ゼロタッチ プロビジョニング

ここでは、次の内容について説明します。

ゼロタッチプロビジョニングの概念

Cisco Crosswork Zero Touch Provisioning（ZTP）アプリケーションを使用すると、ネットワーキングデバイスをリモートでプロビジョニングできます。工場出荷時の状態のデバイスをブランチオフィスまたはリモートサイトに出荷できます。ローカルオペレータは、イメージをインストールしたり、設定したりすることなく、これらのデバイスをネットワークにケーブル接続できます。ZTP を使用するには、まず DHCP サーバーと ZTP アプリケーションで各デバイスのエントリを確立します。その後、デバイスをネットワークに接続して電源を投入するか、リロードすることで、ZTP 処理をアクティブ化できます。ZTP は、認定されたイメージと 1 つ以上の設定を自動的にダウンロードしてデバイスに適用します（設定のみを適用することもできます）。設定が完了すると、ZTP は新しいデバイスを Cisco Crosswork デバイスインベントリにオンボーディングします。その後、他の Cisco Crosswork アプリケーションを使用して、デバイスをモニターおよび管理できます。

Cisco Crosswork ZTP では、次の基本用語と概念を使用します。

- **クラシック ZTP**：ソフトウェアと設定ファイルをダウンロードしてデバイスに適用するプロセス。iPXE ファームウェアと HTTP を使用してデバイスを起動し、ダウンロードを実行します。パブリックネットワークでの使用には適していません。
- **セキュア ZTP**：ソフトウェアイメージと設定ファイルをダウンロードしてデバイスに適用するセキュアなプロセス。セキュアなトランスポートプロトコルと証明書を使用してデバイスを検証し、ダウンロードを実行します。
- **評価ライセンスのカウントダウン**：ZTP を使用してオンボードされたデバイスのライセンスには、通常 90 日間の評価期間があります。Cisco Crosswork は、評価期間中、カウントダウンバナーを表示します。評価期間が終了するまでに、ライセンスのプールを購入する

ようにしてください。有効期限が切れると、購入したライセンスを適用するまで、Cisco Crossworkは警告バナーを表示し、新しいデバイスのオンボーディングをブロックします。

- **イメージファイル**：デバイスにネットワーク オペレーティング システムをインストールするために使用するバイナリ ソフトウェアイメージファイル。シスコのデバイスの場合、これらのファイルはCisco IOS-XR イメージのサポートされているバージョンです。これを行うように設定すると、クラシック ZTP プロセスは Cisco Crosswork からイメージをダウンロードし、オープンソースのブートファームウェア iPXE を使用してインストールします。SMU をインストールする必要がある場合、ZTP は設定処理の一部としてそれらを適用します。

- **設定ファイル**：新しくイメージ化されたデバイスや再イメージ化されたデバイスの動作パラメータを設定するために使用するファイル。ファイルには、Python スクリプト、Linux シェルスクリプト、または ASCII テキストとして保存された一連の Cisco IOS CLI コマンドを使用できます。ZTPプロセスは、新しくイメージ化されたデバイスに設定ファイルをダウンロードし、実行します。ZTP 処理には設定ファイルが必要です。

- **クレデンシャルプロファイル**：SNMP、SSH、HTTP、およびその他のネットワークプロトコルを介してデバイスにアクセスするために使用するパスワードとコミュニティ文字列の集まり。Cisco Crossworkは、クレデンシャルプロファイルを使用してデバイスにアクセスし、デバイスアクセスを自動化します。すべてのクレデンシャルプロファイルは、パスワードとコミュニティ文字列を暗号化形式で保存します。

- **ブートファイル名**：ZTP リポジトリに保存されているソフトウェアイメージの明示的なパスと名前。ZTP を使用してオンボーディングする予定のデバイスごとに、DHCP のデバイス設定の一部としてブートファイル名を指定します。

- **HTTPS/TLS**：Hypertext Transport Protocol Secure（HTTPS）は、HTTP プロトコルのセキュアな形式です。暗号化したレイヤで HTTP をラップします。このレイヤは Transport Layer Security（TLS）（以前の Secure Sockets Layer、つまり SSL）です。

- **iPXE**：オープンソース ブート ファームウェア iPXE は、ブート前実行環境（PXE）クライアントファームウェアとブートローダの一般的な実装です。 iPXE を使用すると、組み込み PXE サポートのないデバイスをネットワークから起動できます。iPXE ブートプロセスは、クラシック ZTP 処理の一部であり、セキュア ZTP 処理の一部ではありません。ただし、オンサイトの技術者は、引き続き iPXE ブートを強制してからセキュア ZTP 処理を開始できます。

