- 折れ線グラフ （488 ページ）で説明した折れ線グラフの 1 つ。
- Area—面グラフ （489 ページ）で説明したグラフ。
- Column- グラフを横方向に垂直バーで表示し、値軸をグラフの左側に表示します。
- Scatter- 散布図は、デカルト座標を使用して、一連のデータの通常 2 つの変数の値を表示するプロットまたは数学図の一種です。

ヒント　各グラフの種類は、異なる方法で、異なる解釈でデータを示しています。どのタイプが最適かを判断できます。

ダッシュボード要素のヘルプの取得

テーブル/グラフウィンドウのヘルプアイコンをクリックすると、各ダッシュボード要素のヘルプウィンドウを開くことができます。

表示のカスタマイズ

ダッシュボードの表示をカスタマイズするには、次の操作を行います。

- データを更新し、自動更新間隔を設定します。
- グラフを展開し、別の形式でレンダリングします。
- グラフィック グラフを表に変換します。
- データをコンマ区切り値 (CSV) 出力にダウンロードします。
- グラフの凡例を表示または非表示にします。
- サーバー グラフの種類を構成します。
- デフォルト表示にリセット

各グラフは次の機能をサポートします。

- サイズ変更
- 新しいセル位置にドラッグ アンド ドロップ
- 最小 化
- クローズ

各グラフには、グラフの説明と、説明の下部にあるリンク (詳細..) をクリックすると詳細なヘルプが表示されたヘルプ アイコンが表示されます。

（注）　ダッシュボード/グラフに加えられた変更は、[ダッシュボード（Dashboard）] ウィンドウで [保存（Save）] をクリックした場合にのみ保持されます。

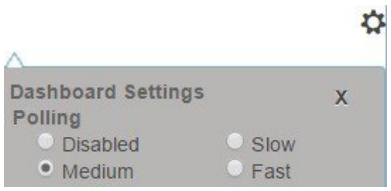
表示の更新

[最新の情報に更新（Refresh）]アイコンをクリックして、最新のポーリングを選択するように各ディスプレイを更新します。

ポーリング間隔の設定

データのポーリング頻度を設定できます。ダッシュボード表示の右上隅の [ダッシュボード設定（Dashboard Settings）] アイコンをクリックします。キャッシュされたデータのポーリング間隔を設定するには、4つのオプションがあり、プロトコルサーバーに更新のポーリングを行います (下の図を参照)。

図 *21 :* グラフのポーリング間隔の設定



キャッシュされたデータポーリング (したがって、自動更新) 間隔を次の値に設定できます。

- Disabled—ポーリングを行わないため、データは自動的に更新されません。
- Slow—30 秒ごとにデータを更新します。
- Medium—20 秒ごとにデータを更新します。
- Fast（プリセット値）—10 秒ごとにデータを更新します。

表としてのグラフの表示

パネルグラフの下部にある [**チャートリンク**（Chart Link）] アイコンを使用して、チャートリンクオプションを表示します(下の図を参照)。[データテーブル（Data Table）] オプションをクリックすると、グラフィック チャートを表として表示できます。

図 *22 :* 表形式へのグラフ変換の指定

CSV形式へのエクスポート

グラフデータは、カンマ区切り値（CSV）ファイル（スプレッドシートなど）にダンプできます。パネルグラフの下部にあるチャートリンクコントロール(上の図を参照)で、[CSV形式でエ

クスポート（CSV Export）]オプションをクリックします。[名前を付けて保存（Save As）]ウィンドウが表示され、CSV ファイルの名前と場所を指定できます。

含めるダッシュボード要素の選択

ページに表示するダッシュボードエレメントの数を決定できます。DHCP サーバーや DNS サーバーなど、1 つのサーバーのアクティビティのみに集中し、他のサーバーの、他のすべてのメトリックを除外する場合があります。このように、ダッシュボードの混雑が少なくなり、要素が大きくなり、読みやすくなります。それ以外の場合は、すべてのサーバーアクティビティの概要を表示し、結果として小さな要素を表示する場合もあります。

[ダッシュボードの設定（Dashboard Settings）]アイコンをクリックし、[ダッシュボードの設定（Dashboard Settings）]ダイアログの [チャート選択（Chart Selections）] をクリックすると、メインの [ダッシュボード（Dashboard）] ページから表示するダッシュボード要素を選択できます。リンクをクリックすると、[チャートの選択（Chart Selection）] ページが開きます（図 23 : ダッシュボード要素の選択 （494 ページ） を参照）。

サーバー チャート タイプの設定

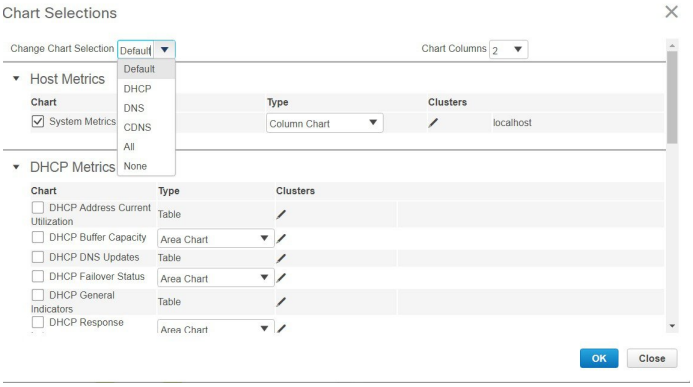
メインダッシュボードビューでデフォルトのグラフタイプを設定できます。ダッシュボードのサーバー・グラフをカスタマイズして、特定のグラフ・タイプのみをデフォルトとして表示できます。

既定のグラフの種類を設定するには、表示するメトリック ス グラフに対応するチェック ボックスをオンにし、Type ドロップダウンリストからグラフの種類を選択します。既定のグラフの種類は、さまざまなユーザーセッション間で一貫性があり、共有されます（下の図を参照）。

（注） サーバーで構成されたサービスに基づいて、[ダッシュボードの設定（Dashboard Settings）] > [グラフの選択（Chart Selection）] ページで CDNS または DNS メトリックを確認できます。

ヒント ダッシュボード要素がグラフの選択リストに表示される順序は、必ずしもページ上での要素の表示順序を決定するものではありません。使用可能な領域を考慮するアルゴリズムによって、グリッドレイアウトの順序とサイズが決まります。ダッシュボード要素の選択を送信するたびにレイアウトが異なる場合があります。選択を変更するには、表示するダッシュボード要素の横にあるチェックボックスをオンにします。

図 *23 :* ダッシュボード要素の選択



上の図は、リージョン Web UI のグラフ選択テーブルを表示します。[クラスター（Clusters）] 列は、リージョン ダッシュボードでのみ使用でき、構成されているローカル クラスターの一覧が表示されます。ローカル クラスターを追加するには、[編集（Edit）] アイコンをクリックし、[ローカル クラスターリスト（Local Cluster List）] ダイアログ ボックスでローカル クラスター名を選択します。

選択を変更するには、表示するダッシュボード要素の横にあるチェックボックスをオンにします。

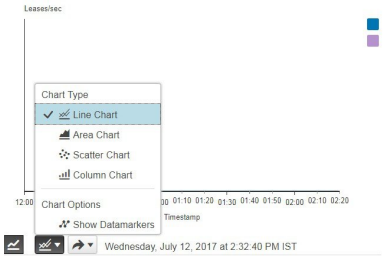
ページの上部にある [チャート選択の変更（Change Chart Selection）] ドロップダウンリストで特定のグループ コントロールを使用できます（上の図を参照）。その内容は：

- すべてのチェックボックスをオフにするには、[なし（None）]を選択します。
- プリセットの選択に戻すには、[デフォルト（Default）]を選択します。DHCP および DNS をサポートする管理者ロール用の事前設定されたダッシュボード要素は次のとおりです。
  - ホストメトリック: システムメトリック
  - DHCP メトリック：一般的なインジケーター
  - DNS メトリック：一般的なインジケーター
- DHCP メトリックのみを選択し、DHCP を選択します（『Cisco Prime Network Registrar 11.0 DHCP User Guide』の「DHCP Metrics」の項を参照）。
- DNS メトリックのみを選択し、DNS を選択します（『Cisco Prime Network Registrar 11.0 Authoritative and Caching DNS User Guide』の「Authoritative DNS Metrics」の項を参照）。
- DNS メトリックのみを選択し、CDNS を選択します（『Cisco Prime Network Registrar 11.0 Authoritative and Caching DNS User Guide』の「Caching DNS Metrics」の項を参照）。
- すべてのダッシュボード要素を選択するには、[すべて（All）]を選択します。

ページの下部にある [OK] をクリックして選択内容を保存するか、または [キャンセル（Cancel）] をクリックして、変更をキャンセルします。

グラフの種類を変更するには、パネル チャートの下部にある [グラフの種類（Chart Type）] アイコンをクリックし、必要なグラフの種類を選択します（下の図を参照）。使用できるグラフには、折れ線グラフ、棒グラフ、面グラフ、散布図があります。

図 *24* : グラフの種類の選択



DHCP メトリック

次の DHCP メトリック要素は、ダッシュボードで使用できます。DHCP サーバー統計情報の完全なリストについては、付録「Cisco プライムネットワーク レジストラー 11.0 管理ガイド のサーバー統計情報（Server Statistics" appendix of）」の「DHCP Statistics（DHCP統計情報）」の項を参照してください。

- DHCP **アドレスの現在の使用率**(を参照)DHCP アドレスの現在の使用率 （495 ページ）
- DHCP **バッファ容量**-を参照してください。DHCP バッファ容量 （497 ページ）
- DHCP DNS **更新**プログラム - 「」を参照DHCP DNS 更新 （498 ページ）
- DHCP **フェールオーバー ステータス**—参照DHCP フェールオーバー ステータス （499 ページ）
- DHCP **一般インジケータ**- 「」を参照してください。DHCP 一般指標 （501 ページ）
- DHCP **更新データ**- 「」を参照してください。DHCP 更新データ （503 ページ）
- DHCP **応答の遅延**時間 (を参照してください)DHCP 応答遅延時間 （503 ページ）
- 1 **秒あたりの** DHCP **サーバーのリース**数 (を参照)DHCP サーバーの 1 秒あたりのデータのリース （504 ページ）
- DHCP**サーバー要求アクティビティー**-を参照してください。DHCP サーバー要求アクティビティ （505 ページ）
- DHCP**サーバー応答アクティビティー**-を参照してください。DHCP サーバー応答アクティビティ （507 ページ）

DHCP アドレスの現在の使用率

表として表示される DHCP アドレスの現在の使用状況ダッシュボード要素は、特定のアドレス集約の DHCPv4 および DHCPv6 アドレスの使用率 (割り当てられたアドレスの数) を示します。選択タグ。このテーブルは、[チャートの選択DHCP Metrics: DHCP Address Current Utilization] ページで選択した場合に使用できます。

結果の表は、次の情報を示しています。

- Name- 集計名 (または住所)
- In —使用中のアドレスUseの数。
- Total：アドレスの合計数。
- Utilization-利用アドレスの割合。
- Mode(エキスパートモードのみで表示されます)-集約モード(スコープ、リンク、プレフィックス、ネットワーク、または選択タグ)

データの解釈方法

グラフには、スコープ、リンク、またはプレフィックス名、使用中アドレスと合計アドレス、および前の 2 つの列に基づくアドレス使用率の 4 つの列を含むテーブルが表示されます。このグラフは、DHCP サーバー拡張サンプル カウンター属性が有効になっている場合にのみ使用できます。

- スコープ モードの SNMP トラップ構成が適用される場合、[名前]列にスコープ名が表示されます。それ以外の場合は、ネットワーク IP アドレスが表示されます。
- トラップが有効になっていない場合 (または DHCP サーバーのデフォルトフリーアドレス構成またはv6-default-free-address-config属性が設定されていない場合)、ネットワーク・アドレスにはアスタリスク ()*が付加されます。
- 選択タグを適用すると、その名前も追加されます。SNMP トラップの詳細については、SNMP 通知イベントの処理のCisco プライムネットワーク レジストラー 11.0 管理ガイド項を参照してください。
- デフォルトフリーアドレス設定(またはv6-デフォルトフリーアドレス設定)属性を定義しない場合、Cisco Prime Network レジストラーは、 という名前default-aggregation-addr-trap-configの内部のリストされていないトラップ設定を作成します。

このため、作成するトラップ構成には default-aggregation-addr-trap-config という名前を使用しないでください。

結果に基づくトラブルシューティング

使用率のアドレスの割合が高い場合、アドレスは飽和点に達します。別のスコープからアドレスを再割り当てする必要がある場合があります。

使用される属性

このグラフの生成には、DHCPScopeAggregationStats クラスの次の属性が使用されます。

表 *61 :* 使用される属性

DHCPScopeAggregationStats 属性	説明
name	スコープまたはプレフィックスの集約の名前を指定します。
in-use-addresses	使用中のアドレスの数を表示します。

DHCPScopeAggregationStats 属性	説明
total-addresses	集約内のアドレスの総数を表示します。
utilized-pct	集約の使用率を表示します。
mode	「スコープ」モードでは、各スコープが独自の空きアドレスレベルを個別に追跡します（デフォルト）。これはIPv4専用のモードです。 「ネットワーク」モードでは、このオブジェクトによって設定されたすべてのスコープが、「プライマリサブネット」を共有している場合、それらのフリーアドレスレベルを集約します。これは IPv4 専用のモードです。 「selection-tags」グループ化では、スコープがプライマリサブネットを共有している場合、および選択タグのリストが正確に一致している場合、スコープによってそれらのフリーアドレス情報が集約されます。これはIPv4専用のモードです。 「プレフィックス」モードでは、各プレフィックスが独自の空きアドレスレベルを個別に追跡します。これは IPv6 専用のモードです。 「リンク」モードでは、このオブジェクトで設定されたすべてのプレフィックスがリンクを共有している場合、フリーアドレスレベルが集約されます。これは IPv6 専用のモードです。 「v6-selection-tags」グループ化では、プレフィクスがリンクを共有している場合、および選択タグのリストが正確に一致している場合、プレフィックスによってフリーアドレス情報が集約されます。これは IPv6 専用のモードです。 「countonly」グループおよび「v6-countonly」グループは、トラップが設定されていない場合に、最も使用率の高い情報を提供するのに使用される、組み込み集約オブジェクトに使われます。これらのモードではトラップは発生しません。

DHCP バッファ容量

領域グラフとして表示される DHCP バッファキャパシティ ダッシュボード要素には、割り当てられた要求と応答の数と、使用中の要求と応答の数をプロットする折れ線グラフが表示されます。この要素は、[チャートの選択DHCP Metrics: DHCP Buffer Capacity] ページで選択した場合に使用できます。

結果の表とグラフのプロット:

- -Requests使用中の要求バッファの数inのUse傾向。
- -Responses使用中の応答バッファの数inのUse傾向。

データの解釈方法

DHCP バッファ容量データは、DHCP 要求バッファと応答バッファの使用パターンを示します。バッファーが異常パターンで増加し始めた場合、割り振られたバッファーの数を増やすことによって補正を試みることなく、取ることができる対策があります。

結果に基づくトラブルシューティング

バッファのしきい値を超えて増加し、一貫して超えている場合は、サーバーの実行速度が遅い理由を見つけます。高レベルのロギング、低速のDHCP拡張またはLDAPサーバー、または、チャットクライアントやケーブルモデム終端システム(CMTS)の頻繁な再起動など、過負荷が考えられます。バッファー・サイズを増やす必要がある場合があります。

使用される属性

このグラフの生成には、DHCPServerActivityStats クラスの次の属性が使用されます。

表 *62 :* 使用される属性

DHCPServerActivityStats 属性	説明
request-buffers-in-use	統計情報の計算時にDHCPサーバーが使用している要求バッファの数を表示します。
response-buffers-in-use	統計情報の計算時にDHCPサーバーが使用している応答バッファの数を表示します。
要求バッファー割り当て	フェールオーバー機能をサポートするためにサーバーが割り当てた要求バッファの数を表示します。
response-buffers-allocated	フェールオーバー機能をサポートするためにサーバーが割り当てた応答バッファの数を表示します。

DHCP DNS 更新

表として表示される DHCP DNS 更新ダッシュボード要素には、関連する DNS サーバーとその現在の状態、および DNS 更新の保留中の DNS 更新の数が表示されます。この表は、[チャートの選択（Chart Selections）] ページで [DHCP Metrics: DHCP DNS Updates] を選択すると表示されます。

結果の表は、次の情報を示しています。

- Server：関連する DNS サーバーと IP アドレス
- State：関連する DNS サーバーの状態
- Pending — 保留中のUpdates更新の合計数

データの解釈方法

特定の DNS サーバーに対する保留中の更新の高レベルは、サーバーが到達不能または利用不能であるか、またはアドレスが間違っていることを示します。

結果に基づくトラブルシューティング

保留中の更新速度が急上昇した場合は、関連付けられている DNS サーバーの到達可能性を確認するか、関連付けられているサーバーのアドレスが正しいことを確認します。

使用される属性

このグラフの生成には、DNSRelatedServer クラスの次の属性が使用されます。

表 *63 :* 使用される属性

DNSRelatedServer 属性	**説明**
ipaddr/ip6address	DNS サーバーのアドレス。
DNS サーバー状態	DHCP サーバーから見た DNS サーバーの状態を報告します。状態は、SEND-UPDATE または PROBE です。 SEND-UPDATE 状態は、DHCP サーバーが現在 DNS サーバーに DNS 更新を送信していることを示します。 PROBE 状態は、DHCP サーバーが DNS 更新を送信しようとして失敗したか、または DNS サーバーへの DNS アップデートの送信をまだ開始していないことを示します。
要求	この時間間隔で受信した DHCPREQUEST パケットの数を示します。

DHCP フェールオーバー ステータス

DHCPフェールオーバーステータスダッシュボード要素は、現在のサーバーとパートナーサーバーの状態、および2つのフェールオーバー パートナー間で送受信されるバインディングの更新と受信確認を示す 2 つの並行トレンド グラフとして表示されます。グラフは、[グラフの選択]DHCP Metrics: DHCP Failover Statusページで選択した場合に使用できます。

（注） フェールオーバーの状態は、関連サーバーの一覧の最初のフェールオーバーペアに対してのみです。

この画面は、関連サーバーの最初のフェールオーバー ペアのフェールオーバー ステータスを示す 2 つのレート ライン トレンド チャートと一緒に表です。

- Local — ローカル DHCP サーバーのフェールオーバー状態と、それが発生Stateしたタイミング。
- Partner —パートナー サーバーのフェールオーバー状態と、それが発生Stateしたタイミング。
- DHCP Failover -Status最初の傾向グラフは、受信したバインディング更新と送信されたバインディング確認の数の比較を示します。 Updates Received
- DHCP Failover - Status 2 番目の傾向グラフは、送信されたバインディング更新と受信したバインディング確認の数の比較を示します。 Updates Sent

データの解釈方法

いくつかの状態データと共に、表示は互いに逆である2つの折れ線の傾向グラフに分割されます。各グラフは、バインディングの更新を受信確認と比較します。最上位のグラフは、受信したバインディングの更新と送信された受信確認を組み合わせます。下のグラフは、受信した受信確認と送信されたバインディングの更新を組み合わせます。

結果に基づくトラブルシューティング

パートナー状態の値が 10 以外の場合は、パートナー サーバーの構成を確認します。送信および受信したデータの更新も、かなりレベルにする必要があります。

使用される属性

このグラフの生成には、FailoverRelatedServer クラスと DHCPFailoverStats クラスの次の属性が使用されます。

表 *64 :* 使用される属性

属性	説明
FailoverRelatedServer 属性	
state	このフェールオーバー関係の終了が存在する状態。
partner-state	パートナーのフェールオーバー関係の終了が存在する最後の既知の状態。
start-time-of-state	現在のフェールオーバー状態が開始された時刻。
start-time-of-partner-state	パートナーの現在のフェールオーバー状態が開始された時刻。
DHCPFailoverStats 属性	
バインディング更新を受信	この時間間隔で受信されたフェールオーバー DHCPBNDUPD パケットの数を表示します。
バインディング Acks 送信	この時間間隔で送信されたフェールオーバー DHCPBNDACK パケットの数を表示します。

属性	説明
v6-binding-updates-received	この時間間隔で受信されたフェールオーバーBNDUPD6メッセージの数を表示します。
v6-binding-acks-sent	この時間間隔で送信された、更新が否定応答されなかったフェールオーバーBNDUPD6メッセージの数を表示します。
binding-updates-sent	この時間間隔で送信されたフェールオーバーDHCPBNDUPDパケットの数を示します。
バインディング・アクス受信	この時間間隔で受信されたフェールオーバーDHCPBNDACKパケットの数を表示します。
v6-binding-updates-sent	この時間間隔で送信されたフェールオーバーBNDUPD6メッセージの数を表示します。
v6-binding-acks-received	この時間間隔で受信された、更新が否定応答されなかったフェールオーバーBNDUPD6メッセージの数を表示します。

DHCP 一般指標

表として表示されるDHCP一般インジケーターダッシュボード要素は、サーバーの状態、データの再読み込み、およびリース数を示します。この表は、[チャートの選択（Chart Selections）]ページで [DHCP Metrics: DHCP General Indicators] を選択すると表示されます。

結果の表は、次の情報を示しています。

- Server - アップまたはダウン (統計情報が使用可能かどうかに基づく)Stateとその期間。
- Last - 最後のサーバーのリReloadロード日時。
- :最後のサーバー プロセス(Cisco Prime Network レジストラーサーバー エージェント)の起動日時。 Start Time
- Total : 構成済みの DHCPv4 スコープScopesの総数。
- -予約を含むアクティブな DHCPv4 リースの数。 V4 Leased Leases
- -予約と範囲を含む、設定済みの DHCPv4 リースの数。 V4 Configured Leases
- Total :設定済みの DHCPv6 プレフィックスPrefixesの数。
- - 予約と委任されたプレフィックスを含むアクティブな DHCPv6 リースの数(それぞれが 1 つのリースとしてカウントされます)。 V6 Leased Leases
- - 予約と委任されたプレフィックスを含む割り当てられたDHCPv6リースの数(それぞれが 1 つのリースとしてカウントされます)。 V6 Allocated Leases

データの解釈方法

この表は、サーバーの状態、プロセスの開始時刻(Cisco Prime Network レジストラーサーバーエージェント経由)、およびリロード データを示し、リース統計情報も示します。データの上位のセットは、実際に有効なDHCPv4 リースと設定されているリースを比較します。データの下部セットは、DHCPv6 リースでも同じです。

最後の再ロードの時間は、リロード操作からサーバー設定に対する最近の変更が発生したかどうかを判断する上で重要です。また、他のインジケータがマークされた予期しない動作の変更を示している場合、サーバーの変更がいつ最後に適用されたのかを特定するのにも役立ちます。最後の再ロード以降は、ログ ファイルを必ず保持してください。

結果に基づくトラブルシューティング

リースのドロップまたは増加は、電力やネットワークの停止を示す可能性がありますが、リース時間や使用パターンによっては通常の変動を示す場合もあります。示されたスコープまたはプレフィックスの数も、ある程度の評価と可能な再構成を必要とするかもしれません。サーバーの状態が [Down] の場合、すべての DHCP チャート インジケータに赤いステータス ボックスが表示されるため、データは使用できません。サーバーが停止している場合は、サーバーを再起動します。

使用される属性

グラフの生成には、DHCPServerStats クラスと DHCPServerActivityStats クラスの次の属性が使用されます。

表 *65 :* 使用される属性

属性	説明
サーバーの状態	サーバーの状態。
DHCPServerStats 属性	
server-start-time	サーバーの起動時刻。
server-reload-time	サーバーが最後にリロードされた時刻。
DHCPServerActivityStats 属性	
active-leases	新しいクライアントが現在使用できない DHCPv4 のリース数および予約数を示します。
configured-leases	サーバーに設定されている DHCPv4 のリースと予約の数を表示します。これには、設定によって定義されている範囲内のすべての可能なリースが含まれます。
total-scopes	サーバーで設定されているスコープの数。

属性	説明
active-leases	新しいクライアントが現在使用できない DHCPv6 のリース数および委任されたプレフィックスの数を示します。
allocated-leases	サーバーに現在割り当てられている DHCPv6 のリース数、予約数、および委任されたプレフィックスの数を示します。
total-prefixes	サーバーに設定されているプレフィックスの数。

DHCP 更新データ

折れ線グラフとして表示される DHCP 更新データ ダッシュボード要素は、DHCP サーバーで予想される更新の負荷を示します。このグラフは、[グラフの選択] ページで[DHCP **メトリック**: DHCP **更新データ**] を選択した場合に使用できます。

結果の折れ線グラフは次の内容で表示されます。

- **クライアント数**—特定の時間間隔内に更新するクライアントの数。

DHCP 応答遅延時間

領域グラフとして表示される DHCP 応答遅延ダッシュボード要素は、応答パケットの遅延 (要求パケットとその応答の間の時間間隔) の傾向を示します。グラフは、[グラフの選択]DHCP Metrics: DHCP Response Latencyページで選択した場合に使用できます。

ヒント　また、このデータに対しては、サンプル カウンターの収集DHCP サーバー属性を設定し、さらに細分性を高めるために拡張サンプル カウンター属性も設定する必要があります。これらの属性値は事前設定されています。最大のパフォーマンスを実現する心配がある場合は、これらの属性を設定解除します。(の「統計の表示」セクションをCisco プライム ネットワーク レジストラー 11.0 管理ガイド参照してください。

結果の面グラフは、次の間隔で応答の待機時間をプロットします。

- 50 ミリ秒未満
- 50 ～ 200 ミリ秒
- 200 ～ 500 ミリ秒
- 500 ~ 1000 ミリ秒 (拡張サンプル カウンター属性が設定されていない場合、このグループに 1 秒未満の値がすべて表示されることに注意してください)
- 1 ～ 2 秒
- 2~3秒

- 3 ～ 4 秒
- 4秒以上

データの解釈方法

このチャートは、着信パケットに応答するのにかかる時間を示す指標として、応答パケット遅延の傾向を示しています。待機時間内のグラデーションは積み重ねられます。

結果に基づくトラブルシューティング

応答パケットの待ち時間が長い場合は、トラブルシューティングを目的としたバッファの使用率が高い場合と似ています。低速 LDAP サーバーまたは DHCP 拡張機能、高レベルのロギング、またはディスク I/O ボトルネックを探します。

使用される属性

このグラフの生成には、DHCPServerActivityStats クラスの次の属性が使用されます。

表 *66 :* 使用される属性

DHCPServerActivityStats 属性	説明
ack-latency-counts	DHCPACK 応答数の順序付きリスト。
reply-latency-counts	返信応答数の順序付きリスト。

DHCP サーバーの 1 秒あたりのデータのリース

領域グラフとして表示される 1 秒あたりの DHCP サーバーリース数ダッシュボード要素には、DHCP サーバーの 1 秒あたりのリース数が表示されます。このグラフは、[グラフの選択DHCP Metrics: DHCP Server Leases Per Second] ページで選択した場合に使用できます。

結果の面グラフには、次の情報が表示されます。

- V4 — 1 秒あたりの IPv4Leasesリース数。
- V6 — 1 秒あたりの IPv6Leasesリース数。

使用される属性

このグラフの生成には、DHCPServerActivityStats クラスと DHCP6Stats クラスの次の属性が使用されます。

表 *67 :* 使用される属性

属性	説明
DHCPServerActivityStats 属性	

属性	説明
acks	この時間間隔で送信された DHCPACK パケットの数を示します。
DHCP6Stats 属性	
replies	この時間間隔で送信された DHCPv6 応答数を示します。

DHCP サーバー要求アクティビティ

区分グラフとして表示される DHCP サーバー要求アクティビティ ダッシュボード要素は、着信 DHCP パケット アクティビティの変化率の合計をトレースします。グラフは、[グラフの選択] DHCP Metrics DHCPServerRequestActivityページで [:] を選択した場合に使用できます。

結果の面グラフには、次の傾向が表示されます。

- V4 :DHCPv4 ディスカバリ パケットDiscoversの数。
- V4 :DHCPv4 要求パケットRequestsの数。
- V4 DHCPv4 リリース、拒否、または情報要求パケットOtherの数。
- V4 Lease DHCPv4 リース クエリQueriesパケットの数。
- V6 :DHCPv6 送信パケットSolicitsの数。
- V6 :DHCPv6 要求、更新、および再バインド パケットRequests/Renews/Rebindsの数。
- V6 :DHCPv6 リリース、拒否、または情報要求パケットOtherの数。
- V6 Lease DHCPv6 リース クエリQueriesパケットの数。
- Invalid :無効な DHCPv4 パケットと DHCPv6 パケットPacketsの合計数。

データの解釈方法

DHCP サーバー要求アクティビティ データは、着信 DHCP 要求に基づくサーバー トラフィックのパターンを示します。この傾向は、無効なパケットの数が急増し、ネットワーク上に誤って構成されたデータがあることを示す傾向にあるはずです。DHCPv4 と DHCPv6 の無効なパケットアクティビティはグループ化されています。