- **所有者証明書**：組織の CA 署名入りのエンドエンティティ証明書。公開キーを組織にバインドします。デバイスに所有者証明書をインストールします。

- **所有権バウチャー**：ZTP でオンボーディングされているデバイスが、組織が所有するドメインにブートストラップされていることを確認するナンスレス監査バウチャー。シスコは、組織からの要求に応じて OV を提供します。

- **PDC**：ピン留めドメイン証明書（PDC）は、組織の CA または自己署名ドメイン証明書です。PDC の公開キーは、組織に割り当てられた DNS ネットワークドメインに PDC を固定します。PDC（ピン留めドメイン証明書）は、ZTP の処理中にダウンロードおよび適用さ

れたイメージと設定が組織内からのものであることをデバイスが確認する際に役立ちます。

- **SUDI**：セキュアな一意のデバイス識別子（SUDI）は、関連付けられたキーペアを持つ証明書です。SUDI には、製品識別子とシリアル番号が含まれています。シスコは製造時に SUDI とキーペアをデバイスハードウェアのトラストアンカーモジュール（TAm）に挿入し、デバイスにイミュータブル ID を付与します。セキュア ZTP 処理時に、バックエンドシステムはデバイスにアイデンティティの検証を要求します。ルータは SUDI ベースのアイデンティティを使用して応答します。このやり取りと TAm 暗号化サービスにより、バックエンドシステムは暗号化されたイメージと設定ファイルを提供できます。これらの暗号化されたファイルを開くことができるのは、特定のルータだけです。これにより、パブリックネットワーク上での転送の機密性が確保されます。
- **SUDI ルート CA 証明書**：認証局（CA）によって発行および署名され、下位の SUDI 証明書を認証するために使用する SUDI のルート認証証明書。
- **UUID**：汎用一意識別子（UUID）は、Cisco Crosswork にアップロードしたイメージファイルを一意に識別します。DHCP ブートファイル URL にソフトウェアイメージファイルの UUID を使用できます。UUID は設定ファイルには必要ありません。
- **ZTP アセット**：ZTP では、新しいデバイスをオンボーディングするために、いくつかのタイプのファイルと情報にアクセスする必要があります。これらのファイルと情報を総称して「ZTP アセット」と呼びます。ZTP 処理を開始する前に、ZTP 設定の一部としてこれらのアセットをロードします。
- **ZTP プロファイル**：（通常は）1 つのイメージと 1 つの設定を 1 つのユニットに結合する Cisco Crosswork ストレージ構成。Cisco Crosswork は、ZTP プロファイルを使用して、イメージ化プロセスと設定プロセスを自動化します。ZTP プロファイルの使用は任意ですが、推奨されています。これらは、デバイスファミリ、クラス、およびロールに関する ZTP イメージと設定の整理を簡単にし、ZTP の使用に一貫性を持たせるのに役立ちます。
- **ZTP リポジトリ**：Cisco Crosswork が ZTP イメージと設定ファイルを保存する場所。

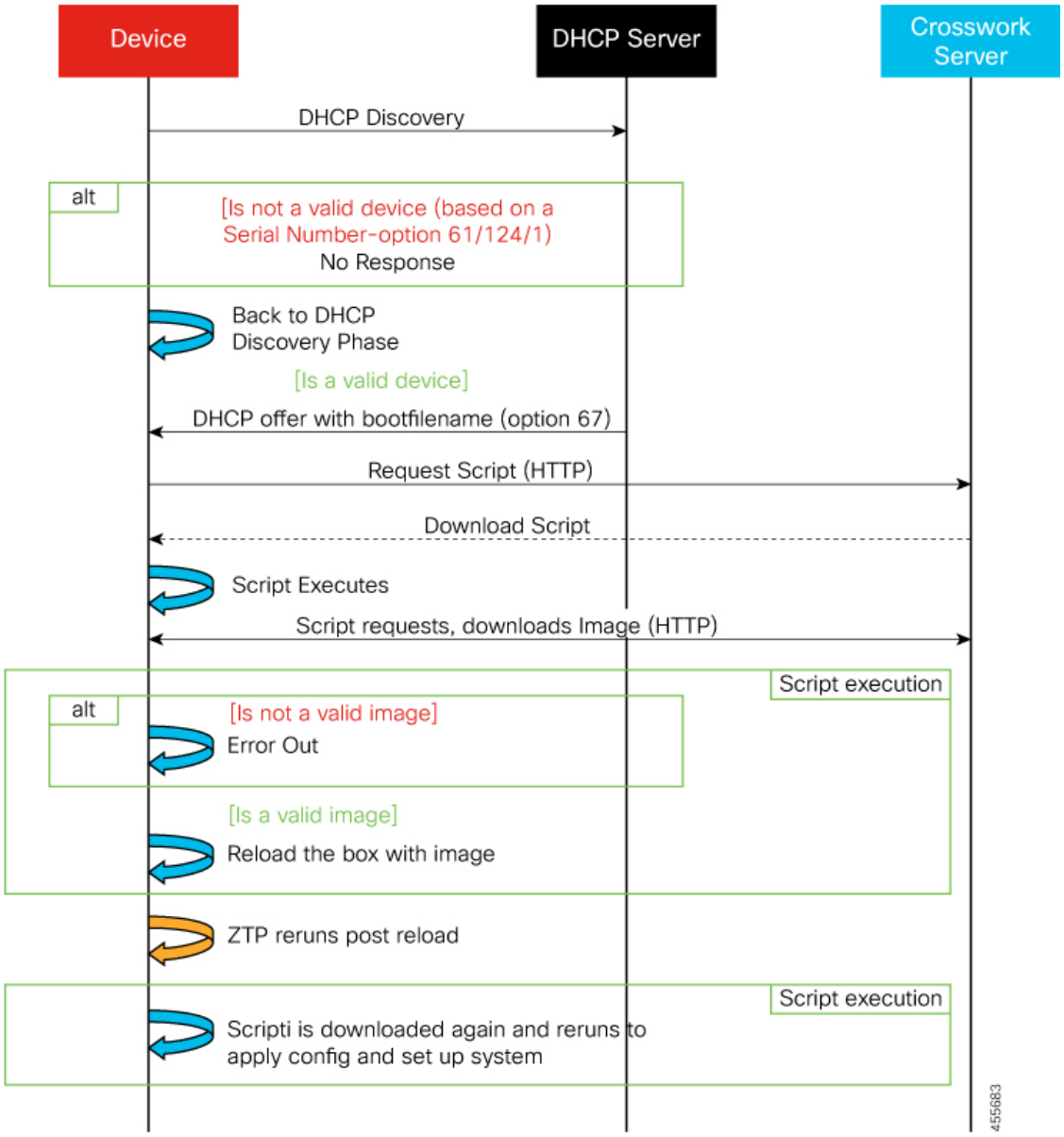
ZTP の処理ロジック

Cisco Crosswork ZTP の処理は、クラシック ZTP またはセキュア ZTP のいずれを実装するかによって異なります。

クラシック ZTP のロジック

次の図に、クラシック ZTP がデバイスのプロビジョニングとオンボーディングに使用する処理ロジックを示します。DHCP サーバーは、デバイスのシリアル番号に基づいてデバイスのアイデンティティを確認してから、ブートファイルとイメージのダウンロードを提供します。ZTP がデバイスをイメージ化すると、デバイスは設定ファイルをダウンロードし、実行します。