結果に基づくトラブルシューティング

特に無効な要求パケットの数で、アクティビティが急激に急増している場合は、DHCP サーバーの構成を確認します。アクティビティが発生している場所を報告するようにサーバーログを設定します。活動の急増や低下は、調査する価値のあるネットワークまたは停電を示している可能性があります。アクティビティの急増は、障害のあるクライアント、悪意のあるクライアントのアクティビティ、または、電源障害または停止後の復旧によって、ペントアップ要求が発生したことを示す場合もあります。

使用される属性

このグラフの生成には、DHCPServerActivityStats クラスと DHCP6Stats クラスの次の属性が使用されます。

表 *68 :* 使用される属性

属性	説明
DHCPServerActivityStats 属性	
discovers	この時間間隔で受信した DHCPDISCOVER パケットの数を示します。
要求	この時間間隔で受信した DHCPREQUEST パケットの数を示します。
lease-queries	この時間間隔で受信した DHCPLEASEQUERY パケットの数を示します。
invalid-packets	この時間間隔で受信した無効な DHCP パケットの数を示します。
releases	この時間間隔で受信した DHCPRELEASE パケットの数を示します。
declines	この時間間隔で受信した DHCPDECLINE パケットの拒否数を示します。
インフォーム	この時間間隔で受信した DHCPINFORM パケットの数を示します。
bootp-received	この時間間隔で受信した bootp パケットの数を表示します。
DHCP6Stats 属性	
solicits	この時間間隔で受信した DHCPv6 送信請求の数を示します。
renews	この時間間隔で受信された DHCPv6 更新の数を表示します。
rebinds	この時間間隔で受信した DHCPv6 の再バインドの数を表示します。
leasequeries	受信した DHCPv6 Leasequery メッセージの数を表示します。
invalid-packets	この時間間隔で受信した 無効な DHCPv6 パケットの数を示します。
確認する	この時間間隔で受信した DHCPv6 確認の数を示します。
releases	この時間間隔で受信した DHCPv6 リリースの数を表示します。
declines	この時間間隔で受信した DHCPv6 拒否の数を表示します。
info-requests	この時間間隔で受信した DHCPv6 情報要求の数を表示します。

DHCP サーバー応答アクティビティ

領域グラフとして表示される DHCP サーバー応答アクティビティ ダッシュボード要素は、発信 DHCP パケット アクティビティの変化率の合計をトレースします。グラフは、[グラフの選択] DHCP Metrics DHCPServerResponseActivityページで [:] を選択した場合に使用できます。

結果の面グラフには、次の傾向が表示されます。

- V4 :DHCPv4 オファー パケットOffersの数。
- V4 DHCPv4 確認応答パケットAcksの数。
- 他の発信 DHCPv4 クライアント パケットの数。 V4 Other Client
- 送信 DHCPv4 リース クエリ パケットの数。 V4 Lease Queries
- V6 :DHCPv6 アドバタイズ パケットAdvertisesの数。
- V6 :DHCPv6 応答パケットRepliesの数。
- V6 :DHCPv6 再設定パケットReconfiguresの数。
- V6 Lease DHCPv6 リース クエリ応答パケットの数。 Query Replies
- Total :ドロップされた DHCPv4 パケットと DHCPv6 パケットDroppedの合計数。

データの解釈方法

DHCP サーバー応答アクティビティ データは、DHCP 要求に応答するサーバー トラフィックのパターンを示します。この傾向は、ドロップされた合計パケット数の急増が、ネットワーク上に誤って構成されたデータがあることを示すサインとして、かなり一貫している必要があります。DHCPv4 と DHCPv6 ドロップパケットアクティビティはグループ化されています。

結果に基づくトラブルシューティング

アクティビティが急激に急増している場合、特にドロップされた応答パケットの総数が急激に増加している場合は、DHCPサーバーの構成を確認します。応答アクティビティは、通常の時間シフトを除き、要求アクティビティと一致する必要があり、同じ診断が適用されます。

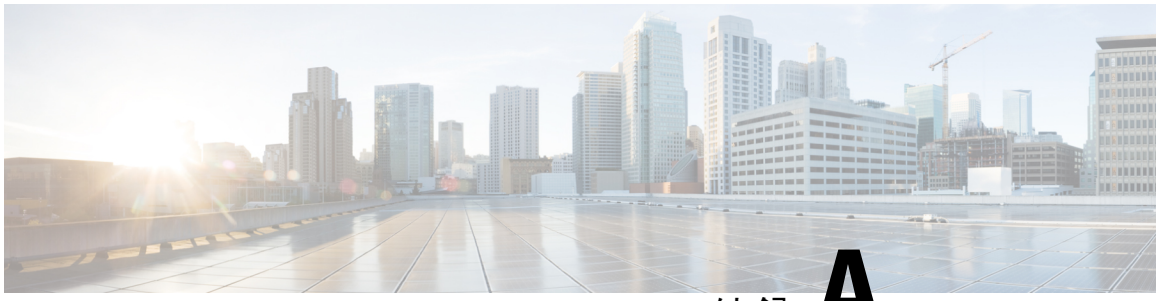
使用される属性

このグラフの生成には、DHCPServerActivityStats クラスと DHCP6Stats クラスの次の属性が使用されます。

表 *69 :* 使用される属性

属性	説明
DHCPServerActivityStats 属性	
offers	この時間間隔で送信されたDHCPOFFERパケットの数を示します。
acks	この時間間隔で送信された DHCPACK パケットの数を示します。
dropped-total	この時間間隔で、サーバーまたはクライアントの設定の問題によりドロップされた DHCP パケットの総数を表示します。

属性	説明
naks	この時間間隔で送信された DHCPNAK パケットの数を表示します。
bootp-sent	この時間間隔で送信された bootp パケットの数を表示します。
lease-queries-unassigned	この時間間隔で送信された DHCPLEASEUNASSIGNED パケット（メッセージ ID 11）の数を表示します。
lease-queries-unknown	この時間間隔で送信された DHCPLEASEUNKNOWN パケット（メッセージ ID 12）の数を表示します。
lease-queries-active	この時間間隔で送信された DHCPLEASEACTIVE パケット（メッセージ ID 13）の数を表示します。
DHCP6Stats 属性	
advertises	この時間間隔で送信された DHCPv6 アドバタイズの数を示します。
replies	この時間間隔で送信された DHCPv6 応答数を示します。
reconfigures	この時間間隔で送信された DHCPv6 再設定の数を表示します。
leasequery-replies	成功したかどうかにかかわらず、DHCPv6 リースクエリメッセージに対する応答の数を表示します。
dropped-total	この時間間隔で、サーバーまたはクライアントの設定の問題によりドロップされた DHCPv6 パケットの合計数を示します。



付録 A

DHCP オプション

DHCP は、TCP/IP ネットワーク上のホストに設定情報を渡すフレームワークを提供します。設定パラメータと他の制御情報は、DHCP メッセージのオプション フィールドに保存されているタグ付きデータ項目で伝送されます。データ項目自体もオプションと呼ばれます。

DHCP オプションには、オプションパラメータの規定された形式と許可値があります。表 70 : 数値による DHCPv4 オプション （510 ページ） および 表 72 : 番号順の DHCPv6 オプション一覧 （530 ページ） リストには、各 DHCP オプションとパラメータの種類が表示されます（[検証] 列に表示されます）。パラメータの形式と許容値は、DHCP およびインターネット RFC から取得されます。すべての DHCP オプションが表示されますが、クライアントは一部のみを制御し、CLI は他のオプションのみを制御します。

次の表は、DHCP オプションをさまざまな方法で示しています。オプションは、数値、Cisco Prime Network Registrar 名別に並べ替えられたものです。

数値による DHCPv4 オプション

次の表は、オプション番号でソートされた DHCPv4 オプションと、検証タイプを示しています。[検証] 列に表示されるオプションの検証の種類の詳細については、「表 74 : 検証タイプ （549 ページ）」を参照してください。[検証]0+ 列の A は、0 以上のオカレンスの繰り返し数、1+ は 1 回以上のオカレンス、2n は、2 の倍数の複数のオカレンスを表します。

ヒント　サブオプションに対してより複雑なオプション・データ値を追加するための構文サブオプションの複雑な値の追加 （208 ページ）については、を参照してください。

表 *70 :* 数値による *DHCPv4* オプション

番号	Cisco Prime Network Registrar 名	検証	説明
[0]	パッド	AT_NOLEN	後続のフィールドを語の境界で揃えるために使用します。RFC 2132 を参照してください。
1	subnet-mask	AT_IPADDR	サブネットマスクを指定します。RFC 2132 を参照してください。
2	time-offset	AT_STIME	クライアントのサブネットのオフセットを、協定世界時（UTC）の秒単位で指定します。RFC 2132 を参照してください。
3	ルータ	AT_IPADDR（1+）	クライアントサブネット上のルータの IP アドレス一覧を指定します。RFC 2132 を参照してください。
4	time-servers	AT_IPADDR（1+）	クライアントが使用できる RFC 868 [6] のタイムサーバー一覧を指定します。RFC 2132 を参照してください。
5	name-servers	AT_IPADDR（1+）	クライアントで使用可能な IEN 116 [7] ネームサーバー一覧を指定します。RFC 2132 を参照してください。
6	domain-name-servers	AT_IPADDR	クライアントで使用可能なドメインネームシステム（STD 13、RFC 1035 [8]）ネーム サーバー一覧を指定します。RFC 2132 を参照してください。
7	log-servers	AT_IPADDR（1+）	クライアントが使用できる MIT-LCS UDP ログサーバーの一覧を指定します。RFC 2132 を参照してください。
8	クッキーサーバー	AT_IPADDR（1+）	クライアントが使用できる RFC 865 [9] の Cookie サーバーの一覧を指定します。RFC 2132 を参照してください。
9	lpr サーバー	AT_IPADDR（1+）	クライアントが使用できる RFC 1179 [10] のラインプリンタサーバーの一覧を指定します。RFC 2132 を参照してください。
10	印象づけるサーバー	AT_IPADDR（1+）	クライアントが利用できる Imagen Impress サーバーの一覧を指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
11	リソース ロケーション サーバー	AT_IPADDR（1+）	クライアントが使用できる RFC 887 [11] リソースロケーションサーバーの一覧を指定します。RFC 2132 を参照してください。
12	host-name	AT_NSTRING	クライアントの名前を指定します。RFC 2132 を参照してください。
13	ブートサイズ	AT_SHORT	クライアントのデフォルトブートイメージの長さを 512 オクテットブロック単位で指定します。RFC 2132 を参照してください。
14	メリットダンプ	AT_NSTRING	クライアントがクラッシュした場合にクライアントのコアイメージをダンプするファイルのパス名を指定します。RFC 2132 を参照してください。
15	domain-name	AT_NSTRING	ドメインネームシステムを介してホスト名を解決するときにクライアントが使用するべきドメイン名を指定します。RFC 2132 を参照してください。
16	スワップサーバー	AT_IPADDR	クライアントのスワップサーバーの IP アドレスを指定します。RFC 2132 を参照してください。
17	root-path	AT_NSTRING	クライアントのルートディスクを含むパス名を指定します。RFC 2132 を参照してください。
18	拡張機能パス	AT_NSTRING	TFTP 経由で取得可能なファイルを指定する文字列です。BOOTP 応答内の 64 オクテットのベンダー拡張フィールドと同じ方法で解釈できる情報が含まれますが、次の例外があります。 • ファイルの長さが制約されていない。 • ファイル内のタグ 18（つまり、BOOTP 拡張パスフィールドのインスタンス）へのすべての参照が無視されます。 RFC 2132 を参照してください。
19	ip-forwarding	AT_BOOL	クライアントが、パケット転送用の IP 層を設定する必要があるかどうかを指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
20	非ローカル ソース ルーティング	AT_BOOL	クライアントが非ローカル ソース ルートでデータグラムを転送できるように IP レイヤを設定するかどうかを指定します。RFC 2132 を参照してください。
21	ポリシー フィルター	AT_IPADDR（2n）	ローカル以外のソースルーティング用にポリシーフィルタを指定します。RFC 2132 を参照してください。
22	マックス・ドグラム再構成	AT_SHORT	クライアントが再構成するために準備する必要がある最大サイズのデータグラムを指定します。RFC 2132 を参照してください。
23	デフォルト-ip-ttl	AT_RANGEBYTE	クライアントが送信データグラムで使用するデフォルトの Time-to-Live（存続可能時間）を指定します。RFC 2132 を参照してください。
24	パス-mtu エージング タイムアウト	AT_TIME	RFC 1191 [12] で定義されているメカニズムによって検出されたパス MTU 値をエージングするときに使用するタイムアウト（秒単位）を指定します。RFC 2132 を参照してください。
25	パス-mtu-プラトーテーブル	AT_RANGESHORT（1+）	RFC 1191 で定義されているパス MTU ディスカバリを実行するときに使用する MTU サイズの表を指定します。RFC 2132 を参照してください。
26	interface-mtu	AT_RANGESHORT	このインターフェイスで使用する MTU を指定します。RFC 2132 を参照してください。
27	すべてのサブネットローカル	AT_BOOL	クライアントが接続先の IP ネットワークのすべてのサブネットが、クライアントが直接接続されているネットワークのサブネットと同じ MTU を使用すると仮定できるかどうかを指定します。RFC 2132 を参照してください。
36	broadcast-address	AT_IPADDR	クライアントサブネットで使用されているブロードキャストアドレスを指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
29	マスク検出の実行	AT_BOOL	クライアントが ICMP を使用してサブネットマスクの検出を実行するかどうかを指定します。RFC 2132 を参照してください。
30	マスクサプライヤー	AT_BOOL	クライアントが ICMP を使用してサブネットマスク要求に応答するかどうかを指定します。RFC 2132 を参照してください。
31	ルーター発見	AT_BOOL	クライアントが RFC 1256 [13] で定義されたルータ発見メカニズムを使用してルーターを要請するかどうかを指定します。RFC 2132 を参照してください。
32	ルーター勧誘アドレス	AT_IPADDR	クライアントがルータ要請を送信するアドレスを指定します。RFC 2132 を参照してください。
33	static-routes	AT_IPADDR（2n）	クライアントがルーティングキャッシュにインストールする静的ルートの一覧を指定します。RFC 2132 を参照してください。
34	トレーラーカプセル化	AT_BOOL	ARP プロトコルを使用する場合に、クライアントがトレーラ（RFC 893 [14]）の使用をネゴシエートするかどうかを指定します。RFC 2132 を参照してください。
35	arp-cache-timeout	AT_TIME	ARP キャッシュエントリのタイムアウト値（秒単位）を指定します。RFC 2132 を参照してください。
36	ieee802.3-encapsulation	AT_BOOL	インターフェイスがイーサネットである場合に、クライアントがイーサネットバージョン 2（RFC 894 [15]）または IEEE 802.3（RFC 1042 [16]）カプセル化を使用するかどうかを指定します。RFC 2132 を参照してください。
37	デフォルト-tcp-ttl	AT_RANGEBYTE	TCP セグメントを送信するときにクライアントが使用するデフォルトの TTL を指定します。RFC 2132 を参照してください。
38	tcp キープアライブ間隔	AT_TIME	クライアント TCP が TCP 接続でキープアライブメッセージを送信するまでに待機する間隔（秒単位）を指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
39	tcp-キープアライブゴミ	AT_BOOL	クライアントが、古い実装との互換性のために、TCP キープアライブメッセージをガベージのオクテットで送信するかどうかを指定します。RFC 2132 を参照してください。
40	nis-ドメイン	AT_NSTRING	クライアントの NIS ドメイン名を指定します。RFC 2132 を参照してください。
41	nis-サーバー	AT_IPADDR（1+）	クライアントが使用できる NIS サーバーを示す IP アドレスの一覧を指定します。RFC 2132 を参照してください。
54	ntp-servers	AT_IPADDR（1+）	クライアントが使用できる NTP サーバーを示す IP アドレスの一覧を指定します。RFC 2132 を参照してください。
43	ベンダーカプセル化オプション	AT_BLOB	RFC 2132 を参照してください。
44	netbios-name-servers	AT_IPADDR（1+）	優先順位の高い順にリスト表示される RFC 1001/1002 [19] [20] NBNS ネームサーバーのリストを指定します。RFC 2132 を参照してください。
45	ネットビオス-dd-サーバー	AT_IPADDR（1+）	優先順位の高い順にリスト表示される RFC 1001/1002 NBDD サーバーのリストを指定します。RFC 2132 を参照してください。
46	netbios-node-type	AT_RANGEBYTE	RFC 1001/1002 の説明に従って設定可能な NetBIOS over TCP/IP クライアントを設定できるようにします。RFC 2132 を参照してください。
47	ネットビオススコープ	AT_NSTRING	RFC 1001/1002 で指定されたクライアントの NETBIOS over TCP/IP スコープパラメータを指定します。RFC 2132 を参照してください。
48	フォントサーバー	AT_BLOB（1+）	クライアントで使用可能な X Window システム [21] フォントサーバーの一覧を指定します。RFC 2132 を参照してください。
49	x ディスプレイマネージャー	AT_BLOB（1+）	X Window システムのディスプレイマネージャを実行し、クライアントが使用できるシステムの IP アドレスの一覧を指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
50	dhcp-requested-address	AT_BLOB	クライアント要求 (DHCPDISCOVER) で使用され、クライアントが特定の IP アドレスを割り当てるように要求できるようにします。RFC 2132 を参照してください。
51	dhcp リース時間	AT_TIME	クライアント要求（DHCPDISCOVER または DHCPREQUEST）で使用すると、クライアントは IP アドレスのリース時間を要求できます。RFC 2132 を参照してください。
52	dhcp オプション過負荷	AT_OVERLOAD	DHCP の「sname」フィールドまたは「file」フィールドを使用して DHCP オプションを実行することにより、これらのフィールドがオーバーロードされていることを示すのに使用されます。RFC 2132 を参照してください。
53	dhcp-message-type	AT_MESSAGE	DHCP メッセージのタイプを伝送するために使用されます。RFC 2132 を参照してください。
54	DHCP サーバー識別子	AT_IPADDR	メッセージを DHCPOFFER および DHCP 要求で使用し、オプションで DHCPACK および DHCPNAK メッセージに含めることができます。RFC 2132 を参照してください。
55	dhcp パラメータ要求 - リスト	AT_INT8（0+）	DHCP クライアントが、指定した構成パラメータの値を要求するために使用します。RFC 2132 を参照してください。
72	dhcp メッセージ	AT_NSTRING	障害が発生した場合に DHCPNAK メッセージで DHCP クライアントにエラー メッセージを提供するために DHCP サーバーによって使用されます。RFC 2132 を参照してください。
57	メッセージサイズ	AT_SHORT	受け入れ可能な DHCP メッセージの最大長を指定します。RFC 2132 を参照してください。
58	dhcp 更新時間	AT_TIME	アドレス割り当てからクライアントが更新状態に移行するまでの時間間隔を指定します。RFC 2132 を参照してください。
59	dhcp 再バインド時間	AT_TIME	アドレス割り当てからクライアントが再バインディング状態に移行するまでの時間間隔を指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
60	dhcp-class-identifier	AT_NSTRING	DHCP クライアントが必要に応じて使用し、DHCP クライアントのベンダータイプと設定を識別することができます。RFC 2132 を参照してください。
61	dhcp-client-identifier	AT_BLOB	DHCP クライアントは、一意の ID を指定するために使用します。RFC 2132 を参照してください。
62	ネットウェアイップドメイン	AT_NSTRING	Netware/IP 製品で使用される NetWare/IP ドメイン名を伝送するために使用されます。RFC 2242 を参照してください。
63	ネットウェア情報	AT_BLOB	NetWare/IP ドメイン名を除く、すべての NetWare/IP 関連情報を伝送するために使用されます。RFC 2242 を参照してください。
64	nis+ドメイン	AT_NSTRING	クライアントの NIS ドメイン名 [17] を指定します。RFC 2132 を参照してください。
65	nis+サーバー	AT_IPADDR（1+）	クライアントが使用できる NIS+ サーバーを示す IP アドレスの一覧を指定します。RFC 2132 を参照してください。
66	tftp-server	AT_NSTRING	DHCP ヘッダーの sname フィールドが DHCP オプションに使用されている場合に、TFTP サーバーを識別するのに使用します。RFC 2132 を参照してください。
67	boot-file	AT_NSTRING	DHCP ヘッダーの file フィールドが DHCP オプションに使用されている場合に、ブートファイルを識別するのに使用します。RFC 2132 を参照してください。
68	モバイル-ip-ホームエージェント	AT_IPADDR（0+）	クライアントが使用できるモバイル IP ホームエージェントを示す IP アドレスの一覧を指定します。RFC 2132 を参照してください。
69	smtp-servers	AT_IPADDR（1+）	クライアントが使用できる SMTP サーバーの一覧を指定します。RFC 2132 を参照してください。
70	pop3-servers	AT_IPADDR（1+）	クライアントが使用できる POP3 の一覧を指定します。RFC 2132 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
71	nntp サーバー	AT_IPADDR（1+）	クライアントが使用できる NNTP の一覧を指定します。RFC 2132 を参照してください。
72	www サーバー	AT_IPADDR（1+）	クライアントが使用できる WWW の一覧を指定します。RFC 2132 を参照してください。
73	指サーバー	AT_IPADDR（1+）	クライアントが使用できる Finger の一覧を指定します。RFC 2132 を参照してください。
74	ircサーバー	AT_IPADDR（1+）	クライアントが使用できる IRC の一覧を指定します。RFC 2132 を参照してください。
75	ストリートトークサーバー	AT_IPADDR（1+）	クライアントが使用できる StreetTalk サーバーの一覧を指定します。RFC 2132 を参照してください。
76	ストリートトークディレクトリ - アシスタンスサーバー	AT_IPADDR（1+）	クライアントが使用できる STDA サーバーの一覧を指定します。RFC 2132 を参照してください。
77	dhcp-user-class-id	AT_TYPECNT	DHCP クライアントがユーザーまたはアプリケーションの種類やカテゴリをオプションで識別するのに使用します。RFC 3004 を参照してください。
78	slp-ディレクトリ-エージェント	AT_BLOB	1 つ以上の SLP ディレクトリエージェントの場所を指定します。RFC 2610 を参照してください。
79	slp サービス スコープ	AT_BLOB	SLP エージェントが使用するように設定されている範囲を示すカンマ区切りのリスト。RFC 2610 を参照してください。
80	rapid-commit	AT_ZEROSIZE	アドレス割り当てに 2 つのメッセージ交換を使うことを示すために使用します。RFC 4039 を参照してください。
81	client-fqdn	AT_BLOB	クライアント FQDN オプション。RFC 4702 を参照してください。
82	relay-agent-info	AT_BLOB	エージェントが提供する特定のサブオプション用の「コンテナ」オプション。サブオプションについては、表 77 : DHCPv4 およびブート・オプション （553 ページ）を参照してください。RFC 3046 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
83	iSNS	AT_BLOB	プライマリサーバー、バックアップ iSNS サーバー、および iSNS クライアントで使用可能な iSNS サービスの場所を指定します。RFC 4174 を参照してください。
85	nds-サーバー	AT_IPADDR（1+）	NDS データベースにアクセスするためにクライアントが接続する 1 つ以上の NDS サーバーを指定します。RFC 2241 を参照してください。
86	nds ツリー	AT_NSTRING	クライアントが接続する NDS ツリーの名前を指定します。RFC 2241 を参照してください。
87	nds コンテキスト	AT_NSTRING	クライアントが使用する 初期 NDS コンテキストを指定します。NDS コンテキストは 16 ビットの Unicode 文字列です。RFC2241 を参照してください。
88	bcmcs サーバー-d	AT_DNSNAME（1+）	DHCPv4 のブロードキャストおよび マルチキャスト ドメイン ネーム サービス リスト。RFC 4280 を参照してください。
89	bcmcs-サーバー-a	AT_IPADDR（1+）	DHCPv4 のブロードキャストおよび IPv4 アドレスオプション。RFC 4280 を参照してください。
90	認証	AT_BLOB	DHCP 認証オプションRFC 3118 を参照してください。
91	lq-client-last-transaction- time	AT_TIME	受信者は、クライアントの最新のアクセス時刻を確認できます。RFC 4388 を参照してください。
92	lq- 関連付け-ip	AT_IPADDR（1+）	特定の DHCPLEASEQUERY メッセージで指定された DHCP クライアントに関連付けられているすべての IP アドレスを返すのに使用されます。RFC 4388 を参照してください。
93	pxe クライアント アーチ	AT_SHORT	クライアント システム アーキテクチャの種類オプションの定義。RFC 4578 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
94	pxe クライアント ネットワーク ID	AT_BLOB	クライアント ネットワーク インターフェイス識別子オプションの定義。RFC 4578 を参照してください。
95	ldap-url	AT_NSTRING	LDAP サーバー。RFC 3679 を参照してください。
97	pxeクライアント-マシン ID	AT_BLOB	クライアントマシン識別子オプションの定義。RFC 4578 を参照してください。
98	ユーザー認証	AT_NSTRING	URL のリストを指定します。各 URL は、ユーザー認証プロトコル（UAP）でカプセル化された認証要求を処理できるユーザー認証サービスを指します。RFC 2485 を参照してください。
99	ジオコンフィシビック	AT_BLOB	DHCP シビック ロケーション オプション。RFC 4776 を参照してください。
100	ポシックスタイムゾーン	AT_NSTRING	IEEE 1003.1 TZ 文字列RFC 4833 を参照してください。
101	tzdb タイムゾーン	AT_NSTRING	TZ データベースへの参照。RFC 4833 を参照してください。
108	ipv6-only-preferred	AT_INT	IPv6 専用優先オプション。RFC 8925 を参照してください。
109	dhcp4o6-s46-saddr	AT_IP6ADDR	DHCP 4o6 ソフトワイヤ ソース アドレス オプション。RFC 8539 を参照してください。
112	ネットインフォ親サーバーアドイン	AT_IPADDR	Netinfo アドレス。RFC 3679 を参照してください。
113	ネットインフォ親サーバータグ	AT_NSTRING	Netinfo タグ。RFC 3679 を参照してください。
114	captive-portal	AT_NSTRING	DHCP キャプティブポータルオプション。RFC 8910 を参照してください。
116	auto-configure	AT_RANGEBYTE	ローカルサブネットで自動設定を無効にするかどうかを確認し、通知する場合に使用します。RFC 2563 を参照してください。
117	ネームサービス検索	AT_SHORT（1+）	ネームサービス検索オプション。RFC 2937 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
118	subnet-selection	AT_IPADDR	サブネット選択オプション。RFC 3011 を参照してください。
119	ドメイン検索	AT_DNSNAME（1+）	ドメイン検索オプション。RFC 3397 を参照してください。
120	sip-servers	AT_BLOB	SIP サーバー DHCP オプション。RFC 3361 を参照してください。
121	クラスレス静的ルート	AT_BLOB	クラスレスルートオプション。RFC 3442 を参照してください。
122	ケーブルラボ-クライアント- コンフィギュレーション	AT_BLOB	CableLabs クライアント設定オプション（表 77 : DHCPv4 およびブート・オプション （553 ページ）を参照してください）。RFC 3495 を参照してください。
123	ジオコンフェ	AT_BLOB	DHCPv4 ジオコンプオプション。RFC 6225 を参照してください。
124	v-i-ベンダークラス	AT_VENDOR_CLASS	ベンダー識別ベンダークラスオプション。RFC 3925 を参照してください。
125	v-i-vendor-opts	AT_VENDOR_OPTS	ベンダー識別のためのベンダー固有の情報オプション。表 77 : DHCPv4 およびブート・オプション （553 ページ） の cablelabs-125 サブオプションも参照してください。RFC 3925 を参照してください。
128	mcns-security-server	AT_IPADDR	DOCSIS「フルセキュリティ」サーバーの IP アドレス。RFC 4578 を参照してください。
136	パナエージェント	AT_IPADDR（1+）	パナ認証エージェント DHCPv4 オプション。RFC 5192 を参照してください。
137	失われたサーバー	AT_DNSNAME	LoST サーバー DHCPv4 オプション。RFC 5223 を参照してください。
138	capwap-ac-v4	AT_IPADDR（1+）	CAPWAP AC DHCPv4 オプション。RFC 5417 を参照してください。
139	モスアドレス	AT_BLOB（0+）	DHCPv4 の MoS IPv4 アドレスオプション。RFC 5678 を参照してください。
140	モスト fqdn	AT_BLOB（0+）	DHCPv4 の MoS ドメイン名リストオプション。RFC 5678 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
141	一ロウアcsドメイン	AT_DNSNAME（0+）	DHCP SIP ユーザーエージェント設定サービスドメインオプション。RFC 6011 を参照してください。
142	andsf-v4	AT_IPADDR	DHCPv4 の ANDSF IPv4 アドレスオプション。RFC 6153 を参照してください。
143	リダイレクト	AT_TYPECNT（0+）	ブートストラップ・サーバーに対して、さらに構成を試みるために接続可能な 1 つ以上の URI をクライアントにプロビジョニングするために使用されます。RFC 8572 を参照してください。
144	ジオロック	AT_BLOB	DHCPv4 GeoLoc オプション。RFC 6225 を参照してください。
145	力を更新-ノンス可能	AT_INT8 （1+）	Forcerenew Nonce プロトコル機能オプション。RFC 6704 を参照してください。
146	選択	AT_BLOB	DNS ルックアップの順方向または逆引きの手順を実行するときに RDNSS に連絡できるリゾルバに通知するために使用されます。RFC 6731 を参照してください。
147	dots-ri	AT_DNSNAME	DHCPv4 DOTS Reference 識別子オプション。RFC 8973 を参照してください。
148	dots-address	AT_IPADDR（1+）	DHCPv4 DOTS アドレスオプション。RFC 8973 を参照してください。
150	tftp-server-address	AT_IPADDR（1+）	TFTP サーバーアドレスオプションの定義。RFC 5859 を参照してください。
151	status-code	AT_BLOB	DHCPBULKLEASEQUERY 要求のステータスに関して、マシンで読み取り可能な値を返すことができます。RFC 6926 を参照してください。
152	ベースタイム	AT_DATE	DHCPv4 サーバーからバルクリースクエリのリクエスト送信者に送信されるメッセージが作成された現在の時刻。RFC 6926 を参照してください。
153	start-time-of-state	AT_TIME	受信者は、IP アドレスが現在の状態に遷移した時刻を判別できます。RFC 6926 を参照してください。