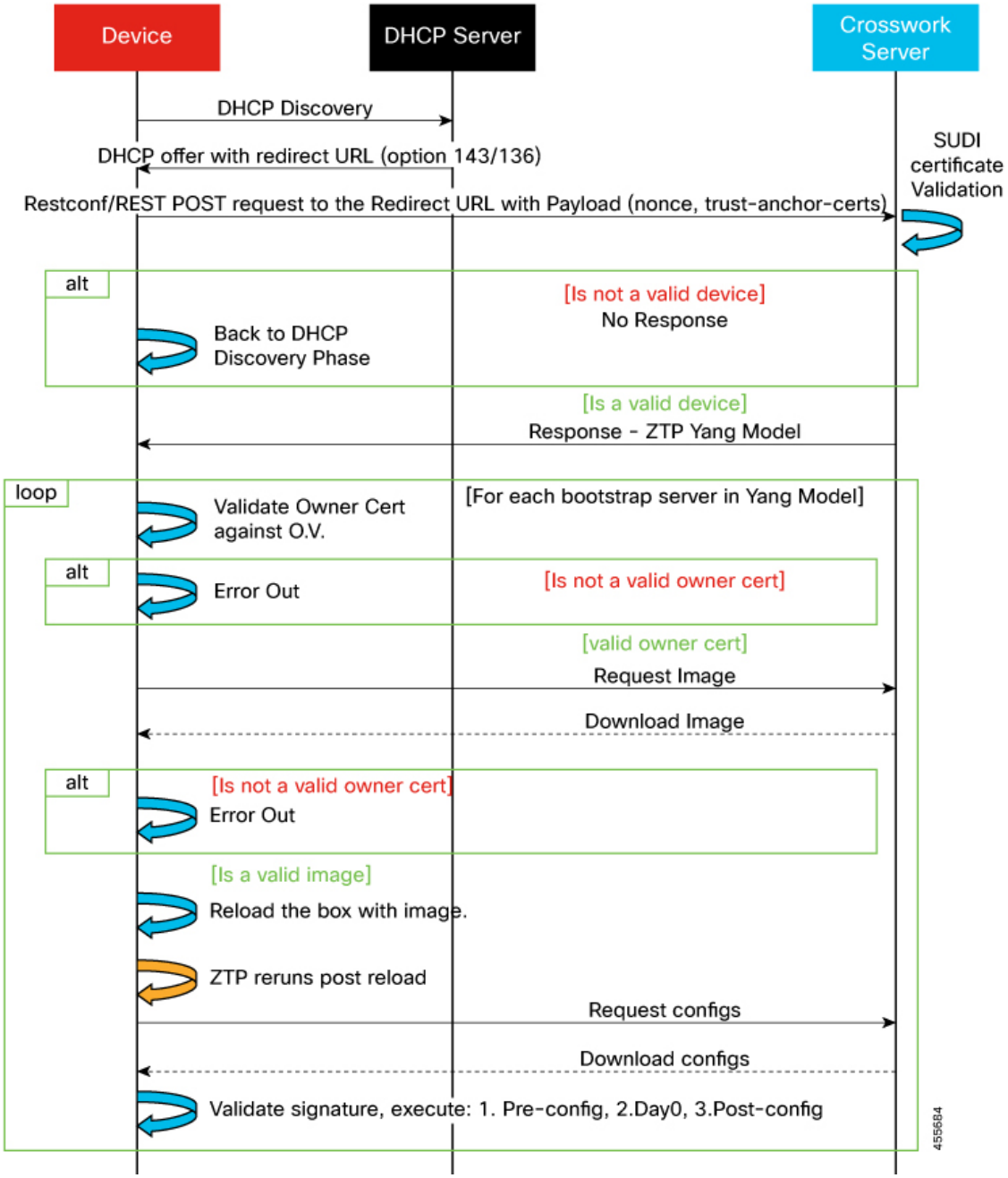
図 *1*: クラシック *ZTP* の処理ロジック



セキュア ZTP のロジック

次の図に、セキュア ZTP がデバイスのプロビジョニングとオンボーディングに使用するプロセスロジックを示します。デバイスと ZTP ブートストラップサーバーは TLS/HTTPS を介してデバイスとサーバー証明書でセキュアな一意のデバイス識別子（SUDI）を使用し、相互に認証します。セキュアな HTTPS チャネルを介して、ブートストラップサーバーはデバイスに署名付きイメージと設定アーティファクトをダウンロードさせます。これらのアーティファクトは、RFC 8572 YANG スキーマに準拠する必要があります。デバイスは新しいイメージ（存在する場合）をインストールしてリロードすると、設定スクリプトをダウンロードして実行します。

図 *2 :* セキュア *ZTP* の処理ロジック



ZTP の状態遷移

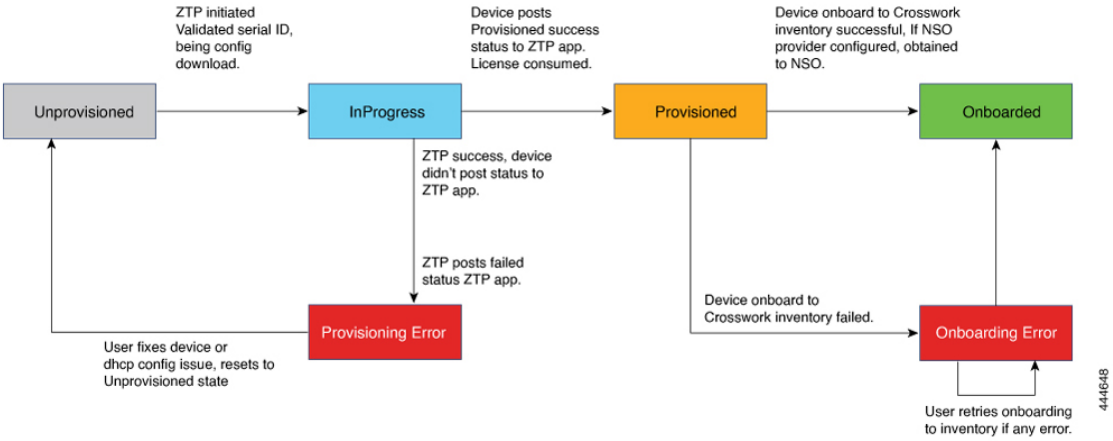
デバイスのリセットまたはリロードによって開始されると、ZTP プロセスは自動的に進行します。また、Cisco Crosswork は、[ゼロタッチデバイス（Zero Touch Devices）] ウィンドウを更新し、各デバイスが到達したプロセスの段階を示すステータスメッセージも表示します。次の 2 つの項で説明するように、状態とその遷移は、クラシック ZTP とセキュア ZTP で異なります。

ZTP で使用する設定スクリプトは、Cisco API コールを使用して、デバイスの状態変化を Cisco Crosswork に報告する必要があります。そうしないと、Cisco Crosswork は状態変化が発生したときにそれを登録できず、プロビジョニングとオンボーディングに失敗します。これらのコールの例を確認するには、**[デバイス管理（Device Management）] > [ZTP 設定ファイル（ZTP Configuration Files）]** を選択し、[サンプルスクリプトのダウンロード（Download Sample Script）] をクリックします。

クラシック ZTP の状態遷移

次の図に、クラシック ZTP 処理の状態変化を示します。

図 *3:* クラシック *ZTP* デバイスの状態遷移



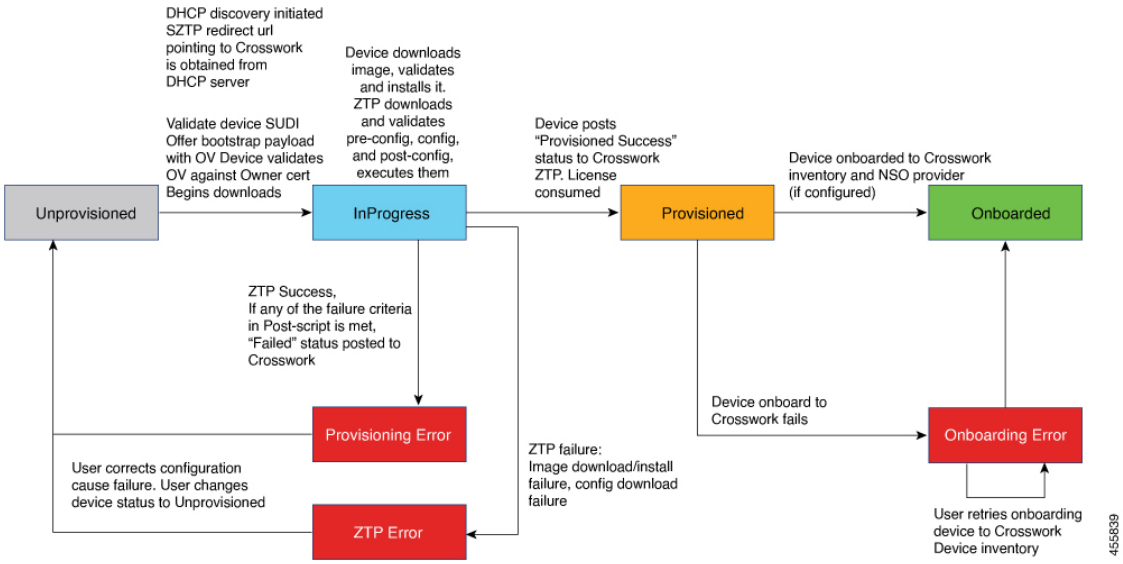
クラシック ZTP デバイスエントリは、[プロビジョニングなし（Unprovisioned）]状態から開始されます。ZTP を開始すると、デバイスはネットワークに接続し、イメージとコンフィギュレーションファイルのダウンロードを開始すると、[進行中（InProgress）] の状態に移行します。デバイスは、[プロビジョニングエラー（Provisioning Error）] の発生、または [プロビジョニング済み（Provisioned）] を報告するまで、[進行中（InProgress）] の状態が維持されます。プロビジョニングが成功すると、デバイスは [プロビジョニング済み（Provisioned）] の状態に移行します。プロビジョニングが完了すると、Cisco Crosswork はデバイスをオンボーディングします。Cisco NSO が Cisco Crosswork プロバイダである場合、Cisco NSO はデバイスもオンボーディングします。オンボーディングが成功すると、デバイスの状態が [オンボーディング済み（Onboarded）] に変わります。これでデバイスがインベントリに組み込まれたため、他の Cisco Crosswork ネットワークデバイスと同様にモニターおよび管理できます。

クラシック ZTP は、デバイスがそのイメージや設定コードを正常にロードし、Cisco Crosswork に接続して、[プロビジョニング済み（Provisioned）] のステータスを報告すると成功します。このステータスの変化により、そのデバイスのシリアル番号に対して 1 つのライセンスがカウントされます。ライセンスはシリアル番号に関連付けられているため、後で [オンボーディング済み（Onboarded）] の状態に移行したり、または ZTP 処理をさらに行ったりしても、ライセンス数には影響しません。