DHCP オプション

番号	Cisco Prime Network Registrar 名	検証	説明
154	クエリ開始時刻	AT_DATE	DHCPv4 サーバーに対するクエリの開始時刻を指定します。RFC 6926 を参照してください。
155	クエリ終了時刻	AT_DATE	DHCPv4 サーバーに対するクエリの終了時刻を指定します。RFC 6926 を参照してください。
156	状態	AT_INT8	DHCPLEASEACTIVE および DHCPLEASEUNASSIGNED メッセージタイプで許可されているよりも詳細な情報を返すことができます。RFC 6926 を参照してください。
157	data-source	AT_INT8	DHCPLEASEACTIVE または DHCPLEASEUNASSIGNED メッセージ内のデータソースに関する情報が含まれます。RFC 6926 を参照してください。
158	v4-pcp-server	AT_BLOB	PCP サーバーの IPv4 アドレスのリストを設定するために使用されます。RFC 7291 を参照してください。
159	v4-portparams	AT_BLOB	DHCPv4 ポートパラメータオプション。RFC 7618 を参照してください。
160	captive-portal-old	AT_NSTRING	キャプティブポータル DHCPv4 オプション。RFC 7710 を参照してください。
161	泥のURL	AT_NSTRING	IPv4 MUD URL クライアントオプション。RFC 8520 を参照してください。
162	Cisco クライアント要求ホスト名	AT_NSTRING	Cisco クライアントの要求ホスト名。RFC 3942 を参照してください。
163	シスコ クライアント-最後のトランザクション時間	AT_INT	Cisco クライアントの最終トランザクション時刻RFC 3942 を参照してください。
185	vpn-id	AT_BLOB	VPN 識別子。RFC 3942 を参照してください。
209	pxelinux-コンフィグファイル	AT_NSTRING	構成ファイルオプション。RFC 5071 を参照してください。
210	pxelinux パス接頭辞	AT_NSTRING	パスプレフィックスオプション。RFC 5071 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
211	pxelinux-リブート時間	AT_TIME	リブート時間オプション。RFC 5071 を参照してください。
212	6rd	AT_BLOB	第6 DHCPv4 オプション。RFC 5969 を参照してください。
213	access-domain	AT_NSTRING	アクセスネットワークのドメイン名 DHCPv4 オプションRFC 5986 を参照してください。
220	サブネット-アロク	AT_TIME	サブネット割り当てオプション。RFC 6656 を参照してください。
221	シスコ-VPN ID	AT_NSTRING	DHCPv4 仮想サブネット選択オプション。RFC 6607 を参照してください。
251	シスコ自動設定	AT_RANGEBYTE	Cisco 自動設定オプション
255	終了	AT_NOLEN	ベンダーフィールドの有効な情報の終わりを示します。RFC 2132 を参照してください。

Cisco Prime Network Registrar 名別 DHCPv4 オプション

次の表に、Cisco Prime Network Registrar 名ごとの DHCPv4 オプションを示します。オプションの検証の種類ごとに、番号で数値による DHCPv4 オプション （509 ページ）と相互参照し、[検証（Validation）] 列を確認します。

表 *71 : Cisco Prime Network Registrar* 名別 *DHCPv4* オプション

Cisco Prime Network Registrar 名	番号	オプション名
6rd	212	IPv4 インフラストラクチャでの IPv6 の迅速な展開（6rd）
access-domain	213	ネットワーク ドメイン名へのアクセス
すべてのサブネットローカル	27	すべてのサブネットがローカル
andsf-v4	142	DHCPv4 の場合の IPv4 アドレス
arp-cache-timeout	35	[ARP キャッシュ タイムアウト（ARP Cache Timeout）]
認証	90	認証
auto-configure	116	自動設定

Cisco Prime Network Registrar 名	番号	オプション名
ベースタイム	152	ベースタイム
bcmcs-サーバー-a	89	BCMCS Address
bcmcs サーバー-d	88	BCMCS Controller Domain
boot-file	67	起動ファイル名
ブートサイズ	13	起動ファイルサイズ
broadcast-address	36	ブロードキャスト アドレス
ケーブルラボ-クライアント構成	122	CableLabs クライアント設定
captive-portal	114	キャプティブポータル DHCPv4
captive-portal-old	160	キャプティブポータル DHCPv4
capwap-ac-v4	138	カプワップ AC
シスコ自動設定	251	Cisco 自動設定
シスコ クライアント-最終トランザクション時間	163	Cisco クライアント最終トランザクション時間
シスコ クライアント要求ホスト名	162	Cisco クライアント要求ホスト名
シスコ-VPN ID	221	Cisco VPN 識別子
クラスレス静的ルート	121	クラスレス静的ルート
client-fqdn	81	DHCP クライアント FQDN
クッキーサーバー	8	クッキーサーバー
data-source	157	data-source
デフォルト-ip-ttl	23	デフォルトの IP 存続時間
デフォルト-tcp-ttl	37	TCP デフォルト TTL
dhcp-class-identifier	60	ベンダー クラス ID
dhcp-client-identifier	61	Client-Identifier
dhcp リース時間	51	[IP アドレス リース時間（IP Address Lease Time）]
メッセージサイズ	57	最大 DHCP メッセージ サイズ
dhcp メッセージ	72	メッセージ

Cisco Prime Network Registrar 名	番号	オプション名
dhcp-message-type	53	DHCP メッセージ タイプ
dhcp オプション過負荷	52	オプション オーバーロード
dhcp-parameter-request-list	55	パラメータ要求リスト
dhcp 再バインド時間	59	Rebinding (T2) Time Value
dhcp 更新時間	58	更新 (T1) 時間値
dhcp-requested-address	50	Requested IP Address
DHCP サーバー識別子	54	Server Identifier
状態	156	IP アドレスの状態
dhcp-user-class-id	77	ユーザークラス ID
dhcp4o6-s46-saddr	109	DHCP 4o6 ソフトワイヤソースアドレス
domain-name	15	ドメイン名（Domain Name）
domain-name-servers	6	ドメイン ネーム サーバー
ドメイン検索	119	ドメイン検索
dots-address	148	DHCPv4 DOTS Address
dots-ri	147	DHCPv4 DOTS Reference 識別子
終了	255	終了（End）
拡張機能パス	18	拡張機能のパス
指サーバー	73	Finger サーバー
フォントサーバー	48	X ウィンドウ システム フォント サーバー
力を更新-ノンス可能	145	ノンス認証の強制更新
ジオコンフェ	123	ジオコンプ
ジオコンフィシビック	99	シビックアドレスの構成
ジオロック	144	不確実性を伴う地理空間の位置
host-name	12	ホスト名（Host Name）
ieee802.3-encapsulation	36	Ethernet Encapsulation
印象づけるサーバー	10	インプレスサーバー

Cisco Prime Network Registrar 名	番号	オプション名
interface-mtu	26	インターフェイス MTU
ip-forwarding	19	IP 転送の有効化/無効化
ipv6-only-preferred	108	IPv6 専用優先
ircサーバー	74	IRC サーバー
iSNS	83	iSNS
ldap-url	95	Lightweight Directory Access Protocol（LDAP）サーバー
log-servers	7	[ログ サーバー（Log Server）]
失われたサーバー	137	ロースト サーバー DHCPv4
lpr サーバー	9	LPR サーバー
lq 関連付け-ip	92	リースクエリ関連 IP アドレス
lq クライアント-最終トランザクション時間	91	リースクエリ クライアント トランザクション時間
マスクサプライヤー	30	マスクサプライヤー
マックス・ドグラム再構成	22	データグラムの最大リアセンブルサイズ
mcns-security-server	128	DOCSIS「フルセキュリティ」サーバーの IP アドレス
メリットダンプ	14	メリットダンプファイル
モバイル-ip-ホームエージェント	68	モバイル IP ホームエージェント
モスアドレス	139	MoS IPv4 アドレス
モスト fqdn	140	MoS ドメイン名リスト
泥のURL	161	IPv4 マッド URL
name-servers	5	[ネーム サーバー（Name Server）]
ネームサービス検索	117	ネームサービス検索
nds コンテキスト	87	NDS コンテキスト
nds-サーバー	85	NDS サーバー
nds ツリー	86	NDS ツリー名

Cisco Prime Network Registrar 名	番号	オプション名
ネットビオス-dd-サーバー	45	NetBIOS over TCP/IP データグラム配信サーバー
netbios-name-servers	44	NetBIOS over TCP/IP name server
netbios-node-type	46	NetBIOS over TCP/IP ノードタイプ
ネットビオススコープ	47	NetBIOS over TCP/IP Scope
ネットインフォ親サーバーアドイン	112	ネット情報親サーバー アドレス
ネットインフォ親サーバータグ	113	親サーバー タグ
ネットウェアイップドメイン	62	ネットウェア/IP ドメイン名
ネットウェア情報	63	ネットウェア/IP情報
nis+ドメイン	64	NIS+ ドメイン
nis+サーバー	65	ネットワーク インフォメーション サービス（NIS+）サービス
nis-ドメイン	40	NIS ドメイン
nis-サーバー	41	ネットワーク インフォメーション サービス（NIS）サービス
nntp サーバー	71	NNTP サーバー
非ローカル ソース ルーティング	20	非ローカル ソース ルーティング
ntp-servers	54	NTP サーバー（NTP Servers）
パッド	[0]	パッド
パナエージェント	136	パナ認証エージェント DHCPv4
パス-mtu エージング タイムアウト	24	パス MTU エージング タイムアウト
パス-mtu-プラトーテーブル	25	パス MTU 台台
マスク検出の実行	29	マスク検出の実行
ポリシー フィルター	21	ポリシーフィルタ
pop3-servers	70	POP3 サーバー
ポシックスタイムゾーン	100	IEEE 1003.1 文字列

Cisco Prime Network Registrar 名	番号	オプション名
pxe クライアント アーチ	93	クライアント システム アーキテクチャの種類
pxe クライアント-マシン ID	97	クライアントマシン識別子
pxe クライアント ネットワーク ID	94	クライアント ネットワーク インターフェイス識別子
pxelinux-コンフィグファイル	209	設定ファイル（Configuration File）
pxelinux パス接頭辞	210	パスプレフィックス
pxelinux-リブート時間	211	リブート時間
クエリ終了時刻	155	クエリ終了時刻
クエリ開始時刻	154	クエリ開始時刻
rapid-commit	80	迅速なコミット
選択	146	選択 DHCPv4
relay-agent-info	82	DHCP リレーエージェント情報
リソース ロケーション サーバー	11	リソースロケーションサーバー
root-path	17	ルート パス
ルーター発見	31	ルータ検出の実行
ルーター勧誘アドレス	32	ルータ要求アドレス
ルータ	3	ルータ
sip-servers	120	[SIPサーバー（SIP Servers）]
一ロウアcsドメイン	141	SIP UA コンフィグレーション サービス ドメイン
slp-ディレクトリ-エージェント	78	SLP ディレクトリ エージェント
slp サービス スコープ	79	SLP サービス スコープ
smtp-servers	69	SMTP サーバー（SMTP Server）
start-time-of-state	153	start-time-of-state
static-routes	33	Static Route
status-code	151	状態コード

Cisco Prime Network Registrar 名	番号	オプション名
ストリートトークディレクトリ-アシスタンスサーバー	76	STDA サーバー
ストリートトークサーバー	75	ストリートトークサーバー
サブネット-アロク	220	サブネット割り当て
subnet-mask	1	サブネット マスク（Subnet Mask）
subnet-selection	118	サブネット選択
スワップサーバー	16	スワップサーバー
リダイレクト	143	DHCPv4 SZTP リダイレクト
tcp-キープアライブゴミ	39	TCP キープアライブ ガベージ
tcp キープアライブ間隔	38	TCP キープアライブ間隔
tftp-server	66	TFTP Server Name
tftp-server-address	150	TFTP サーバー アドレス
time-offset	2	オフセット時間（Time Offset）
time-servers	4	Time Server
トレーラーカプセル化	34	Trailer Encapsulation
tzdb タイムゾーン	101	TZ データベース文字列
ユーザー認証	98	ユーザー認証
v-i-ベンダークラス	124	ベンダー識別ベンダー クラス
v-i-vendor-opts	125	ベンダー識別オプション
v4-pcp-server	158	DHCPv4 PCP サーバー
v4-portparams	159	DHCPv4 ポートパラメータ
vendor-encapsulated-options	43	ベンダー固有情報
vpn-id	185	VPN 識別子
www サーバー	72	WWWサーバー
x ディスプレイマネージャー	49	X ウィンドウシステムディスプレイマネージャ

番号順の DHCPv6 オプション一覧

次の表は、オプション番号でソートされた DHCPv6 オプションと、検証タイプを示しています。[検証] 列に表示されるオプションの検証の種類の詳細については、「表 74 : 検証タイプ（549ページ）」を参照してください。すべてのオプションパケットには、少なくともオプション長（option-len）と可変長データフィールドが含まれます。また、表に示すように、追加のパラメーター設定を使用することもできます。これらのオプションの多くは RFC 8415 で説明されています。

（注） RFC 8415 は、以前の RFC である RFC 3315、RFC 3633、RFC 3736、RFC 4242、および RFC 7083 を組み込み、廃止しました。

表 *72 :* 番号順の *DHCPv6* オプション一覧

番号	Cisco Prime Network Registrar 名	検証	説明
1	client-identifier	AT_BLOB	クライアントとサーバー間のクライアントを識別する DUID。RFC 8415 を参照してください。
2	server-identifier	AT_BLOB	クライアントとサーバーの間のサーバーを識別する DUID。RFC 8415 を参照してください。
3	ia-na	AT_BLOB	関連するパラメーターとアドレスを含む一時アドレス オプション。パラメータは、一意の ID と、クライアントが IA 内のアドレスにアクセスする時間と、クライアントが利用可能な任意のサーバーにアクセスする時間（どちらもアドレスの有効期間を拡張するため）です。RFC 8415 を参照してください。
4	ia-ta	AT_BLOB	関連するパラメーターとアドレスを含む一時アドレス オプション。RFC 8415 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
5	イアアドル	AT_BLOB	IA_NAまたはIA_TAに関連付けられたIPv6アドレス。(IAADRR は、IA_NA オプションまたはIA_TA オプションのオプション フィールドにカプセル化する必要があります。IAADDR オプションには、優先および有効な有効期間フィールド、およびこのアドレスに固有のオプションをカプセル化するオプション・フィールドが含まれます。RFC 8415 を参照してください。
6	oro	AT_SHORT（0+）	オプション要求オプション(ORO)は、クライアントとサーバーの間のメッセージ内のオプションのリストを識別します。クライアントは、要請、要求、更新、再バインド、確認、または情報要求メッセージにこのオプションを含め、クライアントがサーバーから必要とするオプションについてサーバーに通知することができます。サーバーは、クライアントが要求する必要があるオプションの更新を示す再設定メッセージにこのオプションを含めることができます。RFC 8415 を参照してください。
7	環境設定	AT_INT8	サーバーは、クライアントが選択するサーバーに影響を与えるために、このオプションをクライアントに送信します。RFC 8415 を参照してください。
8	elapsed-time	AT_SHORT	クライアントは、このオプションをサーバーに送信して、クライアントがメッセージ交換を完了しようとしている時間を示します。RFC 8415 を参照してください。
9	リレーメッセージ	AT_BLOB	リレー転送メッセージまたはリレー応答メッセージの DHCP メッセージ。RFC 8415 を参照してください。
11	auth	AT_BLOB	DHCP メッセージの ID と内容を認証します。パラメータは、認証プロトコル、認証アルゴリズム、再生検出方法(RDM)、および認証情報です。RFC 8415 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
12	サーバーユニキャスト	AT_IP6ADDR	サーバーは、このオプションをクライアントに送信して、クライアントがサーバーにメッセージをユニキャストできることを示します。RFC 8415 を参照してください。
13	status-code	AT_BLOB	DHCP メッセージまたは DHCP メッセージが表示されるオプションに関連する状態を示すメッセージを返します。パラメータは、ステータス コードとステータス メッセージです。RFC 8415 を参照してください。
18	rapid-commit	AT_ZEROSIZE	アドレス割り当てに 2 つのメッセージ交換を使用するシグナルです。RFC 8415 を参照してください。
15	ユーザー クラス	AT_TYPECNT	クライアントはこのオプションを使用して、それが表すユーザーまたはアプリケーションの種類またはカテゴリを識別します。ゼロ型カウント値フィールドの後にユーザーデータ (BLOB として) が続く。RFC 8415 を参照してください。
16	vendor-class	AT_VENDOR_CLASS	クライアントは、このオプションを使用して、クライアントが稼働しているハードウェアを製造したベンダーを識別します。RFC 8415 を参照してください。
17	vendor-opts	AT_VENDOR_OPTS	クライアントとサーバーは、このオプションを使用して、ベンダー固有の情報を交換します。CableLabs ベンダーのエンタープライズ ID は 4491 です。ケーブルラボのサブオプションはに記載表 78 : DHCPv6 のオプション（562 ページ）されています。RFC 8415 を参照してください。
18	interface-id	AT_BLOB	リレーエージェントはこのオプションを使用して、クライアントメッセージを受信するインターフェイスを識別します。RFC 8415 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
19	再設定-メッセージ	AT_INT8	サーバーは、クライアントが更新または情報要求メッセージで応答する必要があるかどうかを示す再構成メッセージに、これを含めます。RFC 8415 を参照してください。
20	再設定-受け入れる	AT_ZEROSIZE	クライアントはこのオプションを使用して、クライアントが再設定メッセージを受け入れるかどうかをサーバーに通知します。RFC 8415 を参照してください。
21	サイプサーバー名	AT_DNSNAME（0+）	クライアントの SIP アウトバウンドプロキシサーバーのドメイン名。RFC 3319 を参照してください。
22	サイプサーバーアドレス	AT_IP6ADDR（0+）	クライアントの SIP アウトバウンド・プロキシー・サーバーの IPv6 アドレス。RFC 3319 を参照してください。
23	dns-servers	AT_IP6ADDR（1+）	DNS 再帰ネームサーバーの IPv6 アドレス。RFC 3646 を参照してください。
24	domain-list	AT_DNSNAME（0+）	ドメイン検索リスト内のドメイン名。RFC 3646 を参照してください。
25	ia-pd	AT_BLOB	IPv6 プレフィックス委任 ID の関連付けと、関連するパラメーターとプレフィックス。パラメータは、一意の ID と、クライアントが IA 内のアドレスにアクセスする時間と、クライアントが利用可能な任意のサーバーにアクセスする時間（どちらもアドレスの有効期間を拡張するため）です。RFC 8415 を参照してください。
26	イアプレフィックス	AT_BLOB	IA_PDに関連付けられた IPv6 プレフィックス。プレフィックスは、IA_PD オプションのオプションフィールドにカプセル化する必要があります。パラメーターは、有効な有効期間と優先の有効期間、プレフィックス長、およびプレフィックスです。RFC 8415 を参照してください。
27	nis-サーバー	AT_IP6ADDR（1+）	クライアントで使用可能なネットワーク情報サービス (NIS) サーバーの IPv6 アドレスのリスト。RFC 3898 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
36	サーバーを使用する	AT_IP6ADDR（1+）	クライアントで使用できる NIS+ サーバーの IPv6 アドレスのリスト。RFC 3898 を参照してください。
29	nis-domain-name	AT_DNSNAME（1+）	NIS ドメイン名をクライアントに伝えます。RFC 3898 を参照してください。
30	nisp-domain-name	AT_DNSNAME（1+）	NIS+ ドメイン名をクライアントに伝えます。RFC 3898 を参照してください。
31	サーバー	AT_IP6ADDR（1+）	クライアントが使用できる簡易ネットワークタイム プロトコル (SNTP) サーバーの一覧。RFC 4075 を参照してください。
32	情報更新時間	AT_TIME	クライアントが DHCPv6 情報を更新するまで待機する時間の上限を設定します。RFC 8415 を参照してください。
33	bcmcs-サーバー-d	AT_DNSNAME（1+）	BCMCS コントローラ ドメインの一覧。RFC 4280 を参照してください。
34	bcmcs-サーバー-a	AT_IP6ADDR（1+）	ブロードキャストおよびマルチキャスト サービス (BCMCS) コントローラの IPv6 アドレスのリスト。RFC 4280 を参照してください。
36	ジオコンフィシビック	AT_BLOB	DHCP 市民アドレスの構成。RFC 4776 を参照してください。
37	remote-id	AT_BLOB	交換回線または恒久回線を終了するリレー エージェントは、このオプションを追加してリモート ホストを識別できます。RFC 4649 を参照してください。
38	relay-agent-subscriber-id	AT_BLOB	サブスクライバー固有のアクションの割り当てとアクティブ化を許可します。RFC 4580 を参照してください。
39	client-fqdn	AT_BLOB	DHCP クライアントの FQDN。RFC 4704 を参照してください。
40	パナエージェント	AT_IP6ADDR（1+）	32 ビット (バイナリ) IPv4 アドレスのリストを持ち、PANA クライアント (PaC) が使用できる PANA 認証エージェント (PAA) を示します。RFC 5192 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
41	新しいポシックス タイムゾーン	AT_NSTRING	POSIX 時間帯、たとえば EST5EDT4、M3.2.0/02:00、M11.1.0/02:00。RFC 4833 を参照してください。
54	新しい tzdb タイムゾーン	AT_NSTRING	POSIX タイムゾーンデータベース名(ヨーロッパ/チューリッヒなど)。RFC 4833 を参照してください。
43	エロ	AT_SHORT（0+）	リレーエージェントエコー要求オプションを使用して、エコーバックするリレーエージェントオプションの一覧をサーバーに通知します。RFC 4994 を参照してください。
44	lq クエリ	AT_BLOB	リースクエリ メッセージでのみ使用されます。は、実行されているクエリを識別します。このオプションには、クエリの種類、リンク アドレス (0::0)、およびクエリに必要なデータを提供するオプションが含まれます。RFC 5007 を参照してください。
45	client-data	AT_CONTAINER6	単一のクライアントのデータを、単一のリンク上の LEASEQUERY-REPLY メッセージにカプセル化します。RFC 5007 を参照してください。
46	clt-time	AT_TIME	クライアント データ オプションにカプセル化されたクライアントの最後のトランザクション時間。は、サーバーがクライアントと最後に通信した時間 (秒単位) を示します。RFC 5007 を参照してください。
47	lqリレーデータ	AT_BLOB	リースクエリ応答メッセージでのみ使用されます。は、クライアントが最後にサーバーと通信したときに使用されるリレーエージェント データを提供します。RFC 5007 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
48	lq クライアントリンク	AT_IP6ADDR（1+）	リースクエリ応答メッセージでのみ使用されます。は、クライアントが 1 つ以上のバインディングを持つリンクを識別します。リンクアドレスが指定されず、クライアントが複数のリンク上にあることが検出された場合、クエリに対する応答で使用されます。RFC 5007 を参照してください。
49	mip6-hnidf	AT_DNSNAME	ホーム ネットワーク ID の FQDN オプションを定義します。RFC 6610 を参照してください。
50	mip6-vdinf	AT_CONTAINER6	[訪問先のホーム ネットワーク情報] オプションを定義します。RFC 6610 を参照してください。
51	失われたサーバー	AT_DNSNAME	DHCPv6 クライアントは、Options Request Option（ORO）で LoST サーバードメイン名を要求します（RFC 8415 を参照）。 このオプションには単一のドメイン名が含まれ、正確に 1 つのルート ラベルを含める必要があります。RFC 5223 を参照してください。
52	capwap-ac-v6	AT_IP6ADDR（1+）	128 ビット(バイナリ)IPv6 アドレスのリストを持ち、ワイヤレス ターミネーション ポイント(WTP)で使用可能なワイヤレス アクセス ポイント(CAPWAP)アクセス コントローラ(AC)の 1 つまたは複数の制御およびプロビジョニングを示します。RFC 5417 を参照してください。
53	リレー ID	AT_BLOB	DHCPv6 サーバーは、リレー転送メッセージからリレー ID オプションを、その結果として処理されるプレフィックスの委任やリースバインディングに関連付けることができます。RFC 5460 を参照してください。
54	モスアドレス	AT_IP6ADDR	DHCP v4 のモビリティ セバー(MoS)IPv6 アドレス。RFC 5678 を参照してください。
55	モスト fqdn	AT_BLOB	DHCPv6 のモビリティ セバー(MoS)ドメイン名リスト。RFC 5678 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
56	ntp-server	AT_BLOB	1 台のネットワーク タイム プロトコル (NTP) サーバーまたは簡易ネットワーク タイム プロトコル (SNTP) サーバーに関連するサーバーの場所情報のコンテナーとして機能します。このオプションは、DHCPv6 メッセージに複数回表示される場合があります。このオプションの各インスタンスは、NTP クライアントまたは SNTP クライアントが構成に含めるサーバーとして考慮されます。 オプション自体には値が含まれていません。代わりに、NTP サーバーまたは SNTP サーバーの場所を伝送する 1 つまたは複数のサブオプションが含まれています。RFC 5908 を参照してください。
57	access-domain	AT_DNSNAME	アクセス ネットワークに関連付けられたドメイン名を定義します。このオプションには単一のドメイン名が含まれ、1 つのルート ラベルを含める必要があります。RFC 5986 を参照してください。
58	一ロウアcsドメイン	AT_DNSNAME（0+）	セッション開始プロトコル (SIP) ユーザー エージェント構成サービス ドメイン内のドメイン名の一覧を定義します。RFC 6011 を参照してください。
59	ブートファイル-URL	AT_NSTRING	ブート ファイルの URL についてクライアントに通知します。RFC 5970 を参照してください。
60	ブートファイルパラム	AT_TYPECNT（0+）	サーバーからクライアントに送信されます。ブートファイルのパラメータを指定するための複数のUTF-8(RFC 3629を参照)文字列で構成されています。RFC 5970 を参照してください。
61	クライアント・アーチ・タイプ	AT_SHORT（1+）	DHCPv4 に定義されたクライアント システム アーキテクチャ タイプ オプション (オプション 93) とのパリティを提供します。RFC 5970 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
62	nii	AT_BLOB	DHCPv4 に定義されたクライアント ネットワーク インターフェイス識別子オプション(オプション 94)とのパリティを提供します。RFC 5970 を参照してください。
63	ジオロック	AT_BLOB	サーバーによって提供されるクライアントの座標ベースの地理的位置を指定します。RFC 6225 を参照してください。
64	aftr-name	AT_DNSNAME	AFTR トンネルエンドポイントの完全修飾ドメイン名を定義します。RFC 6334 を参照してください。
65	erp-local-domain-name	AT_DNSNAME	ローカル ERP ドメインの名前が含まれます。RFC 6440 を参照してください。
66	ルスー	AT_CONTAINER6	リレー エージェントが DHCPv6 サーバーに提供するオプションをカプセル化します。RFC 6422 を参照してください。
67	pd除外	AT_BLOB	デリゲートされたプレフィックスから 1 つのプレフィックスを除外するために使用します。RFC 6603 を参照してください。
68	vpn-id	AT_BLOB	VPN を識別するために使用されます。RFC 6607 を参照してください。
69	mip6-idinf	AT_CONTAINER6	識別されたホーム ネットワークに関する情報を提供するために、リレーエージェントおよび DHCP サーバーによって使用されます。RFC 6610 を参照してください。
70	mip6-udinf	AT_CONTAINER6	DHCP サーバー管理者によって指定されたホーム ネットワークに関する情報を提供します。RFC 6610 を参照してください。
71	mip6-hnp	AT_BLOB	ホーム ネットワークのプレフィックスを定義します。RFC 6610 を参照してください。
72	mip6-haa	AT_IP6ADDR	DHCP サーバーおよびリレー エージェントがホーム エージェントの IP アドレスを指定するために使用します。RFC 6610 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
73	mip6-haf	AT_DNSNAME	ホーム エージェントの FQDN を指定して、必要に応じて、ホーム エージェントの IPv4 または IPv6 アドレスを含む 1 つまたは複数の A または AAAA レコードを検索します。RFC 6610 を参照してください。
74	選択	AT_BLOB	DNS 参照の順方向または逆引きの手順を実行するときに RDNSS に連絡できるリゾルバに通知します。RFC 6731 を参照してください。
75	krb プリンシパル名	AT_BLOB	クライアントが DHCPv6 サーバーに送信し、クライアントまたは Kerberos アプリケーション・サーバーの特定の構成パラメーターのセットを選択するためにクライアントを使用します。RFC 6784 を参照してください。
76	krb-レルム名	AT_NSTRING	クライアントがアクセスするレルムを DHCPv6 サーバーに指定します。RFC 6784 を参照してください。
77	krb-default-realm-name	AT_NSTRING	Kerberos システム (クライアントおよび Kerberos アプリケーション・サーバー) のデフォルト・レルム名を指定します。RFC 6784 を参照してください。
78	クルブ-ク dc	AT_BLOB	KDC に関する構成情報を提供します。RFC 6784 を参照してください。
79	client-linklayer-address	AT_BLOB	クライアント リンク レイヤアドレスを示します。RFC 6939 を参照してください。
80	link-address	AT_IP6ADDR	クライアントが存在するリンクをサーバーに示します。RFC 6977 を参照してください。
81	radius	AT_BLOB	DHCPv6 リレー エージェントと DHCPv6 サーバーの間で承認および識別情報を交換するメカニズムを提供します。RFC 7037 を参照してください。
82	ゾルマックス-rt	AT_TIME	sol-max-rt のデフォルト値をオーバーライドします。RFC 8415 を参照してください。
83	インフ-マックス-rt	AT_TIME	inf-max-rt のデフォルト値をオーバーライドします。RFC 8415 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
84	アドルセル	AT_BLOB	ポリシーテーブルと、その他の構成パラメーターを提供します。RFC 7078 を参照してください。
85	アドルセルテーブル	AT_BLOB	アドレス選択ポリシー テーブル オプションを提供します。RFC 7078 を参照してください。
86	v6-pcp-server	AT_IP6ADDR（1+）	PCP サーバーの IPv6 アドレスのリストを構成します。このオプションは単一インスタンスのみをサポートします。RFC 7291 を参照してください。
87	dhcpv4-msg	AT_BLOB（0+）	クライアントまたはサーバーによって送信される DHCPv4 メッセージを運びます。このようなメッセージは、IP ヘッダーまたは UDP ヘッダーを除外します。RFC 7341 を参照してください。
88	dhcp4-o-dhcp6-server	AT_IP6ADDR（0+）	クライアントが IPv4 構成を取得するために接続する必要がある DHCP 4o6 サーバーの IPv6 アドレスのリストを持ちます。RFC 7341 を参照してください。
89	s46-rule	AT_BLOB	基本マッピング ルール(BMR)と転送マッピング ルール(FMR)を伝達します。RFC 7598 を参照してください。
90	s46-br	AT_IP6ADDR	ボーダーリレーの IPv6 アドレスを伝えます。RFC 7598 を参照してください。
91	s46-dmr		デフォルト マッピング ルール(DMR)の値を伝達します。RFC 7598 を参照してください。
92	s46-v4v6bind	AT_BLOB	CE の完全または共有 IPv4 アドレスを指定します。IPv6 プレフィックス フィールドは、トンネル ソースに使用する正しいプレフィックスを識別するために CE によって使用されます。RFC 7598 を参照してください。
93	s46-portparams	AT_BLOB	CEに提供される可能性のあるオプションのポートセット情報を指定します。RFC 7598 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
94	s46-cont-mape	AT_CONTAINER6	指定したドメインのすべてのルールとオプションのポートパラメータをグループ化するために使用するコンテナを指定します (Softwire46 MAP-E ドメイン)。RFC 7598 を参照してください。
95	s46-cont-mapt	AT_CONTAINER6	指定したドメインのすべてのルールとオプションのポートパラメータをグループ化するために使用するコンテナを指定します (Softwire46 MAP-T ドメイン)。RFC 7598 を参照してください。
96	s46-cont-lw	AT_CONTAINER6	指定したドメインのすべてのルールとオプションのポートパラメータをグループ化するために使用するコンテナを指定します (Softwire46 Lightweight 4over6 ドメイン)。RFC 7598 を参照してください。
97	4rd	AT_CONTAINER6	4rd (IPv4 残留展開) の DHCPv6 オプションを示します。RFC 7600 を参照してください。
98	4rd-map-rule	AT_BLOB	4rd ドメインのマッピングルールパラメータを示します。RFC 7600 を参照してください。
99	4rd-non-map-rule	AT_BLOB	4rd ドメインの非マッピングルールパラメータを示します。RFC 7600 を参照してください。
100	lqベースタイム	AT_INT	要求者がアクティブクエリまたはバルク リースクエリ要求で同じことを要求した場合、DHCPv6 サーバーによってアクティブクエリまたはバルク リースクエリの要求者に送信されるメッセージが作成された現在時刻。RFC 7653 を参照してください。
101	lq-開始時間	AT_INT	DHCPv6 サーバーに対する照会開始時刻を指定します。RFC 7653 を参照してください。
102	lq終了時刻	AT_INT	DHCPv6 サーバーに対するクエリの終了時刻を指定します。RFC 7653 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
103	captive-portal	AT_NSTRING	クライアントにキャプティブポータルの背後に配置されていることを通知し、認証ページにアクセスするための URI を提供します。RFC 7710 を参照してください。
104	mpl パラメータ	AT_BLOB（0+）	DHCP サーバーによって管理されるネットワーク内の MPL ドメインの構成またはすべての MPL ドメイン(ワイルドカード)のデフォルト値を配布する手段を提供します。RFC 7774 を参照してください。
105	アニアット	AT_BLOB	クライアントがネットワークに接続するために使用するアクセステクノロジの種類を交換するために使用されます。RFC 7839 を参照してください。
106	ani-ネットワーク名	AT_NSTRING	モバイル ノードが接続されているアクセスネットワークの名前。RFC 7839 を参照してください。
107	アニ・アプ・ネーム	AT_NSTRING	モバイル ノードが接続されているアクセスポイントの名前 (物理デバイス名)。RFC 7839 を参照してください。
108	アニ・アプ・ブシド	AT_BLOB	モバイル ノードが接続されているアクセスポイントの 48 ビット基本 SSSID (BSSID)。RFC 7839 を参照してください。
109	ani演算子 ID	AT_BLOB	ネットワークバイト順でエンコードされた可変長のプライベート・エンタープライズ番号 (PEN)。RFC 7839 を参照してください。
110	アニ演算子レルム	AT_NSTRING	演算子の領域。RFC 7839 を参照してください。
111	s46-priority	AT_SHORT（1+）	IPv4 サービス継続性メカニズムの優先順位を示します。RFC 8026 を参照してください。
112	泥のURL	AT_NSTRING	既存のツールセットでポリシーを簡単に見つけることができるように、ネットワークに対する Thing の種類を構造化された方法で識別します。RFC 8520 を参照してください。

番号	Cisco Prime Network Registrar 名	検証	説明
113	prefix64	AT_BLOB	IPv4 埋め込み IPv6 アドレスを合成するために使用される IPv6 プレフィックス (例えば、mB4) を伝えます。RFC 8115 を参照してください。
135	リレーポート	AT_SHORT	DHCPv6 用のリレー送信元ポート オプション。これは、特殊なサーバー処理を必要とし、「構成可能」オプションではありません - オプションは、リレーエージェントによって追加され、Relay-Replyでサーバーによってエコーされ、リレーを介してリレーに応答を返すためにサーバー内の特殊な処理が必要です。リレーパケットの送信元ポート。RFC 8357 を参照してください。
136	リダイレクト	AT_TYPECNT（0+）（URI 文字列のリスト）	ブートストラップ・サーバーに対して、さらに構成を試みるために接続可能な 1 つ以上の URI をクライアントにプロビジョニングするために使用されます。RFC 8572 を参照してください。
137	s46-bind-ipv6-prefix	AT_VPREFIX	DHCPv6 ソフトワイヤー ソース バインディング プレフィックス ヒント オプション。RFC 8539 を参照してください。
141	dots-ri	AT_DNSNAME	DHCPv6 DOTS Reference 識別子オプション。RFC 8973 を参照してください。
142	dots-address	AT_IP6ADDR（1+）	DHCPv6 DOTS Address オプション。RFC 8973 を参照してください。
143	ipv6-address-andsf	AT_IP6ADDR（1+）	モバイル・ノード (MN) が ANDSF サーバーを検索できるようにします。RFC 6153 を参照してください。

Cisco Prime Network Registrar 名別 DHCPv6 オプション

次の表に、Cisco Prime Network Registrar 名ごとの DHCP オプションを示します。オプションの検証の種類ごとに、番号で番号順の DHCPv6 オプション一覧 （530 ページ）と相互参照し、[検証（Validation）] 列を確認します。

DHCP オプション

表 *73 : Cisco Prime Network Registrar* 名別 *DHCPv6* オプション

Cisco Prime Network Registrar 名	番号	オプション名
4rd	97	IPv6 経由の IPv4 残留展開 (4 番目)
4rd-map-rule	98	第4マップルール
4rd-non-map-rule	99	第4回非地図ルール
access-domain	57	ネットワーク ドメイン名へのアクセス
アドルセル	84	アドレスの選択
アドルセルテーブル	85	アドレス選択ポリシー テーブル
aftr-name	64	AFTR トンネル エンドポイント ドメイン名
アニ・アプ・ブシド	108	DHCPv6 アクセス ポイント-BSSID
アニ・アプ・ネーム	107	DHCPv6 アクセスポイント名
アニアット	105	DHCPv6 アクセステクノロジータイプ
ani-ネットワーク名	106	DHCPv6 ネットワーク名
ani演算子 ID	109	DHCPv6 オペレーター ID
アニ演算子レルム	110	DHCPv6 オペレータ レルム
auth	11	認証
bcmcs-サーバー-a	34	BCMCS アドレス v6
bcmcs-サーバー-d	33	BCMCS コントローラ ドメイン v6
ブートファイルパラム	60	起動ファイルパラメータ
ブートファイル-URL	59	起動ファイルの Uniform Resource Locator（URL）
captive-portal	103	キャプティブポータル DHCPv6
capwap-ac-v6	52	カプワップ AC
クライアント・アーチ・タイプ	61	クライアント システム アーキテクチャの種類
client-data	45	リースクエリ応答クライアント データ
client-fqdn	39	DHCP クライアント FQDN

Cisco Prime Network Registrar 名	番号	オプション名
client-identifier	1	Client Identifier
クライアント リンクレイヤー アドレス	79	DHCPv6 クライアントリンク層アドレス
clt-time	46	リースクエリ クライアントの最後のトランザクション時刻
dhcp4-o-dhcp6-server	88	DHCP 4o6 サーバーアドレス
dhcpv4-msg	87	DHCPv4 メッセージ
dns-servers	23	DNS 再帰ネームサーバー
domain-list	24	ドメイン検索リスト（Domain Search List）
dots-address	142	DHCPv6 DOTS Address
dots-ri	141	DHCPv6 DOTS Reference Identifier
elapsed-time	8	経過時間（Elapsed Time）
エロ	43	リレー エージェントエコー要求オプション
erp-ローカル・ドメイン名	65	ローカル ERP ドメイン名
ジオコンフィシビック	36	シビックアドレスの構成
ジオロック	63	位置情報（GeoLocation）
ia-na	3	非一時アドレスの ID アソシエーション
ia-pd	25	プレフィックス委任
ia-ta	4	一時アドレスの ID アソシエーション
イアアドル	5	IA アドレス
イアプレフィックス	26	IA プレフィックス
インフ-マックス-rt	83	最大情報要求タイムアウト
情報更新時間	32	情報更新時間
interface-id	18	インターフェイス ID
ipv6-address-andsf	143	アンドスフ IPv6 アドレス
krb-デフォルト領域名	77	ケルベロスレルム名
クルブ-クdc	78	ケルベロス KDC

Cisco Prime Network Registrar 名	番号	オプション名
krb プリンシパル名	75	ケルベロスプリンシパル名
krb-レルム名	76	Kerberos Realm Name
link-address	80	リンクアドレス
失われたサーバー	51	ロケーションからサービスへの変換(LoST)サーバー DHCPv6
lqベースタイム	100	リースクエリベースタイム
lq クライアントリンク	48	リースクエリ クライアント リンク応答
lq終了時刻	102	リースクエリ終了時刻
lq クエリ	44	リースクエリ
lqリレーデータ	47	リースクエリ リレー エージェントの応答
lq-開始時間	101	リースクエリの開始時刻
mip6-haa	72	MIPv6 ホームエージェントアドレス
mip6-haf	73	MIPv6 ホーム エージェント FQDN
mip6-hnidf	49	MIPv6 ホーム ネットワーク ID FQDN
mip6-hnp	71	MIPv6 ホーム ネットワーク プレフィックス
mip6-idinf	69	MIPv6 によって識別されたホーム ネットワーク情報
mip6-udinf	70	MIPv6 無制限ホーム ネットワーク情報
mip6-vdinf	50	MIPv6 訪問ホームネットワーク情報
モスアドレス	54	MoS IPv6 アドレス
モスト fqdn	55	MoS ドメイン名リスト
mpl パラメータ	104	MPL パラメーター
泥のURL	112	IPv6 マッド URL
新しいポシックスタイムゾーン	41	POSIX タイムゾーン文字列
新しい tzdb タイムゾーン	54	POSIX タイム ゾーン データベース名

Cisco Prime Network Registrar 名	番号	オプション名
nii	62	クライアント ネットワーク インターフェイス識別子
nis-domain-name	29	NIS ドメイン名
nis-サーバー	27	NIS サーバー
nisp-domain-name	30	NIS+ ドメイン名
サーバーを使用する	36	NIS+ サーバー
ntp-server	72	メッセージ
oro	6	オプションリクエストオプション
パナエージェント	40	パナ認証エージェント DHCPv6
pd除外	67	プレフィックスの除外
環境設定	7	設定
prefix64	113	Prefix64
radius	81	DHCPv6 の半径
rapid-commit	14	迅速なコミット
選択	74	選択 DHCPv6
再設定-受け入れる	20	再設定の承認
再設定-メッセージ	19	再設定メッセージ
リレーエージェントサブスクライバーID	38	リレーエージェントのサブスクライバID
リレー ID	53	リレー ID
リレーメッセージ	9	リレー メッセージ
リレーポート	135	リレー送信元ポート
remote-id	37	リレー エージェントのリモート ID
ルスー	66	リレー提供オプション
s46-br	90	ソフトワイヤー46(S46)ボーダーリレー(BR)
s46-cont-lw	96	S46軽量4オーバー6コンテナ
s46-cont-mape	94	S46 MAP-E コンテナ

Cisco Prime Network Registrar 名	番号	オプション名
s46-cont-mapt	95	S46 MAP-T コンテナ
s46-dmr	91	S46 デフォルト マッピング ルール (DMR)
s46-portparams	93	S46 ポートパラメータ
s46-priority	111	S46 優先順位
s46-rule	89	S46 ルール
s46-v4v6bind	92	S46 IPv4/IPv6 アドレス バインディング
s46- bind-ipv6-prefix	137	DHCPv6 ソフトワイヤー ソース バインディング プレフィックス ヒント
server-identifier	2	DHCPv6 サーバー識別子
サーバー ユニキャスト	12	サーバーユニキャスト
サイプサーバーアドレス	22	SIP サーバー IPv6 アドレス一覧
サイプサーバー名	21	SIP サーバーのドメイン名リスト
一ロウアcsドメイン	58	SIP ユーザー エージェント構成サービス ドメイン
サーバー	31	SNTP 設定
ゾルマックス-rt	82	SOL_MAX_RT
status-code	13	状態コード
リダイレクト	136	DHCPv6 SZTP リダイレクト
ユーザー クラス	15	User クラス
v6-pcp-server	86	DHCPv6 PCP サーバー
vendor-class	16	ベンダークラス
vendor-opts	17	ベンダー固有情報
vpn-id	68	VPN 識別子

オプションの検証タイプ

次の表に、DHCP オプションの検証の種類を示します。カスタム オプションを定義するために、それらの一部を使用できないことに注意してください。

表 *74* : 検証タイプ

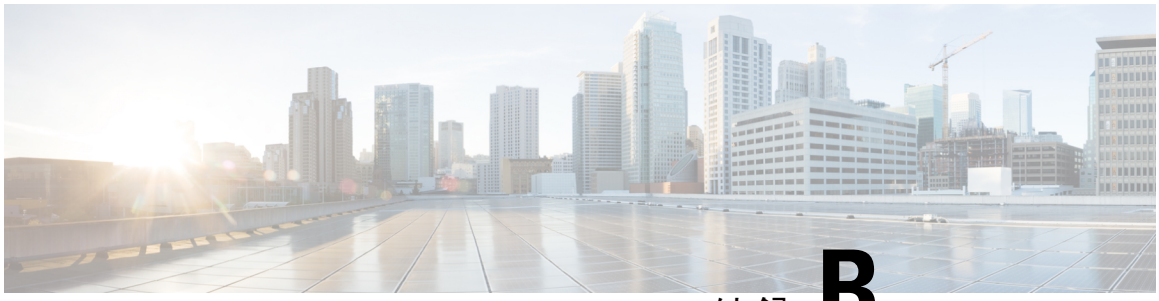
検証	説明 - Web UI の同等
AT_BLOB	バイナリバイトの一覧-バイナリ
AT_BOOL	ブール値 - ブール値
AT_CONTAINER6	DHCPv6 オプション コンテナ (カスタム オプションには使用できません)
AT_DATE	日付を表すバイト数 (日付)
AT_DNSNAME	DNS 名：DNS 名
AT_INT	符号なし 32 ビット整数 -符号なし 32 ビット
AT_INT8	8 ビット整数：符号なし 8 ビット
AT_INTI	符号なし 32 ビット整数 (Intel) - 符号なし 32 ビット (インテル)
AT_IPADDR	32 ビット IP アドレス:IP アドレス
AT_IP6ADDR	128 ビット IPv6 アドレス— IPv6 アドレス
AT_MACADDR	MAC アドレスを表すバイト—MAC アドレス
AT_MESSAGE	符号なし 8 ビット メッセージ (カスタム オプションでは使用できません)
AT_NOLEN	長さなし(PAD と END の場合のみ)
AT_NSTRING	ASCII 文字のシーケンス - 文字列
AT_OVERLOAD	オーバーロードバイト (カスタム オプションには使用できません)
AT_RANGEBYTE	バイトの範囲 (カスタム オプションには使用できません)
AT_RANGESHORT	ショートの範囲(カスタムオプションでは使用できません)
AT_RDNSNAME	相対 DNS 名 — 相対 DNS 名
AT_SHORT	符号なし 16 ビット整数 -符号なし 16 ビット
AT_SHRTI	符号なし 16 ビット整数 (Intel) - 符号なし 16 ビット (インテル)
AT_SINT	符号付き 32 ビット整数- 符号付き 32 ビット
AT_SINT8	8 ビット整数 - 符号付き 8 ビット
AT_SINTI	符号付き 32 ビット整数 (Intel) - 符号付き 32 ビット (インテル)
AT_SSHORT	符号付き 16 ビット整数 - 符号付き 16 ビット

検証	説明 - Web UI の同等
AT_SSHRTI	符号付き 16 ビット整数 (インテル) - 符号付き 16 ビット (インテル)
AT_STIME	時間を表す符号付き 32 ビット符号付き整数 - 符号付き時間
AT_STRING	ASCII 文字の無制限シーケンス -文字列
AT_TIME	時間を表す符号なし 32 ビット整数：符号なしの時間

AT_TYPECNT	2 つの子定義を必要とする型: 型フィールドのサイズとデータの型 - カウント型:
	DHCPv4 dhcp-user-class-id オプション (77) の場合、繰り返しパターンは次のようになります。
	`[ len (1 byte) ] [ data, of single type ]`
	DHCPv6 ユーザー・クラス・オプション (15) の場合、繰り返しパターンは次のようになります。
	`[ len (2 byte) ] [ data, of single type ]`

AT_VENDOR_CLASS	ベンダー クラス オプション (エンタープライズ ID の後に不透明データが続く、DHCPv4 の場合はエンタープライズ ID の後に EID 長さ) - ベンダー クラス
AT_VENDOR_OPTS	ベンダー固有のオプション・データ (エンタープライズ ID にベンダー固有データの TLV が続くもの、 DHCPv4 の場合はエンタープライズ ID の後に EID 長さ) - ベンダーが選択
AT_VPREFIX	IPv6 可変長プレフィックス
AT_ZEROSIZE	32 ビットのゼロ・サイズ (PAD および END では使用されなくなった)

（注） AT_TIMEは、デフォルトで秒単位で入力された値を受け取ります。たとえば、60 と入力すると、60秒と見なされます。60s、60m、2h、3d、4w、または1y と入力すると、それぞれ60秒、60 分、2 時間、2 日、4 週間、または 1 年と見なされ、60s、60m、2h、2d、4w、または 1y と表示されます。10m30s などの値を使用することもできます。この場合、値は 630 秒になり、10m30s と表示されます。



付録 B

DHCP 拡張ディクショナリ

この付録では、DHCP 拡張ディクショナリエントリと、拡張ディクショナリへのアプリケーション プログラム インターフェイス (API) について説明します。このクラスは、要求ディクショナリと応答ディクショナリで使用できるデータ項目、および Tcl 拡張機能および共有ライブラリから辞書にアクセスするときに使用する呼び出しについて説明します。

この付録の構成は、次のとおりです。

拡張ディクショナリ エントリ

ディクショナリは、キーと値のペアを含むデータ構造です。ディクショナリには、要求ディクショナリと応答ディクショナリで使用する属性ディクショナリと環境ディクショナリの2種類があります。このセクションでは、要求ディクショナリと応答ディクショナリについて説明します。環境辞書のエントリについては、TCL 環境ディクショナリ メソッド （599 ページ）で説明します。

復号化された DHCP パケット データ項目

デコードされた DHCPv4 パケット データ項目は、DHCP パケットの情報を表し、要求ディクショナリと応答ディクショナリの両方で使用できます。これらのディクショナリは、デコードされた要求とデコードされた応答よりもかなり多くの内部サーバーデータ構造にアクセスできます。

アスタリスク ()*が付いたすべてのオプションは複数であり、各オプションに複数の値が関連付けられている可能性があります。DHCP/BOOTP パケットでは、これらのデータ項目はすべて同じオプションに表示されます。ただし、拡張インターフェイスでは、インデックスを使用してこれらの複数のデータ項目にアクセスできます。

DHCP 拡張ディクショナリ

名前を持たないオプションには、オプション–nを指定します。表 77 : DHCPv4 およびブート・オプション （553 ページ）すべてのフィールドは読み取り/書き込み可能です。表 75 : DHCPv4 およびブート・フィールド （552 ページ） DHCPv4 パケットのフィールド値を記述します。表 76 : DHCPv6 フィールド （552 ページ）は DHCPv6 メッセージのフィールド値を記述しています。

表 *75 : DHCPv4* およびブート・フィールド

名前	値
chaddr	blob (バイトのシーケンス)
ciaddr	IP アドレス
file	文字列
Flags	16 ビットの符号なし整数
giaddr	IP アドレス
hlen	8 ビットの符号なし整数
hops	8 ビットの符号なし整数
htype	8 ビットの符号なし整数
op	8 ビットの符号なし整数
secs	16 ビットの符号なし整数
siaddr	IP アドレス
sname	文字列
xid	32 ビットの符号なし整数
yiaddr	IP アドレス

表 *76 : DHCPv6* フィールド

名前	値
hop-count	8 ビットの符号なし整数
link-address	[IPv6 アドレス（IPv6 address）]
msg-type	8 ビットの符号なし整数
peer-address	[IPv6 アドレス（IPv6 address）]
xid	32 ビットの符号なし整数

次の表に、DHCPv4 の DHCP オプションと BOOTP オプションを示します。

表 *77 : DHCPv4* およびブート・オプション

名前（* = 複数値）	ケース	値
6rd	212	binary
access-domain	213	DNS name
すべてのサブネットローカル	27	バイト値ブール型
andsf-v4	142	IP アドレス
arp-cache-timeout	35	符号なし時間
認証	90	blob(バイトのシーケンス)。5フィールド
auto-configure	116	8 ビットの符号なし整数
ベースタイム	152	date
bcmcs サーバー-a*	89	IP アドレス
bcmcs サーバー-d*	88	DNS name
boot-file	67	文字列
ブートサイズ	13	16 ビットの符号なし整数
broadcast-address	36	IP アドレス
ケーブルラボ-125(v-i-ベンダー情報 ID: 4491)	125 サブオプション：	binary
oro	1	オプション要求、8 ビット符号なし整数 (8 ビット符号なし整数)
tftp-servers	2	TFTP サーバーの IP アドレス
eルーターコンテナ	3	Erouter コンテナ オプション (バイナリ;TLV エンコード オプション)
パケットケーブルミブ・エンブ	4	MIB 環境インジケーター (8 ビット列挙)
モデム機能	5	モデム機能エンコーディング(バイナリ;TLV5 エンコードデータ)
acs-server	6	ACS サーバー サブオプション(バイナリ)

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
radius-server	7	RADIUS サーバー サブオプション (バイナリ)
dhcpv6-servers	123	DHCPv6 サーバーサブオプション (バイナリー)
IP-プレフ	124	IPv4 または IPv6 の基本設定 (8 ビット列挙)
ケーブルラボ-クライアント-コンフィギュレーション	122 サブオプション：	BLOB（バイト シーケンス）
primary-dhcp- server	1	IP アドレス（IP Address）
secondary-dhcp- server	2	IP アドレス
provisioning- server	3	BLOB (最初のバイトはタイプバイトで、RFC 1035 エンコーディングでは 0、IP アドレスエンコーディングの場合は1で、アドレスはネットワーク順にする必要があります)
バックオフ再試行 - BLOB	4	12 バイトの BLOB (3 つの符号なし 4 バイト整数、ネットワーク順にする必要があります)。Kerberos AS-REQ/AS-REP タイムアウト、バックオフ、および再試行メカニズムを設定します
ap-backoff-再試行- BLOB	5	12 バイトの BLOB (3 つの符号なし 4 バイト整数、ネットワーク順にする必要があります)。Kerberos AP-REQ/AP-REP タイムアウト、バックオフ、および再試行メカニズムを設定します。
kerberos-realm	6	可変長ブロブ(RFC 1035スタイル名)。Kerberos 領域名が必要です
使用-tgt	7	1 バイトの符号なし整数ブール値。アプリケーション サーバーの 1 つのサービス チケットを取得するときに、チケット保証チケット (TGT) を使用するかどうかを示します

名前（* = 複数値）	ケース	値
provisioning-timer	8	1 バイトの符号なし整数。プロビジョニングプロセスの完了に必要な最大時間を定義します。
チケットコントロール-マスク	9	ホスト順に 2 バイトの符号なし整数
kdc アドレス - ブロブ	10	可変長 (4 の倍数) IP アドレス (ネットワーク順)
captive-portal	114	文字列
captive-portal-old	160	文字列
capwap-ac-v4*	138	IP アドレス
シスコ自動設定	251	境界バイト
シスコ クライアント-最終トランザクション時間	163	32 ビットの符号なし整数
シスコ クライアント要求ホスト名	162	文字列
シスコ-VPN ID	221	ブロブ (構造化)
クラスレス静的ルート	121	BLOB（構造化）
client-fqdn	81	blob (バイトのシーケンス)。4 フィールド: フラグ、rcode-1、rcode-2、およびドメイン名
クッキーサーバー*	8	IP アドレス
data-source	157	8 ビットの符号なし整数
デフォルト-ip-ttl	23	8 ビットの符号なし整数
デフォルト-tcp-ttl	37	8 ビットの符号なし整数
dhcp4o6-s46-saddr	109	[IPv6 アドレス（IPv6 address）]
dhcp-class-identifier	60	文字列
dhcp-client-identifier	61	BLOB（バイト シーケンス）
dhcp リース時間	51	符号なし時間
メッセージサイズ	57	16 ビットの符号なし整数