セキュア ZTP の状態遷移

次の図に、セキュア ZTP 処理の状態変化を示します。

図 4 : セキュア *ZTP* の状態遷移



セキュア ZTP デバイスのエントリは、[プロビジョニングなし（Unprovisioned）] の状態で始まります。ZTP を開始すると、デバイスとブートストラップサーバーが相互に検証し、ペイロードを検証します。この 2 つは、デバイスの SUDI 、所有権バウチャー、およびデバイス所有者証明書を使用し、HTTP/TLS を介して検証を行います。検証後、デバイスエントリはネットワークに接続し、イメージと設定ファイルのダウンロードを開始すると、[進行中（InProgress）] の状態に移行します。デバイスは、[プロビジョニングエラー（Provisioning Error）]、[ZTP エラー（ZTP Error）]、または [プロビジョニング済み（Provisioned）] のステータスを Cisco Crosswork に通知するまで、[進行中（InProgress）] の状態のままになります。プロビジョニングが成功すると、デバイスは [プロビジョニング済み（Provisioned）] の状態に移行します。プロビジョニングが完了すると、Cisco Crosswork はデバイスをオンボーディングします。Cisco NSO が Cisco Crosswork プロバイダである場合、Cisco NSO はデバイスもオンボーディングします。オンボーディングが成功すると、デバイスの状態が [オンボーディング済み（Onboarded）] に変わります。これでデバイスがインベントリに組み込まれたため、他の Cisco Crosswork ネットワークデバイスと同様にモニターおよび管理できます。

検証手順のいずれかが失敗すると、セキュア ZTP は [プロビジョニングエラー（Provisioning Error）] を通知します。イメージまたは設定コードが検証またはインストールに失敗すると、セキュア ZTP は代わりに [ZTP エラー（ZTP Error）] を通知します。クラシック ZTP と同様に、セキュア ZTP は、デバイスがそのイメージや設定コードを正常にロードし、Cisco Crosswork に接続して、[プロビジョニング済み（Provisioned）] のステータスを通知すると成功します。ライセンスの消費量は、クラシック ZTP と同じです。

ZTP と評価ライセンス

すべてのライセンスは、90 日間の評価期間から始まります。評価期間が終了すると、Cisco Crosswork は、評価ライセンスの期限が切れたことをユーザーに警告するバナーを表示します。ZTP はこのバナーを表示しますが、構成のダウンロードを含む一部の操作をブロックします。組織がスマートライセンスに登録し、一部のオンボードデバイスにライセンスを適用すると、ZTP はブロックを削除します。ZTP は、すべてのオンボードデバイスのライセンスを取得するまで、警告バナーを表示します。

オンボーディング済みの ZTP デバイスは、常に次のいずれかに関連付けられます。

- シリアル番号、または
- Option 82 ロケーション ID 属性の値（リモート ID と回線 ID）。

シリアル番号とロケーション ID によって「許可」リストが形成されます。ZTP は、デバイスをオンボーディングしてライセンスを割り当てることを決定するときに、このリストを使用します。オンボーディング済みの ZTP デバイスをインベントリから削除し、後で再度オンボーディングする場合は、同じシリアル番号またはロケーション ID を使用します。別のシリアル番号やロケーション ID を使用すると、ライセンスが余分に消費される場合があります。現在のリリースでは、このシナリオの回避策は提供されていません。いずれの場合も、同じシリアル番号またはロケーション ID を持つ 2 つの異なる ZTP デバイスを同時にアクティブにすることはできません。

ZTP でのプラットフォームサポート

このトピックでは、シスコ製とサードパーティ製のソフトウェアおよびデバイスに対する Cisco Crosswork Zero Touch Provisioning のサポートについて詳しく説明します。

クラシック ZTP でのプラットフォームサポート

次のプラットフォームは、クラシック ZTP をサポートしています。

- **ソフトウェア**：Cisco IOS-XR バージョン 6.6.3、7.0.1、7.0.2、7.0.12、7.3.1 以降。
- **ハードウェア**：
  - Cisco Network Convergence Systems（NCS）540 シリーズ ルータ
  - Cisco NCS 1000-1004 シリーズ ルータ
  - Cisco NCS 5500 シリーズ ルータ
  - Cisco NCS 8000 および 8800 シリーズ ルータ（Spitfire 固定モード）

クラシック ZTP は、サードパーティ製のデバイスまたはソフトウェアをサポートしていません。

セキュア ZTP でのプラットフォームサポート

次のプラットフォームでセキュア ZTP がサポートされています。

- **ソフトウェア**：Cisco IOS-XR バージョン 7.3.1 以降。

  単一イメージのインストールとして、IOS-XR 6.6.3 から 7.3.1 にアップグレードできます。

- **ハードウェア**：
  - Cisco Network Convergence Systems（NCS）540 シリーズ ルータ
  - Cisco NCS 1000-1004 シリーズ ルータ
  - Cisco NCS 5500 シリーズ ルータ
  - Cisco NCS 8000 および 8800 シリーズ ルータ（Spitfire 固定モード）

セキュア ZTP は、サードパーティ製デバイスのプロビジョニングをサポートしています。

- Secure ZTP RFC 8572（https://tools.ietf.org/html/rfc8572）に 100% 準拠していること。
- デバイス証明書と所有権バウチャーのシリアル番号がシスコ形式のガイドラインと一致していること。詳細については、次のセクション「サードパーティ製デバイス証明書および所有権バウチャーのガイドライン」を参照してください。