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
dhcp メッセージ	72	文字列
dhcp-message-type	53	8 ビットの符号なし整数
dhcp オプション過負荷	52	8 ビットの符号なし整数
dhcp-parameter-request-list*	55	8 ビットの符号なし整数
dhcp パラメーター要求 - リスト BLOB*	55	BLOB（バイト シーケンス）
dhcp 再バインド時間	59	符号なし時間
dhcp 更新時間	58	符号なし時間
dhcp-requested-address	50	IP アドレス
DHCP サーバー識別子	54	IP アドレス
状態	156	8 ビットの符号なし整数
dhcp-user-class-id	77	カウントされた len バイト配列のセット。2 つのフィールド: サイズとユーザー データ
domain-name	15	文字列
domain-name-servers*	6	IP アドレス
ドメイン検索	119	BLOB（バイト シーケンス）
dot-address*	148	IP アドレス
dots-ri	147	DNS name
終了	255	長さなし
拡張機能パス	18	文字列
指サーバー*	73	IP アドレス
フォントサーバー*	48	IP アドレス
力を更新可能*	145	8 ビットの符号なし整数
ジオコンフェ	123	BLOB（バイト シーケンス）
ジオコンフィシビック	99	BLOB（バイト シーケンス）
ジオロック	144	binary

名前（* = 複数値）	ケース	値
host-name	12	文字列
ieee802.3-encapsulation	36	バイト値ブール値
印象づけるサーバー*	10	IP アドレス
interface-mtu	26	16 ビットの符号なし整数
ip-forwarding	19	バイト値ブール型
ipv6-only-preferred	108	32 ビットの符号なし整数
ircサーバー*	74	IP アドレス
iSNS	83	blob(バイトのシーケンス)。7フィールド
ldap-url	95	文字列
log-servers*	7	IP アドレス
失われたサーバー	137	DNS 名 (RFC 5223 を参照)
lpr サーバー*	9	IP アドレス
lq 関連付け IP*	92	IP アドレス
lq クライアント最終トランザクション時間	91	符号なし時間
マスクサプライヤー	30	バイト値ブール型
マックス・ドグラム再構成	22	16 ビットの符号なし整数
mcns-security-server	128	IP アドレス
メリットダンプ	14	文字列
モバイルip-ホームエージェント*	68	IP アドレス
モスアドレス	139 サブオプション：	バイナリ;3 サブオプション
は	1	IP アドレス（IP Address）
cs	2	IP アドレス
es	3	IP アドレス

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
モスト fqdn	140 サブオプション：	バイナリ;3 サブオプション
は	1	DNS name
cs	2	DNS name
es	3	DNS name
泥のURL	161	文字列
name-servers*	5	IP アドレス
ネームサービス検索*	117	16 ビットの符号なし整数
nds コンテキスト	87	文字列
nds-サーバー*	85	IP アドレス
nds ツリー	86	文字列
ネットビオス-ddサーバー*	45	IP アドレス
netbios-name-servers*	44	IP アドレス
netbios-node-type	46	8 ビットの符号なし整数
ネットビオススコープ	47	文字列
ネットインフォ親サーバーアドイン	112	IP アドレス
ネットインフォ親サーバータグ	113	文字列
ネットウェアイップドメイン	62	文字列
ネットウェア情報	63	BLOB（バイト シーケンス）
nis+ドメイン	64	文字列
nis+サーバー*	65	IP アドレス
nis-ドメイン	40	文字列
nis-サーバー*	41	IP アドレス
nntp サーバー*	71	IP アドレス

名前（* = 複数値）	ケース	値
非ローカル ソース ルーティング	20	バイト値ブール型
ntp-servers*	54	IP アドレス
パッド	[0]	長さなし
パナエージェント	136	IP アドレス (RFC 5192 を参照)
パス-mtu エージング タイムアウト	24	符号なし時間
パス-mtu-プラトーテーブル*	25	16 ビットの符号なし整数
マスク検出の実行	29	バイト値ブール型
ポリシー フィルター*	21	IP アドレス (2 つのポリシー フィルタがあり、それぞれに独自の IP アドレスを持つことができます)
pop3-servers*	70	IP アドレス
ポシックスタイムゾーン	100	文字列 (RFC 4833 を参照)
pxe クライアント アーチ	93	16 ビットの符号なし整数
pxe クライアント-マシン ID	97	blob (バイトのシーケンス)。2 フィールド: タイプフラグと uuid
pxe クライアント ネットワーク ID	94	blob (バイトのシーケンス)。2 フィールド: タイプフラグとバージョン
pxelinux-コンフィグファイル	209	文字列
pxelinux パス接頭辞	210	文字列
pxelinux-リブート時間	211	符号なし時間
クエリ終了時刻	155	date
クエリ開始時刻	154	date
rapid-commit	80	ヌル長さ
選択	146	バイナリ;4つのフィールド:予約済み、プライマリ再帰的な名前サーバー、セカンダリ再帰的な名前サーバー、およびドメインとネットワーク

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
リレーエージェント情報サブオプション:	82 サブオプション：	BLOB（バイト シーケンス）
relay-agent-circuit-id- data	1	BLOB（バイト シーケンス）
relay-agent-remote-id- data	2	BLOB（バイト シーケンス）
リレーエージェント-デバイス-クラスデータ	4	4 バイト符号なし整数
リレーエージェント-サブネット- 選択データ	5	IP アドレス
subscriber-id	6	ネットワーク クライアントまたはサブスクライバを識別する文字列
radius-attributes	7	サポートされる属性は、ユーザー、クラス、およびフレームプールです。
認証	8	binary
v-i-vendor-opts	9	ベンダー オプション
シスコサブネット選択	150	IP アドレス
シスコ-VPN ID	151	binary
シスコ サーバー ID オーバーライド	152	IP アドレス
（注） サブオプション・データの前に 2 バイト (サブオプション・コードとデータ長) を戻したリレー・エージェント回線 ID、リレー・エージェント・リモート ID、およびリレー・エージェント・デバイス・クラス・サブオプションは非推奨ですが、まだ使用可能です。		
リソースロケーションサーバー*	11	IP アドレス
root-path	17	文字列
ルーター発見	31	バイト値ブール型
ルーター勧誘アドレス	32	IP アドレス
routers*	3	IP アドレス
sip-servers	120	blob (バイトのシーケンス)。2 フィールド: フラグと sip サーバー リスト

名前（* = 複数値）	ケース	値
一ロウアcsドメイン	141	DNS name
slp-ディレクトリエージェント*	78	blob (バイトのシーケンス)。2 フィールド: 必須およびエージェント IP リスト
slp サービス スコープ*	79	blob (バイトのシーケンス)。2 フィールド: 必須および slp スコープリスト
smtp-servers*	69	IP アドレス
start-time-of-state	153	符号なし時間
static-routes*	33	IP アドレス
status-code	151	バイナリ;2 つのフィールド: ステータス コードとステータス メッセージ
ストリートトークディレクトリ - アシスタンスサーバー*	76	IP アドレス
ストリートトークサーバー*	75	IP アドレス
サブネット-アロク	220	blob（バイトのシーケンス）。5 つのフィールド : フラグ、サブネット要求、サブネット情報、サブネット名、サブネット推奨リース時間
subnet-mask	1	IP アドレス（IP Address）
subnet-selection	118	IP アドレス
スワップサーバー	16	IP アドレス
リダイレクト	143	カウントされた len バイト配列のセット。2 つのフィールド: サイズと url
tcp-キープアライブゴミ	39	バイト値ブール型
TCP キープアライブ内部	38	符号なし時間
tftp-server	66	文字列
tftp-server-address*	150	IP アドレス
time-offset	2	署名された時間
time-servers*	4	IP アドレス

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
トレーラーカプセル化	34	バイト値ブール型
tzdb タイムゾーン	101	文字列（RFC 4833 を参照）
ユーザー認証	98	文字列
v4-pcp-server*	158	binary
v4-portparams	159	バイナリ;3 つのフィールド: オフセット、psid-len、および psid
v-i-ベンダークラス	124	BLOB（バイト シーケンス）
v-i-ベンダー情報	125	BLOB（バイト シーケンス）
vendor-encapsulated-options	43	BLOB（バイト シーケンス）
vpn-id	185	ブロブ(構造化)。2 フィールド: フラグと vpn-id
www サーバー*	72	IP アドレス
xディスプレイマネージャ*	49	IP アドレス

次の表に、DHCPv6 オプションを示します。

（注） これらのオプションへのアクセスは、 、 putOptiongetOption、およびremoveOptionメソッドを使用してのみ使用できます。

表 *78 : DHCPv6* のオプション

名前（* = 複数値）	ケース	値
4rd	97	コンテナ (オプション)
4rd-map-rule	98	バイナリ;6 つのフィールド: prefix4-len、プレフィックス6レン、ea-len、フラグ、ルール ipv4 プレフィックス、およびルール ipv6 プレフィックス
4rd-non-map-rule	99	バイナリ;3 つのフィールド: フラグ、トラフィック クラス、およびドメイン pmtu
access-domain	57	DNS name
アドルセル	84	バイナリ;1 フィールド: 予約済み AP

名前（* = 複数値）	ケース	値
アドルセルテーブル	85	バイナリ;3 つのフィールド: ラベル、優先順位、およびプレフィックス
aftr-name	64	DNS name
アニ・アプ・ブシド	108	BLOB（バイト シーケンス）
アニ・アプ・ネーム	107	文字列
アニアット	105	バイナリ;2 フィールド: 予約済みおよび att
ani-ネットワーク名	106	文字列
ani演算子 ID	109	BLOB（バイト シーケンス）
アニ演算子レルム	110	文字列
auth	11	バイナリ;5つの分野:プロトコル、アルゴリズム、リプレイ検出方式、リプレイ検出、認証情報
bcmcs-サーバー-a*	34	[IPv6 アドレス（IPv6 address）]
bcmcs-サーバー-d*	33	DNS name
ブートファイルパラム	60	カウント型。2 つのフィールド: typecnt サイズとパラメーター
ブートファイル-URL	59	文字列
cablelabs-17 (vendor-opts ID: 4491)	17 サブオプション：	vendor-opts; 27 サブオプション
oro	1	16 ビットの符号なし整数
device-type	2	string
埋め込みコンポーネント-リスト	3	文字列
device-serial-number	4	文字列
hardware-version-number	5	文字列
ソフトウェアバージョン番号	6	string
ブート・ロム・バージョン	7	文字列

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
vendor-oui	8	文字列
モデル番号	9	文字列
vendor-name	10	文字列
ecm-cfg-カプセル化	15	文字列
tftp-servers	32	[IPv6 アドレス（IPv6 address）]
config-file-name	33	文字列
syslog-servers	34	[IPv6 アドレス（IPv6 address）]
モデム機能	35	binary
device-id	36	binary
rfc868-servers	37	[IPv6 アドレス（IPv6 address）]
time-offset	38	符号付き時間
IP-プレフ	39	8 ビットの符号なし整数
acs-server	40 サブオプション：	バイナリ;2 サブオプション
flag	[0]	8 ビットの符号なし整数
サーバー	[0]	
radius-server	41 サブオプション：	バイナリ;2 サブオプション
flag	[0]	8 ビットの符号なし整数
サーバー	[0]	
セル ID	54	[IPv6 アドレス（IPv6 address）]
チャッピングコア	61	[IPv6 アドレス（IPv6 address）]
cmts機能	1025	binary
cm-mac-address	1026	binary
eルーターコンテナ	1027	binary

名前（* = 複数値）	ケース	値
ケーブルラボ-クライアント構成	2170 サブオプション：	バイナリ;2 サブオプション (各種データ・タイプ)
primary-dhcp-server	1	IP アドレス（IP Address）
secondary-dhcp-server	2	IP アドレス
cablelabs-client-configuration-v6	2171 サブオプション：	バイナリ;9 サブオプション (各種データ・タイプ)
プライマリ dhcpv6-server- セレクタ ID	1	binary
セカンダリ dhcpv6-server- セレクタ ID	2	binary
provisioning-server	3	binary
バックオフ再試行	4	binary
ap-backoff 再試行	5	binary
kerberos-realm	6	DNS name
使用-tgt	7	符号なし 8 ビット
provisioning-timer	8	符号なし 8 ビット
チケットコントロールマスク	9	符号なし 16 ビット
ケーブルラボ-相関 ID	2172	符号なし 32 ビット
captive-portal	103	文字列
capwap-ac-v6*	52	[IPv6 アドレス（IPv6 address）]
クライアントアーチタイプ*	61	符号なし 16 ビット
client-data	45	（オプションの）コンテナ
client-fqdn	39	バイナリ;2 フィールド: フラグとドメイン名
client-identifier	1	BLOB（バイト シーケンス）
クライアント リンクレイヤー アドレス	79	バイナリ;2 つのフィールド: リンク層タイプとリンク層アドレス

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
clt-time	46	符号なし時間 (RFC 5007 を参照)
dhcp4-o-dhcp6-server	88	[IPv6 アドレス（IPv6 address）]
dhcpv4-msg	87	BLOB（バイト シーケンス）
dns-servers*	23	[IPv6 アドレス（IPv6 address）]
domain-list	24	DNS name
dot-address*	142	[IPv6 アドレス（IPv6 address）]
dots-ri	141	DNS name
elapsed-time	8	符号なし 16 ビット
エロ	43	符号なし 16 ビット (RFC 4994 を参照)
erp-ローカル・ドメイン名	65	DNS name
ジオコンフィシビック	36	binary
ジオロック	63	BLOB（バイト シーケンス）
ia-na	3	バイナリ;3つの分野:アイド、t1、およびt2
ia-pd	25	バイナリ、3個のフィールド（iaid、t1、t2）
ia-ta	4	バイナリ;1フィールド:アイド
イアアドル	5	バイナリ;3 つのフィールド: 住所、優先存続時間、および有効な有効期間
イアプレフィックス	26	バイナリ;4つのフィールド:優先存続時間、有効期間、プレフィックス長、およびプレフィックス
インフ-マックス-rt	83	符号なし時間
情報更新時間	32	符号なし時間
interface-id	18	BLOB（バイト シーケンス）
ipv6-adddress-andsf*	143	[IPv6 アドレス（IPv6 address）]
krb-デフォルト領域名	77	文字列
クルブ-クdc	78	バイナリ;5 つのフィールド: 優先順位、重み、トランスポート・タイプ、kdc-ipv6-アドレス、およびレルム名

名前（* = 複数値）	ケース	値
krb プリンシパル名	75	バイナリ;2 つのフィールド: 名前型と名前文字列
krb-レルム名	76	文字列
link-address	80	[IPv6 アドレス（IPv6 address）]
失われたサーバー	51	DNS 名（RFC 5223 を参照）
lqベースタイム	100	符号なし 32 ビット
lq クライアントリンク*	48	IPv6 アドレス (RFC 5007 を参照)
lq終了時刻	102	符号なし 32 ビット
lq クエリ	44	バイナリ構造 (RFC 5007 を参照)
lqリレーデータ	47	バイナリ (DHCPv6 メッセージ) (RFC 5007 を参照)
lq-開始時間	101	符号なし 32 ビット
mip6-haa	72	[IPv6 アドレス（IPv6 address）]
mip6-haf	73	DNS name
mip6-hnidf	49	DNS name
mip6-hnp	71	バイナリ;2 つのフィールド: プレフィックス長とプレフィックス
mip6-idinf	69	（オプションの）コンテナ
mip6-udinf	70	（オプションの）コンテナ
mip6-vdinf	50	（オプションの）コンテナ
モスアドレス	54 サブオプション：	バイナリ;3 サブオプション
は	1	[IPv6 アドレス（IPv6 address）]
cs	2	[IPv6 アドレス（IPv6 address）]
es	3	[IPv6 アドレス（IPv6 address）]

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
モスト fqdn	55 サブオプション：	バイナリ;3 サブオプション
は	1	DNS name
cs	2	DNS name
es	3	DNS name
mpl パラメータ	104	BLOB（バイト シーケンス）
泥のURL	112	文字列
新しいポシックスタイムゾーン	41	文字列 (RFC 4833)
新しい tzdb タイムゾーン	54	文字列（RFC 4833）
nii	62	バイナリ;3 つのフィールド: タイプ、メジャー、マイナー
nis-domain-name*	29	DNS name
nis-サーバー*	27	IP アドレス
nisp-domain-name*	30	DNS name
nispサーバー*	36	IP アドレス
ntp-server	72	バイナリ;3 サブオプション (各種データ・タイプ)
oro	6	符号なし 16 ビット
パナエージェント*	40	IPv6 アドレス (RFC 5192 を参照)
pd除外	67	バイナリ;2 フィールド: プレフィックス長とサブネット ID
環境設定	7	符号なし 8 ビット
prefix64	113	バイナリ;3 つのフィールド: ASM-m プレフィックス64、SSM-m プレフィックス64、および u プレフィックス 64
radius	81	BLOB（バイト シーケンス）
rapid-commit	14	ゼロサイズ

名前（* = 複数値）	ケース	値
選択	74	バイナリ;3 つのフィールド: 再帰名サーバー、予約および prf、およびドメインとネットワーク
再設定-受け入れる	20	0 サイズ
再設定-メッセージ	19	符号なし 8 ビット
リレー エージェント サブスクライバー ID	38	binary
リレー ID	53	BLOB（バイト シーケンス）
リレーメッセージ	9	binary
リレーポート	135	符号なし 16 ビット
remote-id	37	バイナリ;2 フィールド: エンタープライズ ID とリモート ID
ルスー	66	（オプションの）コンテナ
s46-br	90	[IPv6 アドレス（IPv6 address）]
s46-cont-lw	96	（オプションの）コンテナ
s46-cont-mape	94	（オプションの）コンテナ
s46-cont-mapt	95	（オプションの）コンテナ
s46-dmr	91	IPv6 可変長接頭部
s46-portparams	93	バイナリ、3 個のフィールド（offset、psid-len、psid）
s46-priority*	111	符号なし 16 ビット
s46-rule	89	バイナリ;5 つのフィールド: フラグ、ea レン、プレフィックス4-len、ipv4-prefix、およびプレフィックス6
s46-v4v6bind	92	バイナリ;2 つのフィールド: ipv4 アドレスおよびバインド ipv6 接頭部
server-identifier	2	BLOB（バイト シーケンス）
サーバー ユニキャスト	12	[IPv6 アドレス（IPv6 address）]
サイプサーバーアドレス	22	[IPv6 アドレス（IPv6 address）]

DHCP 拡張ディクショナリ

名前（* = 複数値）	ケース	値
サイプサーバー名	21	DNS name
一ロウアcsドメイン	58	DNS name
サーバー*	31	[IPv6 アドレス（IPv6 address）]
ゾルマックス-rt	82	符号なし時間
status-code	13	バイナリ、2 個のフィールド（status-code、status-message）
リダイレクト	136	カウント型。2 つのフィールド：typecnt サイズと url
ユーザー クラス	15	カウント型。2 つのフィールド: typecnt サイズとユーザーデータ
v6-pcp-server*	86	[IPv6 アドレス（IPv6 address）]
vendor-class	16	vendor-class
vendor-opts	17	ベンダーオプト(cablelabs-17も参照)
vpn-id	68	バイナリ;2 フィールド: フラグと vpn-id

（注） 複数のインスタンス オプションもあります(つまり、1 つのオプションで複数の値だけでなく、複数のインスタンスを設定することもできます)。複数のインスタンスを持つことができるオプションは次のとおりです。

- ia-na
- ia-pd
- ia-ta
- イアアドル
- イアプレフィックス
- 選択
- s46-br
- s46-cont-mape
- v6-pcp-server

要求ディクショナリ

次の表は、要求ディクショナリでいつでも設定できるデータ項目を示しています。DHCPサーバーは、さまざまな時間にこれらのファイルを読み取ります。特に指定されていない限り、すべての操作は読み取り/書き込み可能です。

表 *79*: 要求ディクショナリ固有のデータ項目

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
アクティブリースクエリコントロール	int（v4、v6）
リースの送信を制御します(特定の状態の変更に対してのみなど)。値は、0(未指定(サーバーが通知を送信するかどうかを決定する)、1 - 送信(サーバーが通知を送信する)、および2は送信しない(サーバーは通知を送信しません) です。アクティブリースクエリコントロールは0(指定なし) として初期化されます。	
許可ブート	int（v4）
1に設定すると、この要求に対して任意のスコープに対してBOOTPを許可します。スコープ選択中およびリースの受容性の確認中に読み取り。	
許可する- dhcp	int（v4）
1 に設定すると、この要求のスコープに対して DHCP を許可します。リースの受け入れ確認中およびスコープの選択中に読み取ります。	
許可動的ブート	int（v4）
1に設定すると、この要求の任意のスコープに対して動的BOOTPを使用できます。リースの受け入れ確認中およびスコープの選択中に読み取ります。	
ブート応答オプション	ブロブ (v4)
任意のポリシー内のv4-bootp-replyオプションをオーバーライドします。出力パケットのデータを収集する際に読み取り。(IPv6 ブート応答オプションはありません。	
client-class-name	文字列 (v4、v6)
クライアント情報を完成させるために使用するクライアント クラスの名前 (存在する場合)。読み取り専用です。	
クライアント クラス ポリシー	string（v4、v6）
クライアントクラスに関連付けられているポリシーの名前。設定する場合は、サーバーで既に構成されているポリシーの名前を持つ必要があります。	
client-domain-name	string（v4、v6）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
クライアントが使用するドメイン名。存在しない場合、DHCP サーバーはスコープで指定されたドメイン名を使用します。安定した記憶域の更新の直前にDNS更新の要求をキューに入れるときに読み取り。DHCPv6 の場合、クライアント FQDN 値を上書きし、DNS 更新に使用します。	
client-host-name	string（v4、v6）
DNS 内のクライアントのホスト名。安定したストレージを更新する直前に DNS 更新の要求をキューに入れる場合に読み取りが行われます。操作が完了すると、実際の名前がDNSに配置されます。DHCPv6 の場合、client-fqdn 値を上書きし、DNS 更新に使用します。	
client-id	ブロブ (v4, v6)
サーバーがクライアントを追跡するために使用するクライアント ID。クライアント ID は、要求と共に送信されるか、または MAC アドレスから内部的に生成されます。client-id-created-from-mac-address を参照してください。DHCPv6 の場合、クライアント識別子オプション値 (クライアントの DUID)。	
client-id-created-from-mac-address	int（v4）
1 に設定した場合、クライアント IDはクライアント提供の MAC アドレスから内部使用するために作成する必要があり、レポートで使用しないでください。	
クライアント ip アドレス	IP アドレス（v4）
クライアントがパケットを送信した IP アドレス。クライアントにまだ IP アドレスがない場合は、0 になる可能性があることに注意してください。	
クライアント制限 ID	blob（v4、v6）
クライアントの制限 ID。	
クライアントルックアップ ID	blob（v4、v6）
クライアントクラスのクライアントルックアップID式によって計算されたクライアントルックアップ ID。	
client-mac-address	blob（v4）
要求ディクショナリに関連付けられたクライアント オブジェクトに格納されている MAC アドレス。同じ形式 (およびから作成された) mac アドレスを持っています。	
クライアント-osタイプ	int（v4）
またはpre-client-lookup拡張ポイントでこれを設定して、要求パケットのクライアント エントリpost-client-lookupを変更します。で読むことcheck-lease-acceptableもできますが、そこで設定することはできません。値を設定するには、最初に要求ディクショナリで os-typepost-packet-decodeを設定する必要があります。	

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
client-packet	blob (v4、v6、読み取り専用)
受信パケットのクライアント部分。DHCPv4 の場合、これは完全なパケットです。DHCPv6 の場合、これはクライアント・メッセージです。(パケット全体を取得するには、パケットを参照してください。	
クライアント ポリシー	string（v4、v6）
クライアント エントリに関連付けられているポリシーの名前。設定する場合は、DHCP サーバーで構成済みのポリシーの名前を指定する必要があります。	
client-port	int (v4、v6)
クライアントが要求を送信したポート。	
クライアントが要求したホスト名	文字列 (v4)
クライアントが要求したホスト名が DNS 更新に使用されます。DHCP サーバーはこの情報を保存して、変更を検出できるようにします。	
クライアント ユニキャスト	ブール値 (v6、読み取り専用)
受信パケットがクライアントによってサーバーにユニキャストされた場合は true。	
クライアントが望む null-in-string	int（v4）
DHCP サーバーが、null で終了した文字列をクライアントに返すかどうかを判断します。1 に設定すると、サーバーは null で文字列を終了します。0 に設定した場合、null で文字列が終了することはありません。応答パケットpost-packet-decodeをエンコードする際に前に設定pre-packet-encodeし、 を読み取ります。	
派生 VPN-ID	int (v4、v6、読み取り専用)
VPN 識別子。詳細については、vpn-nameを参照してください。	
destination-ipaddress	IP アドレス (v6、読み取り専用)
パケットの宛先 IPv6 アドレス。	
dhcp 応答オプション	blob（v4、v6）
ポリシーで指定されたv4応答オプションまたはv6応答オプションをオーバーライドします。出力パケットのデータを収集する際に読み取り。	
ダンプパケット	int (v4、v6、書き込み専用)