サードパーティ製デバイス証明書および所有権バウチャーのガイドライン

デバイスのセキュア ZTP 処理は、デバイスと Cisco Crosswork 間の正常な HTTPS/TLS ハンドシェイクから始まります。ハンドシェイク後、セキュア ZTP はデバイス証明書からシリアル番号を抽出する必要があります。セキュア ZTP は、抽出したシリアル番号を内部のシリアル番号の「許可」リストと照合して検証します。許可リストを作成するには、デバイスのシリアル番号を Cisco Crosswork にアップロードします。所有権バウチャーを使用してダウンロードを検証する場合も、同様のシリアル番号検証手順が後で実行されます。

Cisco IOS-XR デバイスとは異なり、サードパーティベンダーのデバイス証明書のシリアル番号の形式はベンダー間で標準化されていません。通常、サードパーティベンダーのデバイス証明書には、`Subject` フィールドまたはセクションがあります。`Subject` には、ベンダーが決定する複数のキーと値のペアが含まれます。通常、キーと値のペアの 1 つは `serialNumber` キーです。このキーの値には、実際のデバイスのシリアル番号が文字列として含まれます。その前には、文字列 `SN:` が付きます。たとえば、サードパーティのデバイス証明書の `Subject` セクションに `serialNumber = PID:NCS-5501 SN:FOC2331R0CW` というキーと値が含まれているとします。セキュア ZTP は `SN:` 文字列の後の値を取得し、その値を許可リスト内のシリアル番号の 1 つと照合します。

サードパーティベンダーのデバイス証明書の形式が異なると、検証エラーが発生する可能性があります。障害の程度は、差異の程度によって異なります。ベンダー証明書がこの形式とまったく一致しない場合があります。証明書の `Subject` フィールドに、`SN:` 文字列を含む値を持つ `serialNumber` キーを含めることはできません。この場合、セキュア ZTP の処理は、デバイスのシリアル番号として `serialNumber` キーの文字列値全体（存在する場合）を使用するようにフォールバックします。次に、その値をシリアル番号の許可リストの 1 つと照合します。この

2つの方法（文字列照合とフォールバック）は、セキュア ZTP がサードパーティ製デバイスのシリアル番号を判別するための唯一の手段です。ベンダー証明書がこの想定と大幅に異なる場合、セキュア ZTP はデバイスをまったく検証できない可能性があります。

セキュア ZTP では、所有権バウチャーに対して同様の形式が想定されます。シスコのツールは、*SerialNumber*.vcj 形式のファイル名で所有権バウチャーを生成します。ここで、*SerialNumber* はデバイスのシリアル番号です。セキュア ZTP は、ファイル名からシリアル番号を抽出し、許可リスト内のいずれかの番号との照合を試みます。マルチベンダーサポートでは、サードパーティベンダーのツールにより同じ形式の OV ファイルが生成されると想定しています。この想定が満たされない場合は、検証が失敗する可能性があります。

ZTP の実装の決定

ZTP には実装のさまざまな選択肢があり、コスト対メリットのトレードオフを事前に検討に値します。

- **クラシック ZTP を使用する場合**：クラシック ZTP はセキュア ZTP よりも簡単に実装できます。PDC、所有者証明書、または所有権バウチャーは必要ありません。デバイスとサーバーの検証が厳密ではなくなり、設定も複雑でないため、処理エラーの影響を受けにくくなります。セキュア ZTP ではサポートされていないため、シスコのデバイスが 7.3.1 より前の IOS XR バージョンを実行している場合は、これが唯一の選択肢となります。クラシック ZTP にはデバイスのシリアル番号チェックが含まれていますが、トランスポート層では安全ではありません。リモートデバイスへのルートがメトロネットワークまたはその他のセキュアでないネットワークを通過する場合は推奨されません。

- **セキュア ZTP を使用する場合**：パブリックネットワークを通過する必要があり、セキュア ZTP をサポートするデバイスがある場合は、セキュア ZTP を使用します。この ZTP が提供する追加のセキュリティには、クラシック ZTP よりも複雑な設定が必要です。設定タスクを初めて使用する場合、この複雑さが原因で処理エラーが発生しやすくなります。セキュア ZTP の設定には、デバイスの製造元からの証明書と所有権バウチャーも必要です。クラシック ZTP はサードパーティ製ハードウェアをサポートしていないため、サードパーティ製のデバイスを使用している場合に使用します。サードパーティ製デバイスとそのソフトウェアは、RFC 8572 と 8366 に 100% に準拠している必要があります。サードパーティ製のデバイスのデバイス証明書には、デバイスのシリアル番号が含まれている必要があります。サードパーティ所有権バウチャーは、デバイスのシリアル番号をファイル名として使用する形式である必要があります。シスコは、すべてのサードパーティ製デバイスとのセキュア ZTP 互換性を保証することはできません。サードパーティ製デバイスのサポートの詳細については、「ZTP でのプラットフォームサポート （8 ページ）」を参照してください。