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
1 に設定すると、Cisco Prime Network レジストラーは、現在デコードされた DHCP/BOOTP パケットをログファイルにダンプします。拡張機能は、その実行の複数のポイントでこのデータ項目に値 1 を配置できます。これは、拡張機能をデバッグするときに役立つ場合があります。	
フェールオーバー ロール	int (v4、v6、読み取り専用)
フェールオーバー サーバーの役割を決定します。フェールオーバー サーバーの役割は、次の 3 つの値のいずれかになります。 • • なし:フェールオーバーが構成されていません。 • メイン/バックアップ:フェールオーバーが構成され、フェールオーバー サーバーの役割が構成されます。	
failover-state	int（v4、v6、読み取り専用）
フェールオーバー サーバーの状態を決定します。フェールオーバーの状態は、正常、パートナーダウン、通信中断、回復、競合の可能性、回復完了、起動、シャットダウン、または一時停止状態にすることができます。フェールオーバーが設定されていない場合、この値は None になります。	
インポート パケット	int（v4）
サーバーがパケットをインポート クライアントから送信されたパケットとして扱うかどうかを決定します。1 に設定すると、サーバーはクライアントをインポート クライアントとして扱い、ACK を送信する前にクライアントに対するすべての DNS 操作を実行します。サーバーのインポート モードを確認するときに (post-packet-decode直後)、DNS 処理の準備、および応答アドレスの設定時に読み取ります。	
制限カウント	int（v4）
同じ制限 IDで許可される同時ユーザー数。	
limitation-id	blob（v4）
この要求が存在するクライアント クラスの制限 id式 (存在する場合) によって計算されます。	
制限 id-null	int（v4、v6）
制限 idが null の場合は 1(TRUE) に設定し、別の値の場合は 0 (FALSE) に設定します。	
ログクライアント基準処理	int（v4、v6）
1 に設定すると、この要求に対するクライアントの基準処理がログに記録されます。クライアントがリースを持たないクライアントの新しいリースを取得しようとする場合、およびリースの受け入れ可能性を確認する場合に、読み取り。	
ログクライアントの詳細	int（v4、v6）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
1 に設定すると、この要求に対するクライアント クラスの処理がログに記録されます。クライアント クラス処理の最後、post-client-lookup後のを読み取ります。	
ログ-dns-更新-詳細	int（v4、v6）
1 に設定すると、この要求の DNS 更新の詳細がログに記録されます。	
ログドロップブートパケット	int（v4）
1 に設定すると、ログはこの要求に対して BOOTP パケットをドロップします。	
ログドロップされた dhcp-パケット	int（v4、v6）
1 に設定すると、ログはこの要求に対して DHCP パケットを廃棄します。	
ログ ドロップ待機パケット	int（v4、v6）
1 に設定すると、ログはこの要求の待機パケットをドロップします。	
ログ フェールオーバーの詳細	int（v4）
1 に設定すると、すべてのフェールオーバー状態の変更など、より詳細なレベルのフェールオーバー アクティビティがログに記録されます。	
ログ着信パケットの詳細	int（v4、v6）
1に設定すると、この要求に対して詳細な着信パケット トレースが発生したかどうかをチェックし、個別のトレースを配置する必要がないようにします。パケットデコード前と最初の拡張ポイントを読み取ります。	
ログ着信パケット	int（v4、v6）
1に設定すると、この要求の着信パケットがログに記録されます。の後post-decode-packetで読んでください。	
ログ ldap 作成の詳細	int（v4）
1 に設定すると、DHCP サーバーがリース状態エントリの作成を開始するたびにメッセージがログに記録され、LDAP サーバーからの応答を受信したり、結果またはエラー メッセージを取得したりします。	
ログ ldap クエリの詳細	int（v4、v6）
1 に設定すると、DHCP サーバーが LDAP サーバーに対する照会を開始したり、LDAP サーバーから応答を受け取ったり、照会結果またはエラー・メッセージを取得したりするたびに、メッセージがログに記録されます。	
ログ ldap 更新の詳細	int（v4）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
1 に設定すると、DHCP サーバーが更新リース状態を開始するたびにメッセージがログに記録されるか、からの応答を受信するか、LDAP サーバーから結果またはエラー・メッセージを取得します。	
ログリースクエリ	int（v4、v6）
1 に設定すると、leasequery パケットが内部エラーなしで処理され、ACK または NAK が生成されたときにメッセージがログに記録されます。	
ログ欠落オプション	int（v4、v6）
1 に設定すると、不足しているオプション (クライアントが要求するが DHCP サーバーから返せない) をログに記録します。応答のデータを収集しながら読み取ります。	
ログ発信パケットの詳細	int（v4、v6）
1 に設定すると、この要求の発信パケットの詳細なダンプが記録されます。pre-packet-encode パケットを DHCP クライアントに送信する直前に読み取ります。	
ログ成功メッセージ	int（v4、v6）
1 に設定すると、成功メッセージがログに記録されます。	
ログ不明基準	int（v4、v6）
1 に設定すると、この要求のクライアント包含基準または除外条件で指定された不明な条件がログに記録されます。新しいクライアント リースを取得するとき、または既存のクライアントのリース受諾性を確認するときに読み取り。	
log-v6-lease-detail	int（v6）
1 に設定すると、DHCPv6 リースアクティビティに関する個々のメッセージがログに記録されます。	
mac-address	blob（v4）
クライアント パケットに入っている MAC アドレス。最初のバイトはハードウェアの種類、2 番目のバイトはハードウェアの長さ、残りのバイトはの直後post-packet-decodeに読み取られたchaddrからの情報です。これは、DHCP パケットのhtype、hlen、およびchaddrフィールドの便利な集約です。読み取り時には、これらのフィールドから構築されます。書き込まれると、これらのフィールドに配置されます。	
最大クライアントルックアップ	整数 (v4、v6)
許容するクライアント データベース ルックアップの最大数。通常は 2 などの小さな整数です。プリセット値は 1 です。	
override-client-id	blob（v4、v6）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
現在のクライアント ID 値に使用される BLOB。着信パケットのクライアント ID を置き換えます (両方の値はリース状態データベースに保持されます)。	
オーバーライドクライアント ID データ型	文字列 (v4、v6、読み取り専用)
オーバーライドクライアント IDのデータ型 (文字列の場合は "nstr" 、BLOB の場合は "blob") を返します。	
オーバーライドクライアント ID 文字列	string（v4、v6）
受信パケットのクライアント ID を置き換える文字列形式の現在のクライアント ID 値 (両方の値はリース状態データベースに保持されます)。getの場合、オーバーライド クライアント IDが文字列でない場合、バイナリ データは BLOB データとして書式設定され、"string" として返されます。	
パケット	blob（v4、v6）
受信パケット。DHCPv4 の場合、これはクライアントパケットと同じです。DHCPv6 の場合、リレーが行われた場合はフルパケット、リレーされない場合はクライアントパケットと同じパケットになります。これは、パケットデコード前の拡張ポイントからのみ書き込む必要があります。その後、サーバーはクライアントから受信したパケットではなく、この新しいパケットをデコードします。	
ping-clients	int（v4）
1 に設定すると、この要求のリースを提供する前に ping を実行します。リースがクライアントに対して許容されるかどうかを判断する前に、直前に読んでください。	
relay-agent-circuit-id	blob（v4、v6）
オプション 82 の回線 ID サブオプションの内容。	
リレーエージェント-サーキット ID データ	blob（v4、v6）
オプション 82 の circuit-id サブオプションのデータ部分のみの内容。	
リレーエージェント-デバイス-クラスデータ	blob（v4、v6）
オプション 82 の装置クラス・サブオプションの内容。	
リレー エージェントの半径属性	blob（v4)
オプション 82 の半径サブオプションの内容。	
リレーエージェント半径クラス	string（v4）
オプション 82 の radius サブオプションのカプセル化された class 属性。	
リレーエージェント半径プール名	string（v4）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
オプション 82 の radius サブオプションのカプセル化された framed-pool 属性。	
リレーエージェント半径-ユーザー	string（v4）
オプション 82 の radius サブオプションのカプセル化された user 属性。	
relay-agent-remote-id	blob（v4、v6）
オプション 82 のリモート ID サブオプションの内容。	
relay-agent-remote-id	blob（v4、v6）
オプション 82 の remote-id サブオプションのデータ部分のみの内容。	
リレー エージェント サーバー ID オーバーライド データ	IPv6 アドレス (v4、v6)
オプション 82 のサーバー ID サブオプションの内容。IANA サブオプション 182 がパケット内にある場合、その値が表示されます。それ以外の場合は、Cisco サブオプション 152 の値が表示されます。	
relay-agent-subscriber-id	string（v4）
オプション 82 のサブスクライバー ID サブオプションの内容。	
リレーカウント	int (v6、読み取り専用)
DHCPv6 リレー ホップの数。	
返信オプション	blob
任意のポリシーで指定された DHCPv4 応答オプションをオーバーライドします。出力パケットのデータを収集するときに読み取り。	
クライアントアドレスへの返信	int（v4、v6）
v4 の場合、1 に設定すると、サーバーは応答パケットをクライアント ip アドレスとクライアントポートに送信します。v6 の場合、1 に設定すると、サーバーは応答パケットを送信側 (クライアントまたはリレー エージェント) のアドレスとポートに返送します。0 の場合、サーバーは RFC の強制アルゴリズムを使用して応答を送信します。	
予約アドレス	IP アドレス (v4、読み取り/書き込み)
クライアント用に予約されているアドレスのリスト。使用可能なスコープに一致する最初の使用可能なアドレス (予約への制限が有効になっている必要があります) がクライアントに割り当てられます。	
reserved-ip6addresses	IP アドレス (v6、読み取り/書き込み)

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
クライアント用に予約されているアドレスのリスト。使用可能なプレフィックスに一致する使用可能なすべてのアドレス (予約に制限が有効になっている必要があります) がクライアントに割り当てられます。	
予約済みプレフィックス	IP アドレス（v6、読み取り/書き込み）
クライアント用に予約されているプレフィックスのリスト。使用可能なプレフィックスに一致するすべての使用可能なプレフィックス (予約制限が有効になっている必要があります) がクライアントに割り当てられます。	
selection-criteria	string（v4、v6）
範囲選択基準を含むコンマ区切りの文字列。	
選択基準 -除外	string（v4、v6）
範囲の除外条件を含むコンマ区切りの文字列。	
送信-ack-ファースト	int（v4、v6)
サーバーはこのデータ項目を無視します。ただし、下位互換性のために、読み込みまたは設定は可能ですが、無視されます。デフォルト値は、0（false）です。	
source-ipaddress	IPv6 アドレス (v6、読み取り専用)
パケットの IPv6 送信元アドレス。	
トレース ID	string（v4、v6、読み取り専用）
パケットをトレースするためにシステムが使用する ID。	
トランザクション時間	int（v4、v6)
入力パケットがデコードされた時間 (1970 年以降の秒数)。	
更新-dns	string（v4、v6）
要求パケットごとに部分的、完全、または動的 DNS 更新を要求しません。入力値と出力値は、1=すべて更新、2=更新-fwdのみ、3=更新-rev-only、および 0=更新なしです。	
update-dns-for-bootp	int（v4）
1 に設定すると、この要求に対する BOOTP 要求の DNS が更新されます。BOOTP の DNS 操作を初期化する直前に読み取ります。	
詳細ログ	int（v4、v6)
1 に設定すると、この要求の詳細メッセージがログに記録されます。処理中のさまざまな時間に読み取り。	

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
VPN 説明	string（v4、v6、読み取り専用）
VPN の説明。詳細については、「vpn-name」を参照してください。	
vpn-name	string（v4、v6、読み取り専用）
VPN の名前。要求ディクショナリには、post-packet-decodeこれらの項目の有効な値が 含まれていませんが、VPNがまだ決定されていないため、他のすべての拡張ポイントで有効な値が設定されています。これは、スクリプトがderived-vpn-idオプションまたはサブオプションを変更post-packet-decodeし、リースに使用される VPN に影響を与えるためです。	
VPN-VPN-ID	blob、通常は7バイト (v4、v6、読み取り専用)
仮想プライベート ネットワーク識別子。詳細については、「vpn-name」を参照してください。	
vpn-vrf-name	string（v4、v6、読み取り専用）
VPN の仮想ルーティングおよび転送テーブル識別子。詳細については、「vpn-name」を参照してください。	

応答ディクショナリ

次の表は、応答辞書でいつでも設定できるデータ項目を示しています。DHCPサーバーは、さまざまな時間にそれらを読み取ります。特に指定されていない限り、操作は読み取り/書き込み可能です。

表 *80 :* 応答ディクショナリ固有のデータ項目

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
アクティブリースクエリコントロール	int（v4、v6）
リースの送信を制御します(特定の状態の変更に対してのみなど)。値は、0(未指定(サーバーが通知を送信するかどうかを決定する)、1 - 送信 (サーバーが通知を送信する)、および2は送信しない (サーバーは通知を送信しません) です。アクティブリースクエリコントロールは 0 (指定なし) として初期化されます。	
クライアント・アクティブ・リース数	int（v6、読み取り専用）
DHCPv6 クライアントのアクティブなリース数。	
クライアント作成時間	int（v4、v6、読み取り専用）
クライアントの作成時刻。	
client-domain-name	文字列 (v4、読み取り専用)

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
リース内のクライアント情報から、クライアントが使用するドメイン名。DHCP サーバーがスコープで指定されたドメイン名を使用する場合、このドメインが存在しない可能性があります。安定した記憶域の更新直前に DNS 更新の要求をキューに入れる時に読み取ります。	
クライアントの有効期限	int（v4、v6、読み取り専用）
このサーバーによってクライアントに与えられた最大のリース有効期限 (1970 年以降の秒単位)。	
client-host-name	string（v4、読み取り専用）
リース内のクライアント情報から、DHCP サーバーが DNS に入れるホスト名。安定したストレージを更新する直前に DNS 更新の要求をキューに入れたときに読み取り。	
client-id	blob（v4、v6、読み取り専用）
リース内のクライアント情報から、サーバーがクライアントを追跡するために使用したクライアント ID。これは、要求と共に送信されたクライアント ID であるか、または MAC アドレスから内部的に生成されたクライアント ID である可能性があります。DHCPv6 の場合、クライアント識別子オプション値 (クライアントの DUID)。	
client-id-created-from-mac-address	int (v4、読み取り専用)
リース内のクライアント情報から。1 に設定した場合、クライアント IDは MAC アドレスから作成する必要があり、レポートでは使用しないでください。	
client-last-transaction-time	int（v4、v6、読み取り専用）
DHCP サーバーがこのクライアントから最後に聞き取った時間 (秒単位)。	
クライアント制限 ID	blob (v4、読み取り専用)
現在のリースに関連付けられているクライアントの制限 ID。	
client-mac-address	blob（v4、読み取り専用）
リース内のクライアント情報から、要求ディクショナリに関連付けられたクライアントオブジェクトに格納されている MAC アドレス。mac-addressと同じ形式 (および作成元) を持つ。	
クライアント-osタイプ	int（v4）
またはpre-client-lookup拡張ポイントでこれを設定して、要求パケットのクライアントエントリpost-client-lookupを変更します。で読むことcheck-lease-acceptableもできますが、そこで設定することはできません。値を設定するには、最初に要求ディクショナリで os-typepost-packet-decodeを設定する必要があります。	
クライアントオーバーライドクライアント ID	blob（v4、v6、読み取り専用）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
現在のクライアント ID 値に使用される BLOB。着信パケットのクライアント ID を置き換えます (両方の値はリース状態データベースに保持されます)。	
クライアントオーバーライド-クライアント ID-データ型	string（v4、v6、読み取り専用）
文字列の場合は nstr またはBLOBの場合、クライアントオーバーライド クライアント IDのデータ型を返します。	
クライアントオーバーライド-クライアント ID 文字列	string（v4、v6、読み取り専用）
着信パケットからのすべての client-id を置き換える文字列形式の現在の client-id 値（両方の値がリース ステータス データベースに保持されていても）。getの場合、クライアントオーバーライド クライアント IDが文字列でない場合、バイナリ データは BLOB データとして書式設定され、"string" として返されます。	
client-packet	blob (v4、v6、読み取り専用)
応答パケットのクライアント部分。DHCPv4 の場合、これは完全なパケットです。DHCPv6 の場合、これはクライアント・メッセージです。(パケット全体を取得するには、パケットを参照してください。パケットエンコード後の拡張ポイントからのみ使用できます。	
クライアント再構成キー	文字列 (v6)
DHCPv6 リースのクライアント再構成キー属性値を返します。	
クライアント再構成キー生成時間	string（v6）
DHCPv6 リースのクライアント再構成-鍵生成時間属性値を戻します。	
クライアントリレーアドレス	IPv6 アドレス（v6、読み取り専用）
(最後の) リレーの送信元 IPv6 アドレス。	
クライアントリレーメッセージ	文字列 (v6、読み取り専用)
最後に中継された DHCPv6 メッセージ (クライアント メッセージを除く)。	
クライアントが要求したホスト名	文字列 (v4)
リース内のクライアント情報から、クライアントが DNS 更新のために要求したホスト名。	
クライアント ユーザー定義データ	string（v4、v6、読み取り専用）
ユーザー定義データ環境ディクショナリ データ項目から派生した、クライアントに以前または現在関連付けられている値を返します。つまりcheck-lease-acceptable、拡張ポイントで要求された場合は、lease-state-change以前に関連付けられた値を返します。またはpre-packet-encode拡張ポイントで要求された場合は、post-send-packet現在の値を返します。	

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
クライアント・ベンダー・クラス	string（v4、v6）
DHCPv4 または DHCPv6 リースのクライアント・ベンダー・クラス属性値を戻します。	
クライアントベンダー情報	string（v4、v6）
DHCPv4 または DHCPv6 リースのクライアント・ベンダー情報属性値を戻します。	
クライアント書き込みシーケンス	int（v6、読み取り専用）
クライアント IPv6 要求の書き込みシーケンス。	
クライアント書き込み時刻	int（v6、読み取り専用）
クライアント IPv6 書き込み要求の時刻。	
派生 VPN-ID	int (v4、v6、読み取り専用)
VPN 識別子。	
ドメイン名変更	int（v4）
1 に設定すると、現在のパケットのドメイン名は、DNS 更新で使用されるドメイン名とは異なります。前check-lease-acceptableと前にpre-packet-encode読んでください。	
ダンプパケット	int (v4、v6、書き込み専用)
1 に設定すると、Cisco Prime Network レジストラーは、現在デコードされた DHCP/BOOTP パケットをログ ファイルにダンプします。拡張機能は、その実行の複数のポイントでこのデータ項目に値 1 を配置できます。これは、拡張機能をデバッグするときに役立つ場合があります。	
フェールオーバー ロール	int (v4、v6、読み取り専用)
フェールオーバー サーバーの役割を決定します。フェールオーバー サーバーの役割は、次の 3 つの値のいずれかになります。 • • なし:フェールオーバーが構成されていません。 • メイン/バックアップ:フェイルオーバーが構成され、フェイルオーバー・サーバーの役割	
failover-state	int（v4、v6、読み取り専用）
フェールオーバー サーバーの状態を決定します。フェールオーバーの状態は、正常、パートナーダウン、通信中断、回復、競合の可能性、回復完了、起動、シャットダウン、または一時停止状態にすることができます。フェールオーバーが設定されていない場合、この値は None になります。	

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
host-name-changed	int（v4）
1 に設定すると、現在のパケットのホスト名は、DNS 更新で使用されるホスト名とは異なります。check-lease-acceptable 後および pre-packet-encode 前に読み取ります。	
ホスト名-イン DNS	int（v4、v6）
1 に設定すると、ホスト名は DNS 内にあります。check-lease-acceptable 後および pre-packet-encode 前に読み取ります。ホスト名が DNS に入った後に書き込まれます。	
リース・バインディング・アイイド	int（v6、読み取り専用）
IPv6 リース バインディング IAID。	
リース バインディング再バインド時間	int（v6、読み取り専用）
IPv6 リース バインディング再バインド時間。	
リースバインディング-更新時間	int（v6、読み取り専用）
IPv6 リース バインディングの更新時間。	
リース バインディング タイプ	string（v6、読み取り専用）
IPv6 リース バインディングの種類: "IA_NA"、"IA_TA"、または "IA_PD"	
リースクライアント予約済み	int（v4、v6、読み取り専用）
リースがクライアント予約の場合は 1 を返し、予約されていない場合は 0 を返します。	
リース作成時間	string（v6、読み取り専用）
IPv6 リースの作成時間。	
リース非アクティブ化	int（v4、v6、読み取り専用）
1 に設定すると、リースが非アクティブであることを報告します。	
リース-DNS-フォワード-バックアップ-サーバーアドレス	IP アドレス (v4、v6、読み取り専用)
リース DNS 転送サーバー アドレスで指定されたサーバーがダウンしている場合に、DHCPv4 および DHCPv6 のリースに対する DNS 更新を受信するバックアップDNS サーバーのアドレス。	
リース-DNS-フォワード-サーバーアドレス	IP アドレス（v4、v6、読み取り専用）
DHCPv4 および DHCPv6 リースの動的 DNS 更新を受信する DNS サーバーのアドレス。	
リース-DNS-フォワード-アップデート	string（v4、v6、読み取り専用）

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
DHCPv4 および DHCPv6 リースの DNS 更新に含める転送ゾーンを決定する更新の構成の名前。更新すべてまたは更新のみ fwdが設定されている場合は TRUE を返します。	
リース DNS 転送ゾーン名	string（v4、v6、読み取り専用）
DNS 更新用のオプションの転送ゾーンの名前。	
リース-DNSリバースバックアップ-サーバーアドレス	IP アドレス（v4、v6、読み取り専用）
リース-DNS-リバースサーバーアドレスで指定されたサーバーがダウンしている場合、DHCPv4 および DHCPv6 リースの DNS 更新を受信するバックアップ DNS サーバーのアドレス。	
リース-dns-リバース-ホストバイト	int（v4、読み取り専用）
リバース ゾーンに使用するリース IP アドレスのバイト数。	
リース-dns-リバース-プレフィックス長	int（v6、読み取り専用）
ip6.arpa 更新の逆ゾーンのプレフィックス長。	
リース-DNS-リバースサーバーアドレス	IP アドレス（v4、v6、読み取り専用）
DHCPv4 および DHCPv6 リースの動的 DNS 更新を受信する DNS サーバー アドレスのアドレス。	
リース-DNSリバース更新	string（v4、v6、読み取り専用）
DHCPv4 および DHCPv6 リースの DNS 更新に含めるリバース ゾーンを決定する更新構成の名前。 update-all または update-fwd-only が設定されている場合は、TRUE を返します。	
リース-dns-リバースゾーン名	string（v4、v6、読み取り専用）
PTR レコードで更新される DNS 逆引き (in-addr.arpa および ip6.arpa) ゾーン。	
リース-fqdn	string（v6、読み取り専用）
サーバーによって DHCPv6 リースに割り当てられた完全修飾ドメイン名 (DNS に正常に入力された場合もあります)。 リース fqdnは、リースの DNS に追加される予定の名前、または実際に追加された名前である可能性があります。ホスト名の dnsが true の場合、実際のリース FQDNは DNS 内にあります。	
リースジャッダー	IP アドレス (v4、読み取り専用)
リース・ジャドル	

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
リース-ipアドレス	IPv4 または IPv6 のアドレスまたはプレフィックス (v4、v6、読み取り専用)
DHCPv4 の場合、クライアントに関連付けられているリースのアドレス。DHCPv6 の場合、現在のコンテキストのリースの IPv6 アドレスまたは IPv6 プレフィックス (アドレスとプレフィックス長) (setObject メソッドを参照)。	
リース優先の有効期間	int（v6、読み取り専用）
IPv6 リースの優先有効期間。	
リースプレフィックス名	string（v6、読み取り専用）
IPv6 リースのプレフィックス名。	
リースリレーエージェント情報	blob（v4）
オプション 82 の内容全体。	
リースリレー-エージェント-回路-ID	blob（v4）
応答のリースに格納されているリレー エージェントの回線 ID にアクセスし、操作します。サブオプション番号 1 を最初のバイトとして指定する必要があります。リースリレーエージェント回線 ID-データ項目を支持して非推奨。	
リースリレー-エージェント-サーキット-IDデータ	blob (v4、廃止されたリースリレーエージェント-サーキット IDの代わりに使用)
応答のリースに格納されたリレー エージェント-回線 ID データにアクセスし、操作します。	
リースリレー-エージェント-デバイス-クラスデータ	blob（v4）
オプション 82 の device-class サブオプションの内容。	
リースリレーエージェント半径属性	blob（v4）
オプション 82 の半径サブオプションの内容。	
リースリレー-エージェント半径クラス	string（v4）
オプション 82 の radius サブオプションのカプセル化された class 属性。	
リースリレー-エージェント-半径-プール名	string（v4）
オプション 82 の radius サブオプションのカプセル化された framed-pool 属性。	
リースリレー-エージェント半径-ユーザー	string（v4）

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
オプション 82 の radius サブオプションのカプセル化された user 属性。	
リースリレー-エージェント-リモート ID	blob（v4)
応答のリースと共に格納されたリレーエージェントリモートIDデータにアクセスし、操作します。サブオプション番号 2 を最初のバイトとして指定する必要があります。リース・リレー・エージェント-リモート ID-データ項目を支持して非推奨。	
リースリレー-エージェント-リモートIDデータ	blob (v4、リースリレー-エージェント-リモート ID項目の代わりに使用)
応答のリースと共に格納されたリレー エージェント-リモート ID データにアクセスし、操作します。	
リースリレー-エージェント-サーバー-id- オーバーライドデータ	IP アドレス（v4）
応答のリースと共に格納されているリレー エージェント-サーバー ID オーバーライドデータにアクセスし、操作します。	
リースリレーエージェント-サブネット- 選択データ	IP アドレス（v4）
応答のリースと共に格納されたリレーエージェントサブネット選択データにアクセスし、操作します。	
リースリレーエージェント加入者 ID	string（v4）
オプション 82 のサブスクライバー ID サブオプションの内容。	
リースリレー-エージェント-vpn-id-データ	blob（v4)
応答のリースと共に格納されているリレー エージェント-vpn-idデータにアクセスし、操作します。	
リース要求の fqdn	string（v6、読み取り専用）
DHCPv6 リースのためにクライアントによって最後に要求された部分的または完全修飾ドメイン名。	
リース要求のプレフィックス長	int（v6、読み取り専用）
記録されたクライアントの要求されたプレフィックス長 (指定されている場合) IA_PD バインディング。クライアントが特定のプレフィックス長の要求を送信しなかった場合は、0 になります。	
リース予約済み	int（v4、v6、読み取り専用）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
リースがリース予約されている場合は 1 を返し、リースが予約されていない場合は 0 を返します。	
リース開始時期の状態	int（v4、v6、読み取り専用）
1970年以降の秒数で、このリースが最初に現在の状態に置かれた時間。	
リース状態	string（v4、v6、読み取り専用）
リースの状態(利用可能、提供、リース、期限切れ、利用不可、リリース済、その他利用可能(DHCPv4 のみ)、保留あり(DHCPv4 のみ)、または取り消し(DHCPv6 のみ)	
リース状態の有効期限	int（v4、v6、読み取り専用）
リース状態の有効期限です。	
リースステータス	string（v4、v6、読み取り専用）
「存在しない」、「クライアントによって所有される」、「存在する」を返します。リースが存在するかどうか、および現在のクライアントがリースを所有しているかどうかを判断するために使用されます。"exists" が返された場合、リースは存在しますが、現在の所有者はリースを所有していません (リースに関する限られた情報が利用可能です)。	
リース有効な有効期間	int（v6、読み取り専用）
IPv6 リースの有効期間が有効です。	
リース VPN 説明	string（v4、v6、読み取り専用）
応答のリースと共に格納された VPN の説明。	
リース VPN-ID	int（v4、v6、読み取り専用）
応答のリースと共に格納される VPN の識別子。	
リース VPN-名前	string（v4、v6、読み取り専用）
応答のリースと共に格納された VPN の名前。	
リース VPN-VPN-ID	blob、通常 7 バイト、v4、v6、読み取り専用
応答のリースと共に格納された仮想プライベート ネットワーク (VPN) 識別子。	
リース vpn-vrf-name	string（v4、v6、読み取り専用）
応答のリースと共に格納された VPN の仮想ルーティングと転送テーブル識別子。	
mac-address	blob（v4）

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
クライアント パケットに入っている MAC アドレス。最初のバイトはハードウェアタイプ、2番目のバイトはハードウェアの長さ、残りのバイトはchaddrからの情報です。これは、DHCPパケットのhtype、 hlen、およびchaddrフィールドの便利な集約です。読み取り時には、これらのフィールドから構築されます。書き込まれると、これらのフィールドに配置されます。	
override-client-id	blob（v4、v6、読み取り専用）
現在のクライアント ID 値に使用される BLOB。着信パケットのクライアント ID を置き換えます (両方の値はリース状態データベースに保持されます)。	
オーバーライドクライアント ID データ型	文字列 (v4、v6、読み取り専用)
オーバーライドクライアント IDのデータ型 (文字列の場合は "nstr" 、BLOB の場合は "blob") を返します。	
オーバーライドクライアント ID 文字列	string（v4、v6、読み取り専用）
受信パケットのクライアント ID を置き換える文字列形式の現在のクライアント ID 値 (両方の値はリース状態データベースに保持されます)。 getの場合、オーバーライド クライアント IDが文字列でない場合、バイナリ データは BLOB データとして書式設定され、"string" として返されます。	
パケット	blob (v4、v6、でのみ使用post-packet-decode)
応答パケット。DHCPv4 の場合、これは client-packet と同じです。DHCPv6 では、リレーの場合、フル パケットとなり、リレーでない場合は、client-packet と同じとなります。これは、パケットエンコード後の拡張ポイントからの読み取りまたは書き込みのみにする必要があります。書き込まれると、サーバーは新しいパケットをクライアントに送信します。	
ping-clients	int（v4）
1 に設定すると、この要求のリースを提供する前に ping を実行します。クライアントリースの受け入れ可を判断する直前に読んでください。	
prefix-address	IPv6 プレフィックス (v6、読み取り専用)
プレフィックス アドレス (17 バイト— IPv6 アドレスおよびプレフィックス長)。	
プレフィックス割り当てランダム	int（v6、読み取り専用）
プレフィックスはランダムに割り当てられます。	
プレフィックス割り当て - 最適な方法で割り当て	int（v6、読み取り専用）
最適フィットを介して割り当てられたプレフィックス。	

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
プレフィックス割り当て-経由クライアント要求	int（v6、読み取り専用）
クライアント要求を介して割り当てられたプレフィックス。	
拡張子によるプレフィックス割り当て-割り当て	int（v6、読み取り専用）
拡張機能を介して割り当てられたプレフィックス。	
プレフィックス割り当てインターフェイス経由の識別子	int（v6、読み取り専用）
インターフェイス識別子を介して割り当てられたプレフィックス。	
予約によるプレフィックス割り当て	int（v6、読み取り専用）
予約を介して割り当てられたプレフィックス。	
プレフィックス割り当てグループ	string（v6、読み取り専用）
プレフィックスの割り当てグループ名。	
プレフィックス割り当てグループの優先順位	int（v6、読み取り専用）
プレフィックスの配賦グループの優先順位。	
プレフィックス非アクティブ化	int（v6、読み取り専用）
プレフィックスが非アクティブかどうかを示します。	
プレフィックス- dhcp タイプ	string（v6、読み取り専用）
プレフィックス DHCP タイプ。	
プレフィックスの有効期限	string（v6、読み取り専用）
プレフィックスの有効期限です。	
プレフィックス リンク グループ名	string（v6、読み取り専用）
リンクのリンク グループ名。	
プレフィックス リンク名	string（v6、読み取り専用）
プレフィックスのリンク。	
プレフィックス-リンクタイプ	string（v6、読み取り専用）
リンクの種類 (トポロジ、場所に依存しない、またはユニバーサル)。	

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
prefix-name	string（v6、読み取り専用）
プレフィックスの名前。	
prefix-range	IPv6 アドレス（v6、読み取り専用）
プレフィックスの IPv6 アドレス範囲。	
プレフィックス範囲終了	IPv6 アドレス（v6、読み取り専用）
プレフィックスの範囲の終点 (範囲の開始または範囲の終わりが構成されている場合)。	
プレフィックス範囲の開始	IPv6 アドレス（v6、読み取り専用）
プレフィックスの範囲の開始 (範囲の開始または範囲の終点が構成されている場合)。	
プレフィックス制限から予約へ	int（v6、読み取り専用）
1 に設定すると、プレフィックスの予約制限が有効になります。	
プレフィックス選択タグ	string（v6、読み取り専用）
プレフィックスの選択タグ。	
リレーカウント	int (v6、読み取り専用)
DHCPv6 リレー ホップの数。	
返信-ipアドレス	IPv4 または IPv6 アドレス (v4、v6)
DHCP クライアントに応答するときに使用する IP アドレス。ちょうど後にpre-packet-encode読んでください。で値をpre-packet-encode変更する場合は、その中に配置する IP アドレスは、(ブロードキャスト アドレスでない限り) ARP クエリに応答できるシステムに対するアドレスである必要があります。ユニキャストが有効で、ブロードキャスト フラグが DHCP 要求で設定されていない場合でも、ローカル ARP キャッシュは、DHCP 要求のMAC アドレス pre-packet-encodeへの新しい応答 IP アドレスからのマッピングで設定されません。	
応答ポート	int（v4、v6)
DHCP クライアントに応答するときに使用するポート。pre-packet encode 直後に読み取ります。	
応答ソース	string（v4、v6、読み取り専用）

DHCP 拡張ディクショナリ

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
応答のソース (拡張機能を呼び出した主要なアクティビティ)。出力値は、クライアント (受信クライアント パケット)、フェールオーバー (フェールオーバー パートナーからのバインディング更新の受信)、タイムアウト (リースの有効期限または猶予期間終了)、オペレータ (ユーザー インターフェイスからの要求)、1 クライアントあたりのリース (クライアントごとに 1 つのリースを削除する)クライアントは、新しいリースのために古いリースから、不明(上記のどれも) このデータ項目は、要求ディクショナリが存在するかどうかの処理を決定する拡張機能を支援します。(isValid メソッドは、ディクショナリが有効かどうかを判断するためにも使用できます)。	
逆の名前の dns	int（v4、v6)
1 に等しい場合、逆の名前は DNS に含まれています。DNS 操作を初期化する前に読んでください。	
スコープ許可ブート	int（v4、読み取り専用）
1 に設定すると、スコープは BOOTP を許可します。DNS 操作が完了した後に書き込まれます。	
スコープ許可- dhcp	int（v4、読み取り専用）
1 に設定すると、スコープは DHCP を許可します。	
スコープ許可動的ブート	int（v4、読み取り専用）
1 に設定すると、スコープは動的 BOOTP を許可します。	
スコープ利用可能なリース	int（v4、読み取り専用）
現在のスコープで使用可能なリースの数。	
スコープ非アクティブ化	int（v4、読み取り専用）
1 に設定すると、スコープは非アクティブになります。	
スコープ-DNS-フォワード-サーバーアドレス	IP アドレス（v4、読み取り専用）
DNS 転送アドレスに使用する DNS サーバー。	
スコープ-DNS転送ゾーン名	string（v4、読み取り専用）
スコープに構成された転送ゾーン名。	
スコープ-DNS-ホストバイト数	int（v4、読み取り専用）
DNS 更新を処理する DHCP サーバー コードによって使用されるホスト バイト数です。	

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
スコープ-DNS-逆サーバーアドレス	IP アドレス（v4、読み取り専用）
DNS 逆アドレスに使用する DNS サーバー。	
スコープ-DNS-逆ゾーン名	string（v4、読み取り専用）
スコープに構成された逆引きゾーン名。	
scope-name	string（v4、読み取り専用）
DHCP サーバーが処理しているリースを含むスコープの名前。	
スコープネットワーク番号	IP アドレス（v4、読み取り専用）
DHCP サーバーが処理しているリースを含むスコープのネットワーク番号。	
スコープ ping クライアント	int（v4、読み取り専用）
1 に設定すると、現在のリースに関連付けられたスコープは、リースを提供する前に ping 操作をサポートするように構成されています。	
スコープ-プライマリ ネットワーク番号	IP アドレス（v4、読み取り専用）
このプライマリ スコープのネットワーク番号。	
スコープ-プライマリ サブネット マスク	IP アドレス（v4、読み取り専用）
このプライマリ スコープのサブネット マスク。	
スコープの更新のみ	int（v4、読み取り専用）
1 に設定すると、スコープは更新のみになります。	
スコープ更新のみ期限切れ時刻	int（v4、読み取り専用）
1970 年 1 月 1 日以降の絶対時間 (秒単位) で、更新のみのスコープは更新のみ停止する必要があります。	
スコープ制限の予約	int（v4、読み取り専用）
1 に設定すると、スコープの予約制限が有効になります。	
スコープ選択タグ	string（v4、読み取り専用）
スコープ選択基準を含むコンマ区切りの文字列。このデータ項目は、スコープに基づく決定に使用します。	
スコープ-送信-ack-最初	int（v4、読み取り専用）
1 に設定すると、スコープは残りの処理を実行する前に ACK を送信します。	

データ項目	値 (プロトコル: v4=DHCPv4, v6=DHCPv6)
スコープ サブネット マスク	IP アドレス（v4、読み取り専用）
DHCP サーバーが処理しているリースを含むスコープのサブネット マスク。	
スコープ更新- DNS	string（v4、読み取り専用）
前方ゾーンまたは逆ゾーンの DNS 更新。出力値は、1 =すべて更新、2=更新-fwd のみ、3=更新-rev-only、および 0=更新なしです。	
スコープ更新-DNS が有効	ブール値 (v4、読み取り専用)
1 に設定すると、スコープの更新 DNS が転送ゾーンと逆方向ゾーンに対して有効になります。スコープ更新 DNSを優先して非推奨。	
スコープ更新- dns-for-bootp	int（v4、読み取り専用）
1 に設定すると、スコープの更新 DNS が BOOTP に対して有効になります。	
トレース ID	string（v4、v6、読み取り専用）
パケットをトレースするためにシステムが使用する ID。	
トランザクション時間	int（v4、v6、読み取り専用）
要求がデコードされた時間 (1970 年以降の秒単位)。	
VPN 説明	string（v4、v6、読み取り専用）
VPN の説明。	
vpn-name	string（v4、v6、読み取り専用）
VPN の名前。	
VPN-VPN-ID	blob、通常は 7 バイト (v4、v6、読み取り専用)
仮想プライベート ネットワーク（VPN）識別子。	
vpn-vrf-name	string（v4、v6、読み取り専用）
VPN の仮想ルーティングおよび転送テーブル(VRF)識別子。	

拡張ディクショナリ API

このセクションには、Tcl 拡張および共有ライブラリから辞書にアクセスするときに使用するディクショナリ メソッドの呼び出しが含まれています。

TCL 属性ディクショナリ API

属性ディクショナリでは、キーは Cisco Prime Network レジストラー DHCP サーバー設定で定義されている属性の名前に制限されます。値は、その特定の属性の有効な値の文字列表現です。たとえば、IP アドレスはアドレスのドット付き 10 進文字列表現で指定され、列挙値は列挙型の名前で指定されます。つまり、数値は、数値の文字列形式で指定されます。

属性ディクショナリは、キーのインスタンスを複数含めることができるという点で珍しいです。これらのインスタンスは順序付けされ、最初のインスタンスはインデックス 0 になります。属性ディクショナリ メソッドの中には、参照するインスタンスのリスト内の特定のインスタンスまたは位置を示すインデックスを許可するものもあります。

TCL の要求ディクショナリと応答ディクショナリ メソッド

属性ディクショナリでは、コマンドを使用して、ディクショナリ内の値を変更したり、値にアクセスしたりできます。次の表は、要求辞書と応答辞書で使用するコマンドの一覧です。この場合、dict変数を またはrequestresponseとして定義できます。

例については、インストールパス/例/dhcp/tcl/tclextension.tcl.tcl ファイルを参照してください。

表 *81 : TCL* の要求ディクショナリと応答ディクショナリ メソッド

方法	構文
get	$dict get 属性[インデックス[bMore]]
ディクショナリから属性の値を返します。ディクショナリに属性が含まれていない場合は、代わりに空の文字列が返されます。インデックス値を含める場合、属性のインデックス番目のインスタンスが返されます。一部の属性は、要求パケットまたは応答パケットに複数回出現する可能性があります。インデックスは、返すインスタンスを選択します。 bMoreを含める場合、getメソッドは、返された属性の後に属性が多い場合は true に設定し、それ以外の場合は FALSE に設定します。これを使用して、 get をもう一度呼び出してその属性の他のインスタンスを取得するかどうかを決定します。	
getOption	$dict getOption arg型[arg-data]
オプションのデータを文字列として取得します。arg 表 82 : Tcl の arg 型と obj 型の値 （597 ページ）型の値については、「」を参照してください。次の引数が数値の場合は、数値、それ以外の場合は名前と見なされます。この関数は文字列へのポインタを返すことに注意してください。オプションが存在しない、またはオプションの長さが 0 の場合は、ポインタは長さ 0 になります。サンプルの使用方法については、「ベンダー クラス オプション データの処理 （616 ページ）」を参照してください。	
isValid isV4 isV6	$ディクストisValid $ディクトisV4 $isV6

DHCP 拡張ディクショナリ

方法	構文
(isValid渡されたディクショナリに応じて) 要求または応答がある場合、メソッドは TRUE を返します。それ以外の場合はFALSE。などの拡張機能はlease-state-change、ディクショナリが使用可能かどうかを判断するために、このメソッドを使用できます。 このisV4拡張が DHCPv4 パケットに対して呼び出されている場合、メソッドは TRUE を返します。それ以外の場合は FALSE。ルーチンからこのメソッドをinit-entry呼び出すと、FALSE が返されます。 このisV6拡張が DHCPv6 パケットに対して呼び出されている場合、このメソッドは TRUE を返します。それ以外の場合は FALSE。init-entry ルーチンからこのメソッドを呼び出すと、FALSE が返されます。	
log	$dict log level message ...
DHCP サーバーログシステムにメッセージを書き込みます。レベルは、LOG_ERROR、LOG_WARNING、またはLOG_INFOにする必要があります。残りの引数は連結され、指定されたレベルでロギングシステムに送信されます。 （注） サーバーはログ ファイルをこれらのレベルでログに記録するメッセージでフラッシュするので、LOG_ERROR レベルとLOG_WARNINGレベルを慎重に使用します。頻繁に発生しそうなメッセージ（クライアント要求など）に対してこれらのレベルを使用すると、ディスク I/O パフォーマンスに重大な影響を及ぼす可能性があります。	
moveToOption	$dict moveToOption arg型[arg-data] ..
後続の get、put、remove オプション操作のコンテキストを設定します。arg-type 値については、表 82 : Tcl の arg 型と obj 型の値 （597 ページ） を参照してください。オプションが削除されるとコンテキストが無効になる可能性があることに注意してください (などremoveOption) 。	
put	$dict put 属性値[インデックス]
ディクショナリ内の属性に値を関連付けます。インデックスを省略するか、特殊な値REPLACE に設定すると、属性の既存のインスタンスが単一の値に置き換えられます。インデックス値を特殊な値APPEND として指定すると、属性のインスタンスリストの末尾に属性の新しいインスタンスが追加されます。インデックス値を数値として含める場合は、指定された位置に属性の新しいインスタンスが挿入されます。インデックス値を特別な値AUGMENT に設定した場合、属性が追加されるのは、属性が存在しない場合のみです。	
putOption	$dict putOptionデータのarg型[arg-data] ..
オプションとそのデータを追加する、またはオプションのデータを変更します。arg-type 値については、表 82 : Tcl の arg 型と obj 型の値 （597 ページ） を参照してください。サンプルの使用方法については、「ベンダー クラス オプション データの処理 （616 ページ）」 を参照してください。	
remove	$dict remove 属性[インデックス]

方法	構文
ディクショナリから属性を削除します。インデックスを省略するか、REMOVE_ALL特殊な値に設定すると、属性の既存のインスタンスが削除されます。インデックスを数値として含める場合、指定された位置にある属性のインスタンスが削除されます。このメソッドは、ディクショナリがそのインデックスに属性を含んでいない場合でも、常に 1 を返します。	
removeOption	$dict removeOption arg-type [arg-data] ...
オプションを削除します。arg-type 値については、表 82 : Tcl の arg 型と obj 型の値 （597 ページ） を参照してください。サンプルの使用方法については、「ベンダー クラス オプション データの処理 （616 ページ）」を参照してください。	
setObject	$setObjectオブジェクト型[データ]
(DHCPv6 のみ)。get、put や removeメソッドのオブジェクトを設定し、新しいオプション メソッドが動作するメッセージを変更します。obj表 82 : Tcl の arg 型と obj 型の値 （597 ページ）型の値については、「」を参照してください。DHCPv6 拡張機能は、主にこのメソッドを使用して、クライアントとリンクで使用可能なリースおよびプレフィックスにアクセスしたり、リレー パケットからメッセージ ヘッダー フィールドまたはオプションを取得したりします。1 つのリースとスコープが応答に関連付けられている DHCPv4 とは異なり、DHCPv6 応答には複数のリースとプレフィックスが含まれる場合があります。オブジェクトが存在する場合は TRUE を返します。それ以外の場合は FALSE。サンプルの使用方法については、オブジェクト データの処理 （617 ページ）を参照してください。 （注） 現在のクライアントに関連付けられていないリースの場合、最小限の情報しか使用できません。	
trace	$dict trace level message ...
DHCP サーバー パケット トレース システム内のメッセージを返します。レベル 0 では、トレースは行われません。レベル 1 では、サーバーがパケットを受信して応答を送信した場合にのみトレースされます。レベル4では、すべてをトレースします。残りの引数は連結され、指定されたレベルでトレース システムに送信されます。デフォルトのトレースは、DHCP サーバー拡張トレースレベル属性を使用して設定されます。	

表 *82 : Tcl* の *arg* 型と *obj* 型の値

arg 型または obj 型	説明
enterprise number/name	オプションまたはサブオプションのオプション定義セットのエンタープライズ ID 番号または名前。
home	コンテキストが現在のクライアントまたはリレー メッセージの "top" にリセットされることを要求します。
index番号/キーワード	操作対象の配列インデックスの番号またはキーワード (置換、追加、拡張、生、またはremove_all)。

DHCP 拡張ディクショナリ

arg 型または obj 型	説明
index-count	オプションの配列インデックスエントリの数を返します。
instance number	オプションのインスタンス番号 (主に DHCPv6 に使用)。
instance-count	オプションが表示された回数を返します (0 の場合、オプションは存在しません)。
moretcl 変数名	オプション データに配列インデックス エントリが存在するかどうかに応じて、TRUE または FALSE に設定されている Tcl 変数の名前。
move-to	コンテキストをオプションに設定することを要求します。
option number/name	操作するオプション番号または名前。
parent	コンテキストを 1 つ上に移動することを要求します。
suboption number/name	操作するサブオプション番号または名前。
vendor name	オプションまたはサブオプションのオプション定義セットのベンダー名。
lease initial | index | address | prefix	と共setObjectに使用すると、応答ディクショナリ内のリース、バインディング、およびプレフィックス データ項目のコンテキストを、指定されたリースに設定します。キーワードinitialは、拡張機能が呼び出されたときの元のコンテキストを復元することを要求します。インデックスは、番号付きリース(0から始まる)が設定され、クライアントのすべてのリースを反復処理するために使用できることを要求します。アドレスまたはプレフィックスは、そのアドレスまたはプレフィックスを持つリースが設定されることを要求します (存在する場合)。
message initial | number	と共setObjectに使用すると、メッセージ データ項目のコンテキストと、要求ディクショナリまたは応答ディクショナリ内のオプションを、指定されたメッセージに設定します。キーワードinitialは、クライアント メッセージにコンテキストを設定します。この番号は、リレー メッセージにコンテキストを設定し、0はクライアントに最も近いリレーを指定します。

arg 型または obj 型	説明
prefix initial |インデックス|住所|プレフィックス|名前	と共setObjectに使用すると、応答ディクショナリ内のプレフィックスデータ項目のコンテキストを、指定されたプレフィックスに設定します。initial キーワードは、拡張が呼び出されたときの元のコンテキストが復元されるよう要求します。インデックスは、番号付きプレフィックス(0 から始まる)が設定され、リンク上のクライアントのすべてのプレフィックスを反復処理するために使用できます。アドレスまたはプレフィックスは、アドレスまたはプレフィックスのプレフィックスが設定されることを要求します(見つかった場合)。名前付きプレフィックスが見つかっていることを要求します。現在のリンクのプレフィックスのみが使用できることに注意してください。

TCL 環境ディクショナリ メソッド

次の表は、環境ディクショナリで使用するコマンドを示しています。この場合、次の手順の例のように、dict変数を としてenviron定義できます。

```
proc tclhelloworld2 { request response environ } {
$environ put trace-level 4
$environ log LOG_INFO "Environment hello world"
}
```

表 *83 : TCL* 環境ディクショナリ メソッド

方法	構文
clear	$dict clear
ディクショナリからすべてのエントリを削除します。	
containsKey	$ディクト containsKeyキー
ディクショナリにキーが含まれている場合は 1 を返し、それ以外の場合は 0 を返します。	
firstKey	$dict firstKey
ディクショナリの最初のキーの名前を返します。キーは名前順に格納されません。キーが存在しない場合は、空の文字列を返します。	
get	$ディクト getキー
ディクショナリからキーの値を返します。キーが存在しない場合は、空の文字列を返します。	
isEmpty	$dict isEmpty
ディクショナリにエントリがない場合は 1 を返し、それ以外の場合は 0 を返します。	
log	$ディクスト logレベルメッセージ..

方法	構文
DHCP サーバー ロギング システムのメッセージを返します。レベルは、LOG_ERROR、LOG_WARNING、またはLOG_INFOのいずれかでなければなりません。残りの引数は連結され、指定されたレベルでロギングシステムに送信されます。 （注） サーバーはログ ファイルをこれらのレベルでログに記録するメッセージでフラッシュするので、LOG_ERROR レベルとLOG_WARNINGレベルを慎重に使用します。頻繁に発生しそうなメッセージ（クライアント要求など）に対してこれらのレベルを使用すると、ディスク I/O パフォーマンスに重大な影響を及ぼす可能性があります。	
nextKey	$dict nextKey
firstKey または nextKey への最後のコールで返されるキーに続くディクショナリ内の次のキーの名前を返します。キーが存在しない場合は、空の文字列を返します。	
put	$ディクト putキー値
キーに値を関連付けて、キーの既存のインスタンスを新しい値に置き換えます。	
remove	$ディクト removeキー
ディクショナリからキーを削除します。ディクショナリにキーが含まれていなくても、常に 1 を返します。	
size	$dict size
ディクショナリ内のエントリ数を返します。	
trace	$dict trace level message ...
DHCP サーバー パケット トレース システム内のメッセージを返します。レベル 0 では、トレースは行われません。レベル 1 では、サーバーがパケットを受信して応答を送信した場合にのみトレースされます。レベル4では、すべてをトレースします。残りの引数は連結され、指定されたレベルでトレース システムに送信されます。デフォルトのトレースは、DHCP サーバー拡張トレースレベル属性を使用して設定されます。	

DEX 属性ディクショナリ API

C/C++ 用の DEX 拡張機能を記述する場合、属性名文字列表現またはタイプ (属性を定義するバイトシーケンス) としてキーを指定できます。つまり、これらのアクセス方法の中には、キーまたは値の文字列または型の組み合わせである 4 つの異なるバリエーションがあります。

基本的な DEX 拡張の例は次のようになります。

```
int DEXAPI dexhelloworld( int iExtensionPoint,
dex_AttributeDictionary_t *pRequest,
dex_AttributeDictionary_t *pResponse,
dex_EnvironmentDictionary_t *pEnviron )
```

```
{
pEnviron->log( pEnviron, DEX_LOG_INFO, "hello world" );
return DEX_OK;
}
```

例については、インストールパス/例/dhcp/デックス/デックスエクステンション.cファイルまたはそのディレクトリ内の他のファイルを参照してください。

DEX の要求ディクショナリと応答ディクショナリ メソッド

DEX 属性ディクショナリでは、メソッドと呼ばれるアクティブなコマンドを使用して、値の変更やアクセスを行うことができます。次の表は、要求ディクショナリと応答ディクショナリで使用するメソッドを示しています。この場合、pDict変数を またはpRequestpResponseとして定義できます。

```
pRequest->get( pRequest, "host-name", 0, 0 );
```

pszAttributeは、const char *アプリケーションがアクセスする属性名へのポインターです。pszValueは、データを表const char *す文字列へのポインターです (getメソッドに対して返され、putメソッドに格納されます)。対応するiObjectType、iObjArgType、およびiArgTypeの各テーブルを参照してください。

ヒント

get、put、Option、Bytes、およびOptionBytes メソッドの違い （608 ページ）とget、put、remove、および ByType メソッドの違い （608 ページ）も参照してください。

表 *84 : DEX* の要求ディクショナリと応答ディクショナリ メソッド

方法	構文
allocateMemory	void *pDict->allocateMemory(dex_AttributeDictionary_t *pDict, unsigned int iSize)
この要求の有効期間だけ保持される拡張機能にメモリを割り当てます。	
get	constchar pDict pDict *　->get(dex_AttributeDictionary_t * , pszAttribute iIndex pbもっとconstchar*,abool_t*),int
ディクショナリから属性のiIndex edインスタンスの値を返します。ディクショナリに属性(またはその多くのインスタンス)が含まれていない場合は、代わりに空の文字列が返されます。pbMoreが 0 以外getの場合、返された属性のインスタンスが多い場合はpbMoreを TRUE に設定し、それ以外の場合は FALSE に設定します。この機能を使用して、属性の他のgetインスタンスを取得するために、 を別の呼び出しを行うかどうかを判断します。	
getBytes	const abytes_t *pDict->getBytes(dex_AttributeDictionary_t *pDict, const char *pszAttribute, int iIndex, abool_t *pbMore)

DHCP 拡張ディクショナリ

方法	構文
ディクショナリから属性のiIndex ed インスタンスの値をバイトシーケンスとして返します。ディクショナリに属性 (またはその多くのインスタンス) が含まれていない場合は、代わりに 0 を返します。pbMoreが 0 以外getBytesの場合、返された属性のインスタンスが多い場合は TRUE に設定され、それ以外の場合は FALSE に設定されます。この機能を使用して、属性の他のgetBytesインスタンスを取得するために、 を別の呼び出しを行うかどうかを判断します。	
getBytesByType	const abytes_t *pDict-> getBytesByType(dex_AttributeDictionary_t *pDict,const abytes_t *pszAttribute, int iIndex, abool_t *pbMore)
ディクショナリから属性のiIndex ed インスタンスの値をバイトシーケンスとして返します。ディクショナリに属性 (またはその多くのインスタンス) が含まれていない場合は、代わりに 0 が返されます。pbMoreが 0 以外の場合、返された属性のインスタンスが多い場合は TRUE を指定し、それ以外の場合は FALSE に設定します。この機能を使用して、属性の他のgetインスタンスを取得するために、 を別の呼び出しを行うかどうかを判断します。	
getByType	const char *pDict->getByType(dex_AttributeDictionary_t *pDict, const abytes_t *pszAttribute, int iIndex, abool_t *pbMore)
ディクショナリから属性のiIndex edインスタンスの値を返します。ディクショナリに属性(またはその多くのインスタンス) が含まれていない場合は、代わりに空の文字列を返します。pbMore が 0 以外の場合、getByType メソッドは、インスタンスが返された後にさらにインスタンスがある場合は pbMore に TRUE を設定し、それ以外の場合は FALSE を設定します。これを使用して、他のインスタンスを取得するためにgetByType別の呼び出しを行うかどうかを判断します。	
getOption	constchar * pディ*クスト iArgType .. getOption(dex_AttributeDictionary_t , , int)
オプションのデータを文字列として取得します。この関数は、常に文字列へのポインターを返しますが、オプションが存在しない場合や長さがゼロである場合は長さがゼロになる可能性があります。オプションが存在するかどうかを確認するには、DEX_INSTANCE_COUNTをgetOptionBytes 使用または指定します。	
getOptionBytes	const abytes_t * getOptionBytes(dex_AttributeDictionary_t *pDict , int iArgType, ...)
オプションのデータをバイトシーケンスとして取得します。この関数は、オプションが存在しない場合は null ポインターを返しabytes_t、オプションが存在するが長さが 0 バイトの場合は長さ 0 のバッファーを持つポインターを返します。	
getType	const abytes_t * pディプト***, const char****->getType(dex_AttributeDictionary_t*pszAttribute)
属性名が構成済みの属性と一致する場合は、属性を定義するバイト シーケンスへのポインターを返します。	

方法	構文
isValidisV4isV6	abool_t isValid(dex_AttributeDictionary_t *pディクスト ***)abool_t isV6(dex_AttributeDictionary_t*** *)abool_t isV4(dex_AttributeDictionary_t *pDict pDict)
(isValid渡されたディクショナリに応じて) 要求または応答がある場合、メソッドは TRUE を返します。それ以外の場合はFALSE。などの拡張機能はlease-state-change、ディクショナリが使用可能かどうかを判断するために、このメソッドを使用できます。 このisV4拡張が DHCPv4 パケットに対して呼び出されている場合、メソッドは TRUE を返します。それ以外の場合は FALSE。ルーチンからこのメソッドをinit-entry呼び出すと、FALSE が返されます。 このisV6拡張が DHCPv6 パケットに対して呼び出されている場合、このメソッドは TRUE を返します。それ以外の場合は FALSE。init-entry ルーチンからこのメソッドを呼び出すと、FALSE が返されます。	
log	abool_tpDict ->log(dex_AttributeDictionary_t * eLevel , pszFormat .., const char * , int)
DHCP サーバー ログ システムでメッセージを返します。eLevelは、DEX_LOG_ERROR、DEX_LOG_WARNING、またはDEX_LOG_INFOのいずれかでなければなりません。pszFormat は printf スタイルの書式指定文字列として扱われ、残りの引数と共に書式が設定され、指定されたレベルでログ 記録システムに送信されます。 （注） DEX_LOG_ERROR レベルとDEX_LOG_WARNINGレベルは、サーバーがログ ファイルにログ ファイルをフラッシュし、ログ レベルでログに記録されるため、慎重に使用します。これらのレベルを頻繁に発生する可能性のあるメッセージ (クライアント要求など) に使用すると、ディスク I/O パフォーマンスに重大な影響を与える可能性があります。	
moveToOption	abool_t moveToOption(dex_AttributeDictionary_t *pDict, int iArgType, ...)
後続getの 、 、 put、removeおよびオプションの操作のコンテキストを設定します。オプションが削除された場合 (例えば、 を使用removeOption) するとコンテキストが無効になる可能性があることに注意してください。	
put	abool_tpディ->put(dex_AttributeDictionary_tプト*・プディ,プトconst・プcharッツ*・プロパティ,const一・サイ・インデックスchar*,int)

DHCP 拡張ディクショナリ

方法	構文
サーバー構成のpszAttributeの定義に従って、pszValueをバイトシーケンスに変換します。そのバイト シーケンスをディクショナリ内の属性に関連付けます。iIndexが特殊な値DEX_REPLACE場合、属性の既存のインスタンスを 1 つの値に置き換えます。特殊な値がDEX_APPEND場合、属性の新しいインスタンスがリストに追加されます。特殊値がDEX_AUGMENT場合、属性が存在しない場合にのみ属性を設定します。それ以外の場合は、指定された位置に新しいインスタンスを挿入します。属性名が構成されている属性と一致しない場合、または値を有効な値に変換できなかった場合は TRUE を返します。	
putBytes	abool_tプディ->putBytes(dex_AttributeDictionary_tプト*・プディ,プトconst・プcharッツ*・プロパティ,const一・サイ・インデックスabytes_t*,int)
ディクショナリ内のpszAttributeにpszValueを関連付けます。iIndexがDEX_REPLACE特殊な値である場合は、属性の既存のインスタンスを 1 つの新しい値に置き換えます。特殊な値がDEX_APPEND場合、属性の新しいインスタンスをリストに追加します。特殊値がDEX_AUGMENT場合、属性が存在しない場合にのみ属性を設定します。それ以外の場合は、指定された位置に新しいインスタンスを挿入します。属性名が構成済みの属性名と一致しない場合は TRUE を返します。	
putBytesByType	abool_tプディ ->putBytesByType(dex_AttributeDictionary_tプト*・プディ,プトconst・プabytes_tッツ*・プロパティ,const一・サイ・インデックスabytes_t*,int)
ディクショナリ内のpszAttributeにpszValueを関連付けます。iIndexがDEX_REPLACE特殊な値である場合、属性の既存のインスタンスを新しい値に置き換えます。特殊な値がDEX_APPEND場合、属性の新しいインスタンスをリストに追加します。特殊値がDEX_AUGMENT場合、属性が存在しない場合にのみ属性を設定します。それ以外の場合は、指定された位置に属性の新しいインスタンスを挿入します。	
putByType	abool_tプディ ->putByType(dex_AttributeDictionary_tプト*・プディ,プトconst・プabytes_tッツ*・プロパティ,const一・サイ・インデックスchar*,int)
サーバー構成のpszAttributeの定義に従って、pszValueをバイトシーケンスに変換します。そのバイト シーケンスをディクショナリ内の属性に関連付けます。iIndexがDEX_REPLACE特殊な値である場合は、属性の既存のインスタンスを 1 つの新しい値に置き換えます。特殊な値がDEX_APPEND場合、属性の新しいインスタンスをリストに追加します。特殊値がDEX_AUGMENT場合、属性が存在しない場合にのみ属性を設定します。それ以外の場合は、指定された位置に新しいインスタンスを挿入します。	
putOption	abool_tputOption(dex_AttributeDictionary_tを*指定します。, const char * , int ,)

方法	構文
オプションとそのデータを追加するか、オプションのデータを変更します。	
putOptionBytes	abool_t putOptionBytes(dex_AttributeDictionary_t *pディクスト**, *int***, const abytes_t *pValueiArgType.. ,)
オプションとそのデータを追加する、またはオプションのデータを変更します。	
remove	abool_tpDict ->remove(dex_AttributeDictionary_t * pszAttribute, int iIndex, const char *)
ディクショナリから属性を削除します。iIndexがDEX_REMOVE_ALL特殊な値である場合は、属性の既存のインスタンスを削除します。それ以外の場合は、指定された位置にあるインスタンスを削除します。属性名が構成されている属性と一致しない場合は、ディクショナリがそのインデックスに属性を含んでいない場合でも TRUE を返します。	
removeByType	abool_tpDict ->removeByType(dex_AttributeDictionary_t * pszAttribute, int iIndex, const abytes_t *)
ディクショナリから属性を削除します。iIndexがDEX_REMOVE_ALL値の場合は、属性の既存のインスタンスを削除します。それ以外の場合は、指定された位置でインスタンスを削除します。ディクショナリにインデックスの属性が含まれていない場合でも、常に TRUE を返します。	
removeOption	abool_tremoveOption(iArgType 、.. dex_AttributeDictionary * , int)
オプションを削除します。DEX_INDEXを省略すると、DEX_REMOVE_ALLのDEX_INDEXが想定されます (これはオプション全体を削除します)。	
setObject	abool_tをsetObject(**指定します。** dex_AttributeDictionary_t * int int)
get、 、 およびputremoveメソッドのオブジェクトを設定し、新しいオプション メソッドが動作するメッセージを変更します。DHCPv6 拡張機能は、主にこのメソッドを使用して、クライアントとリンクで使用可能なリースおよびプレフィックスにアクセスしたり、リレー パケットからメッセージ ヘッダー フィールドまたはオプションを取得したりします。1 つのリースとスコープが応答に関連付けられている DHCPv4 とは異なり、DHCPv6 応答には複数のリースとプレフィックスが含まれる場合があります。オブジェクトが存在する場合はTRUEを返します。それ以外の場合はFALSE。サンプルの使用方法については、オブジェクトデータの処理 （617 ページ）を参照してください。 （注） 現在のクライアントに関連付けられていないリースの場合、最小限の情報しか使用できません。	
trace	abool_tpDict ->trace(dex_AttributeDictionary_t * iLevel , pszFormat .., const char * , int)

方法	構文
DHCP サーバー パケット トレース システム内のメッセージを返します。レベル 0 では、トレースは行われません。レベル 1 では、サーバーがパケットを受信して応答を送信した場合にのみトレースされます。レベル4では、すべてをトレースします。残りの引数は連結され、指定されたレベルでトレース システムに送信されます。デフォルトのトレースは、DHCP サーバー拡張トレースレベル属性を使用して設定されます。	

DEX 環境ディクショナリ メソッド

環境ディクショナリでは、メソッドと呼ばれるアクティブなコマンドを使用して、辞書の値を変更したりアクセスしたりできます。次の表は、環境ディクショナリで使用するメソッドを示しています。この場合、pDict変数をpEnvironとして定義できます。

```
pEnviron->log( pEnviron, DEX_LOG_INFO, "Environment hello world");
```

表 *85 : DEX* 環境ディクショナリ メソッド

方法	構文
allocateMemory	pディクスト ->allocateMemory(dex_EnvironmentDictionary_t*iSize, unsigned int void *)
この要求の有効期間だけ保持される拡張機能のメモリを割り当てます。	
clear	void pディクスト->clear(dex_EnvironmentDictionary_t * pDict)
ディクショナリからすべてのエントリを削除します。	
containsKey	abool_tpディ ->containsKey(dex_EnvironmentDictionary_tプト *pszKey, const char *)
ディクショナリにキーが含まれている場合は TRUE を返します。	
firstKey	const char *pDict->firstKey(dex_EnvironmentDictionary_t *pDict)
ディクショナリの最初のキーの名前を返します。キーは名前順に格納されません。キーが存在しない場合は、0 を返します。	
get	const char *pDict->get(dex_EnvironmentDictionary_t *pDict, const char *pszKey)
ディクショナリからキーの値を返します。キーが存在しない場合は、空の文字列を返します。	

方法	構文
isEmpty	abool_t pディクスト->isEmpty(dex_EnvironmentDictionary_t * pDict)
ディクショナリが 0 個のエントリを持つ場合は TRUE、それ以外の場合は FALSE を返します。	
log	abool_tpDict ->log(dex_EnvironmentDictionary_t * eLevel , pszFormat .., const char * , int)
DHCP サーバー ログ システムでメッセージを返します。eLevelは、DEX_LOG_ERROR、DEX_LOG_WARNING、またはDEX_LOG_INFOのいずれかでなければなりません。pszFormat は printf スタイルの書式指定文字列として扱われ、残りの引数と共に書式が設定され、指定されたレベルでログ 記録システムに送信されます。 （注） DEX_LOG_ERROR レベルとDEX_LOG_WARNINGレベルは、サーバーがログ ファイルにログ ファイルをフラッシュし、ログ レベルでログに記録されるため、慎重に使用します。これらのレベルを頻繁に発生する可能性のあるメッセージ (クライアント要求など) に使用すると、ディスク I/O パフォーマンスに重大な影響を与える可能性があります。	
nextKey	const char *pDict->nextKey(dex_EnvironmentDictionary_t *pDict)
最後の呼び出しで返されたキーの後に続くディクショナリ内の次のキーfirstKeyのnextKey名前を返します。キーが存在しない場合は、0 を返します。	
put	abool_tpDict ->put(dex_EnvironmentDictionary_t * pZKey, constpszValue char* , const char *)
キーの既存のインスタンスを新しい値に置き換えて、キーに値を関連付けます。	
remove	abool_t pDict->remove(dex_EnvironmentDictionary_t *pDict, const char *pszKey)
ディクショナリからキーと関連付けられた値を削除します。ディクショナリにキーが含まれていなくても、常に TRUE を返します。	
size	int pディクスト->size(dex_EnvironmentDictionary_t * pDict)
ディクショナリ内のエントリ数を返します。	
trace	abool_tpDict ->trace(dex_EnvironmentDictionary_t * iLevel , pszFormat .., const char * , int)

方法	構文
DHCP サーバー パケット トレース システム内のメッセージを返します。レベル 0 では、トレースは行われません。レベル 1 では、サーバーがパケットを受信して応答を送信した場合にのみトレースされます。レベル4では、すべてをトレースします。残りの引数は連結され、指定されたレベルでトレース システムに送信されます。デフォルトのトレースは、DHCP サーバー拡張トレースレベル属性を使用して設定されます。	

get、put、Option、Bytes、および OptionBytes メソッドの違い

次の DEX 拡張メソッドには、違いがあります。

- get および put
- getOption および putOption
- getBytes および putBytes
- getOptionBytes および putOptionBytes

メソッドgetとgetOptionメソッドは、文字列として書式設定された要求された情報を返します。サーバーは、ディクショナリ項目の予期されるデータ型に応じて、データを文字列に変換します。データ型が不明な場合、サーバーはデータを BLOB 文字列形式で返します。

メソッドgetBytesはgetOptionBytes、要求された情報を生のバイト (バッファーへのポインターとそのバッファーのサイズ) として返します。サーバーはこのバッファを読み取るだけで、オプションのデータだけが含まれている必要があります (たとえば、null 終端文字は追加されていません)。

putメソッドとputOptionメソッドは、データが書式設定された文字列として書き込まれると想定しています。サーバーは、ディクショナリ項目の予期されるデータ型に応じて、文字列からデータを変換します。データ型が不明な場合は、BLOB 文字列形式であることが想定されます。

サーバーはputBytes、未処理のバイトをメソッドputOptionBytes(バッファーへのポインターとそのバッファーのサイズ) に渡します。サーバーは、これらのバイトのみを読み取ります。

get、put、remove、および ByType メソッドの違い

次の DEX 拡張メソッドの間には違いがあります。

- get、put、および remove
- getByType、putByType、および removeByType

サーバーはget、 、put、およびremoveメソッドに、目的のデータ項目の名前を文字列として渡します。この場合、サーバーは文字列を内部データ テーブルにマップする必要があります。

サーバーは、文字列のgetByTypegetTypeメソッドputByTypeをremoveByType呼び出すことによって、サーバーが以前に取得した内部データ テーブル参照を渡します (拡張 init-entry など)。これにより、拡張機能の処理が高速化され、高いパフォーマンスを必要とするアプリケーションで重要な場合があります。

（注） getTypeメソッドが参照する内部データ テーブルは、要求または応答ディクショナリに対して要求されたかどうかにかかわらず同じです。同じデータ項目名にgetType対して、各ディクショナリで個別の呼び出しを行う必要はありません。

表 *86 :* 値

オブジェクトタイプ	説明
一般定義: コンテキストを変更するオブジェクト。	
DEX_LEASE	リース (およびプレフィックス) コンテキストを変更します。応答ディクショナリのみ。を使用して次のコマンドを実行します。 DEX_BY_IPV6ADDRESS DEX_BY_IPV6PREFIX DEX_BY_INSTANCE DEX_INITIAL
DEX_MESSAGE	メッセージコンテキストをリレーメッセージまたはクライアントメッセージに変更します。要求辞書と応答辞書。次の iObjArgType を使用できます。 DEX_INITIAL DEX_RELAY DEX_BY_NUMBER
DEX_PREFIX	プレフィックスコンテキストを変更しますが、リースコンテキストは変更しません。応答ディクショナリのみ。を使用して次のコマンドを実行します。 DEX_BY_IPV6ADDRESS DEX_BY_IPV6PREFIX DEX_BY_INSTANCE DEX_BY_NAME DEX_INITIAL

表 *87 :* 値

型	説明
一般定義: コンテキストが変更される方法。	

型	説明
DEX_BY_INSTANCE	DEX_LEASEまたは DEX_PREFIX iObjectTypeで使用します。インスタンス番号intを指定するために次の値を必要とします (0 から始まります)。利用可能なすべてのオブジェクトのリストを順を確認するために使用されますが、現在の要求または応答に適用可能なオブジェクトのリストを通してのみ使用 DEX_LEASEされます。DEX_PREFIXの場合、現在のリンクのプレフィックス (存在する場合)。DEX_RELAYの同義語であるDEX_MESSAGEで使用されます。
DEX_BY_IPV6ADDRESS	DEX_LEASEおよび DEX_PREFIX iObjectTypeでのみ使用されます。16 バイトのアドレスを指定するために、次のアドレスが必要です。const unsigned char *
DEX_BY_IPV6PREFIX	DEX_LEASE または DEX_PREFIX の iObjectType で使用します。17 バイトのプレフィックス バッファー (16 バイトのアドレスの後に 1 バイトのプレフィックス長) を指定する必要があります。 const unsigned char *
DEX_BY_NAME	DEX_PREFIX iObjectTypeと共にのみ使用されます。次の項目const char *を実行して、目的のオブジェクトの名前を指定する必要があります。
DEX_INITIAL	要求または応答の元のコンテキストに戻し、追加の引数はありません。エクステンションが最初に呼び出されたときのリースとプレフィックス (DEX_LEASE)、プレフィックス (DEX_PREFIX)、またはメッセージ (DEX_MESSAGE) を設定します(なしの場合があります)。
DEX_RELAY	iObjectTypeでのみ使用DEX_MESSAGE。リレーをint指定するために、次のことが必要です(0はクライアントに最も近いリレーを指定します)。メッセージコンテキストをクライアントに戻すには、setObject(pDict, DEX_MESSAGE, DEX_INITIAL)を使用します。

iArgType	説明
一般定義: コンテキストに従うアクションと引数。呼び出しには、iArgTypeインスタンスの数に任意の数を指定できます。	
DEX_ARG_ARRAY	引数リストを指定する代わりに、dex_OptionsArgs_t次の配列へのポインタを必要とします。各dex_OptionsArgs_t構造体には、次の 2 つのフィールドがあります。 • iArgType —このテーブルのiArgType DEX 値の 1 つ。 • pData—データ (整数)、データへのポインタ (文字列やその他のデータ型の場合)、または無視(iArgTypeが引数を取らない場合)。 サーバーが (引数リストまたは配列dex_OptionsArgs_t内の) DEX_ARG_ARRAYを検出すると、元のリスト内の後続の引数は無視されます。
DEX_END	（注） 必須で、追加の引数を持たない、引数リストの末尾をマークします。
DEX_ENTERPRISE_NAME	次に、オプション定義セット名を指定する必要があり、そこからサーバーがエンタープライズ ID を抽出してベンダー・オプション・データを取得します。 const char * ベンダー識別オプションに対してのみ有効です。ベンダーオプション定義セットが存在している必要があります。
DEX_ENTERPRISE_ID	ベンダーのintエンタープライズ ID を指定するには、次の手順を実行する必要があります。
DEX_HOME	コンテキストをクライアントまたはリレーメッセージ オプションに戻します。追加の引数はありません。常に成功を返します。使用する場合は、最初のiArgTypeである必要があります。有効なのはgetOption、 getOptionBytes、moveToOption 、および メソッドだけです。

iArgType	説明
DEX_INDEX	オプションintデータのインデックスを指定する必要があります(データの配列が処理される場合)。省略した場合、インデックス 0 は、DEX_REMOVE_ALLremoveOptionを除いてと見なされます。オプションデータ全体を取得、配置、または削除するには、特別な値DEX_RAWを使用します。ただし、DHCPv4ベンダー識別オプション(RFC 3925およびRFC 4243) では、DEX_RAWは 1 つのベンダー (インスタンスまたはエンタープライズIDに基づく) のデータのみを返し、オプション全体のデータを返しません。 特殊値DEX_RAWは、オプション(またはサブオプション) データ全体にアクセスします。データ型のデータ型と繰り返しカウントの観点からオプション定義で指定するデータに関係なく、データへの一貫したアクセスを提供します。データをデコードする汎用拡張機能に推奨されます。 DEX_REPLACE (値を置換する)、DEX_APPEND (endputOptionに追加putOptionBytesput)、およびDEX_AUGMENT (値が存在しない場合は追加) とメソッドを使用し、putByTypeメソッドputBytesはputBytesByType、、、、およびメソッドと同じように動作します。オプションを完全removeOptionに削除するには、[DEX_REMOVE_ALL を使用します。
DEX_INDEX_COUNT	結果として、intオプション データではなく、オプションのインデックス付きエントリの数を返す値が返されます。追加の引数を持たないため、DEX_INDEXやDEX_INSTANCE_COUNTには使用できません。DEX_END従わなければなりません。getOptionと でのみgetOptionBytes有効です。
DEX_INSTANCE	intオプションのインスタンスを指定する必要があります(複数のインスタンスを持つことができる DHCPv6 オプションでのみ有効)。0 は最初のインスタンスを指定します。

iArgType	説明
DEX_INSTANCE_COUNT	結果として、intオプション データではなく、オプションのインスタンス数のカウントを返す値が返されます。追加の引数を持たないため、DEX_INSTANCEで使用することはできません。DEX_END従わなければなりません。getOptionと でのみgetOptionBytes有効です。
DEX_MORE	フラグがabool_t書き込まれる場所を指定する必要*moreがあります。指定したインデックスを超えて配列項目が存在する場合、この場所DEX_INDEX TRUE に設定されます。メソッドとgetOptionメソッドgetOptionBytesに対してのみ有効です。
DEX_MOVE_TO	コンテキストは、DEX_MOVE_TO直前のオプションまたはサブオプションに置きます。これ以外の引数はありません。省略した場合、コンテキストは変更されません。データmoveToOptionを取得せずにコンテキストを移動するために使用します。getOption と getOptionBytes のメソッドに対してのみ有効です。 （注） 存在しないオプションまたはサブオプションに移動しようとすると、エラーが記録されます。拡張機能moveToOptionがオプションが存在することを以前に確認しなかった場合に使用します。
DEX_OPTION_NAME	必要なオプション名を指定するには、次の手順を実行する必要があります。 const char * オプション名はdhcpv4-config、または dhcpv6-configオプション定義セットに含める必要があります。
DEX_OPTION_NUMBER	必要なオプション名を指定するには、次の手順を実行する必要があります。 const char * オプション名はdhcpv4-config、または dhcpv6-configオプション定義セットに含める必要があります。

iArgType	説明
DEX_PARENT	コンテキストを親オプションに移動します。これ以外の引数はありません。クライアントメッセージまたはリレーメッセージを越えて移動せず、コンテキストが変更されない場合はFALSEを返します。使用する場合は、最初のiArgTypeである必要があります。有効なのはgetOption、 getOptionBytes、 moveToOption 、およびメソッドだけです。
DEX_SUBOPTION_NAME	その後に、目的のサブオプションの名前を指定する必要があります。 const char * サブオプションは現行オプション定義に含まれる必要があります。
DEX_SUBOPTION_NUMBER	必要なintサブオプション番号を指定するには、次の手順を実行する必要があります。定義が存在する必要はありませんが、サブオプション番号は現行のオプション定義に含まれている必要があります。ただし、サブオプションが存在しない場合は、データのバイト BLOB と見なされます。
DEX_VENDOR_NAME	ベンダー文字列を指定するには、次の手順を実行する必要があります。 const char * この文字列は、適切なオプション定義セットを検索するためだけに使用されます。

オブジェクトとオプションの処理

以下のセクションでは、拡張でDHCPオブジェクトとオプションを処理する特殊な方法について説明します。

オブジェクトとオプションの処理方法の使用

拡張機能は、DHCP オブジェクトを設定するメソッドを呼び出し、DHCP オプションの取得、移動、配置、および削除を行うことができます。メソッドはsetObject、 getOption、 moveToOptionputOption、 removeOption 、および Tcl および C/C++ のメソッドです。

これらの新しいコールバック メソッドは、主に DHCPv6 をサポートするために導入されました。ただし、DHCPv4 のオプション関連機能を使用できます。実際には、DHCPv4getでは、これらのメソッドは、元の [ ]Bytes、 get[ ]Bytes、ByTypeput[Bytes] 、 put[BytesByType]removeByType メソッドよりも豊富なオプションにアクセスできるため、DHCPv4 に使用することをお勧めします。

ヒント C/C++ でのこれらのメソッドの使用方法の違いについては、を参照してくださいDEX の要求ディクショナリと応答ディクショナリ メソッド （601 ページ）。

DHCPv6 の場合は、オプションsetObjectにgetOptionアクセスmoveToOptionするためにputOption、removeOption 、 、 、 、およびメソッドを使用する必要があります。このsetObjectメソッドは、拡張がアクセスする可能性がある多くのリース、プレフィックス、およびメッセージ (クライアントまたは複数のリレー) が存在する可能性があるため、DHCPv6 に導入されました。したがって、setObject要求と応答のディクショナリ データ項目とオプションを取得する後続の呼び出しのコンテキストを設定するのに役立ちます。サーバーが拡張機能を呼び出すと、コンテキストは現在のリース (該当する場合)、プレフィックス (該当する場合)、およびクライアント メッセージに設定されます。たとえば、サーバーが拡張ポイントをpre-packet-encode呼び出すと、要求および応答のディクショナリ メッセージ コンテキストのみが有効になり、この拡張ポイントに関連付けられたリースまたはプレフィックスがないため、対応するクライアント メッセージに設定されます。ただし、サーバーが拡張ポイントをlease-state-change呼び出すと、応答ディクショナリのリース コンテキストを状態が変更されたリースに設定し、応答ディクショナリ プレフィックス コンテキストをリースのプレフィックスに設定し、要求および応答ディクショナリ メッセージ コンテキストを対応するクライアント メッセージに設定します。

C/C++ のオプションとサブオプション

一部の C/C++ 拡張では、DHCP オプションとサブオプションを処理するための特殊な引数型の値が提供されます。DEX_OPTION_*引数タイプは、オプション (またはサブオプション) の下の定義ではなく、標準 DHCPv4 または DHCPv6 オプション定義セットを使用することを指定します。したがって、DEX_OPTION_*は、サーバーが標準 DHCPv4 または DHCPv6 オプション定義セット内のオプション名または番号を参照することを意味しますが、DEX_SUBOPTION_*は、サーバーが現行オプション定義のサブオプション名または番号(存在する場合) を参照することを意味します。

したがって、DHCPv6 でオプションにアクセスする場合、オプションがカプセル化されるときに、DEX_OPTION_*の後に DEX_OPTION_*を付けて使用することがよくあります。ベンダー オプションを調べるときは、DEX_SUBOPTIONを使用します。DHCPv4 の場合は、クライアント パケット レベルでDEX_OPTIONを使用し、ネスト レベルに応じて 1 回以上DEX_SUBOPTIONします。一般的に、エンタープライズ番号またはベンダー名を持つオプションのみが含まれていますが、このオプションは禁止されません。オプション定義セットは、何が有効かを決定します (ただし、定義を順に処理できますが、その時点ではすべてがバイナリバイトとして扱われるため、可能な限り制限され、オプション名またはサブオプション名を使用することはできませんが、数字を使用する必要があります)。

メソッドgetOptionmoveToOption、 、 、 、putOptionおよびremoveOptionメソッドのオプションの順序request付け規則は、式の構文に似ています。順序は一般的に次の要素で構成されます。

- 前文節 ([parent |home])
- オプション句option(vendor [|enterprise] [instance])
- サブオプション句suboption(vendor [| |enterprise] [instance])

- 終了句 ([|instance-countindex-count |[index] [more] end)

呼び出しは、前文節、ゼロ以上のオプション節、ゼロ以上のサブオプション節 (それ自体にオプションおよびサブオプション節が続く場合があります) を使用して、終了句を続けて作成できます。一部の処理getはinstance-count、 、 index-count、およびmoreなどのメソッドを通じてのみmove-to可能であり、コンテキストを現在のオプションまたはサブオプションに移動するために、任意の場所に表示できることに注意してください。

オプション定義によってデータ形式が決まりますが、これは、以前の関数が特定のオプションに対して返すデータ形式とは異なる場合があります。特定のオプションを処理するには、

- ベンダークラスのオプション (DHCPv4 の場合はv-i-vendor クラス[124]、DHCPv6 のベンダークラス[16]) では、オプションの特定のインスタンスを (エンタープライズ ID または名前ではなく) 要求する場合、エンタープライズ ID を取得する唯一の方法は、生データを要求する (DEX_RAW を使用するDEX_INDEX)。
- DHCPv4 ベンダーオプション(v-i-vendor-class[124] およびvv-i-vendor-opts[125]) では、未処理データ (DEX_RAWを使用したDEX_INDEX) に対する操作は、オプション全体ではなく、そのオプションのインスタンス (プリセット値 0) にのみ適用されます。このオプションputOptionのデータ全体を取得する方法はありません。DHCPv6 ベンダー・オプションは個別のオプションであるため、これは問題ではありません。
- DHCPv4 ベンダー オプション (124 または 125) の 1 つが正しく書式設定されていない場合、データ全体が BLOB として返されます (インスタンス 0 を求めて特定のエンタープライズ ID を指定しなかった場合)。ただし、操作によっては拡張機能を使用putOptionしようとすると、そのデータが既存のデータに追加され、結果の形式が正しく設定されません。
- putOption(pDict,"01:02",DEX_OPTION_NUMBER,124,DEX_END)ベンダーオプションの場合、オプションがない場合、enterprise-id が使用できないために失敗します。putOption(pDict, "00:00:00:09:04:03:65:66:67", DEX_OPTION_NUMBER, 124, DEX_END)ただし、00:00:00:09 がエンタープライズ ID であり、04 で始まるバイトが、そのエンタープライズ ID のオプション データの長さであるために機能します。この場合、長さバイトが検証されputOption、正しい長さがない場合は失敗します。データ追加の推奨方法は、putOption(pDict, "65:66:67", DEX_OPTION_NUMBER, 124, DEX_ENTERPRISE_ID, 9, DEX_END) を使用することです。

オプションとオブジェクトのメソッド コールの例

これらのセクションでは、DHCP オプションとオブジェクトデータを処理するメソッドの使用方法の例をいくつか紹介します。

ベンダー クラス オプション データの処理

DHCPv4 の場合、クライアントへの応答に 2 つのエンタープライズ ID のベンダー識別ベンダクラス オプション (124) データを含めるには、次の方法putOptionを使用するいくつかのサンプル Tcl コードを示します。

```
$response putOption 65:66:67 option 124 enterprise 999998
```

```
 #adds "abc" (65:66:67) under enterprise-id 999998
$response putOption 68:69:6a:6b option v-i-vendor-class enterprise 999998 index append
 #appends "defg" (68:69:6a:6b) under the same enterprise-id
$response putOption 01:02:03:04 option 124 enterprise 999999
 #adds 01:02:03:04 under enterprise-id 999999
```

オプションを取得するには、次のgetOption方法を使用します。

```
$response getOption option v-i-vendor-class instance-count
 #returns 2 because there were two instances added (enterprise id 999998 and enterprise
 id 999999)
$response getOption option 124
 #returns index 0 of instance 0, which is 65:66:67
$response getOption option 124 index-count
 #returns 2 because there were two vendor classes added for the first enterprise id
(9999998)
$response getOption option 124 index raw
 #returns 00:0f:42:3e:09:03:65:66:67:04:68:69:6a:6b for the complete encoding of the
enterprise-id 999998 data (see RFC 3925)
$response getOption option 124 index 1
 #returns 68:69:6a:6b
$response getOption option 124 instance 1 index-count
 #returns 1 because there is only one vendor class
$response getOption option 124 instance 1 index raw
 #returns 00:0f:42:3f:05:04:01:02:03:04 for the complete encoding of the enterprise-id
999999 data (see RFC 3925)
$response getOption option 124 enterprise 999999
 #returns 01:02:03:04
```

データを削除するには、2removeOptionつの個別のエンタープライズ ID があるため、2 つの呼び出しが必要です。

```
$response removeOption option 124
$response removeOption option 124
```

オブジェクト データの処理

pre-packet-encode拡張ポイントで、クライアントのすべてのリースのデータを抽出するとします。このメソッドを使用する TclsetObjectコードのサンプルを次に示します。

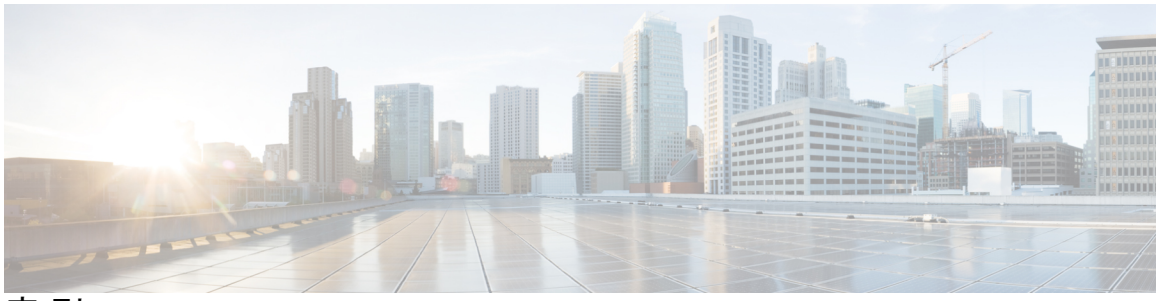
```
proc logleasesinit { request response environ } {
   if { [$environ get "extension-point"] == "initialize" } {
     # Set up for DHCPv6 only]
     $environ put dhcp-support "v6"
     $environ put extension-extensionapi-version 2
  }
}
proc logleases { request response environ } {
   for { set i 0 } { 1 } { incr i } {
     # Set context to next lease
     if { ![$response setObject lease $i] } {
     # Lease does not exist, so done
     break
  }
  # Log the lease address, prefix name, and prefix address
  $environ log LOG_INFO "Lease [$response get lease-ipaddress], Prefix\
     [$response get lease-prefix-name] - [$response get prefix-address]"
  }
```

DHCP 拡張ディクショナリ

```
  # Restore the lease context to where we started
    $response setObject lease initial
    # Do other things...
}
```

これに対する C++ と同等のコードは次のようになります。

```
// Print the current leases for the client
for( int i=0; ; i++ ) {
   if( !pRes->setObject( pRes, DEX_LEASE, DEX_BY_INSTANCE, i ) )
      break;
   const char *pszLeaseAddress =
      pRes->get( pRes, "lease-ipaddress", 0, 0 );
   if( pszLeaseAddress == 0 )
      pszLeaseAddress = "<error>";
   const char *pszPrefixName =
      pRes->get( pRes, "prefix-name", 0, 0 );
   if( pszPrefixName == 0 )
      pszPrefixName = "<error>";
   pEnv->log(pEnv, DEX_LOG_INFO,
     "Lease %s, Prefix %s",
   pszLeaseAddress, pszPrefixName );
}
```



索 引

数字

A

B

C

D

E

F

G

I

K

L

M

N

P

R

S

T

V

W

Z

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

な

に

ぬ

ね

は

ひ

ふ

へ

ほ

ま

み

め

も

ゆ

り

る

れ

ろ