- **イメージデバイスで ZTP を使用**：クラシックまたはセキュア ZTP を使用する場合、ソフトウェアイメージを指定する必要はありません。この機能を使用すると、ソフトウェアイメージがすでにインストールされている 1 台以上のデバイスをリモートの場所に出荷できます。その後、これらのデバイスに接続し、リモートで ZTP 処理をトリガーできます。設定方法に応じて、次を適用できます。

  - 設定のみ

- 複数の設定を持つ 1 つ以上のイメージまたは SMU。

すべてのライセンスは、90 日間の評価期間から始まります。評価期間が終了すると、Cisco Crosswork は、評価ライセンスの期限が切れたことをユーザーに警告するバナーを表示します。ZTP はこのバナーを表示しますが、設定のダウンロードを含む一部の操作をブロックします。組織がスマートライセンスに登録し、一部のオンボードデバイスにライセンスを適用すると、ZTP はブロックを削除します。ZTP は、すべてのオンボードデバイスのライセンスを取得するまで、警告バナーを表示します。

セキュア ZTP は、事前設定、Day 0、および設定後のスクリプト実行機能を提供するため、事前にイメージ化されたデバイスにより高い柔軟性が実現します。ただし、どちらの ZTP モードでも、イメージ、SMU、および設定をロードする設定ファイルを連鎖させることができます。

どちらの場合も、結果としてデバイスがオンボーディングされます。Cisco Crosswork にオンボーディングすると、ZTP を使用してデバイスを設定することはできません。

- **設定の整理**：デバイス間で可能な限り一貫した設定を維持します。一貫性により、問題の解決が容易になります。新しいデバイスをオンラインにするために実行する必要がある追加設定の量を最小限に抑えます。また、デバイスを再設定またはアップグレードする際に留意すべき「特別な」事項の数を減らします。最初に、同じデバイスファミリの同じロールを持つすべてのデバイスの基本設定が同じか、または類似していることを確認します。

  デバイスが果たす役割の定義方法は、組織、その運用方法、およびネットワーク環境の複雑さによって異なります。たとえば、組織が金融サービス企業であるとします。路上の ATM、標準的な営業時間中に開いている小売店、民間のトレーディングオフィスの 3 つのタイプのブランチがあります。各タイプのブランチのすべてのデバイスを対象とする 3 つのセットの基本プロファイルを定義できます。これらプロファイルのそれぞれに設定ファイルをマッピングできます。

  一貫性を強制する別の方法は、同様のタイプのデバイスの基本的なスクリプト設定を開発し、スクリプトロジックを使用して他のスクリプトを呼び出すことです。Classic ZTP を使用している場合、スクリプトは指定した設定ファイルにあります。このスクリプトは、基本設定をダウンロードしてから、ブランチタイプに応じて他のスクリプトをダウンロードします。セキュア ZTP を使用する場合は、メイン設定スクプリトまたは Day 0 設定スクリプトに加えて、事前設定および設定後のスクリプトを指定できるため、柔軟性が高まります。

ZTP 設定のワークフロー

ゼロタッチプロビジョニングでは、ZTP ブートと設定をトリガーする前に、次の設定タスクを最初に実行しておく必要があります。

1. 環境が、セキュリティ、プロバイダ設定、およびデバイス接続に関する ZTP の前提条件を満たしていることを確認します。

2. ZTP で処理に必要となるアセットをアセンブルします。必要なアセットは次のとおりです。
   - インストールするソフトウェアイメージ。
   - 適用する設定。
   - デバイスにアクセスするためのクレデンシャル。
   - デバイスのシリアル番号。

   セキュアZTPを使用している場合、これらのアセットには、デバイス所有者証明書、PDC、所有権バウチャーも含まれます。
3. アセンブルした ZTP アセットを Cisco Crosswork にロードします。
4. アセンブルしたクレデンシャルアセットを使用してクレデンシャルプロファイルを作成します。
5. ZTP デバイスエントリファイルを準備します。これらのファイルで、ZTP がデバイスを Cisco Crosswork デバイスインベントリにオンボーディングするために使用する Cisco Crosswork デバイスエントリを作成します。オンボーディングするデバイスが多数ある場合は、CSV ファイルをインポートしてエントリを一括で作成します。オンボーディングするデバイスが少数の場合は、Cisco Crosswork の UI を使用してこれらのエントリを 1 つずつ作成するほうが便利です。

この項の残りのトピックでは、これらの各タスクの実行方法について説明します。

ZTP の前提条件を満たす

ZTP との互換性を確保するために、Cisco Crosswork のインストールは次の前提条件を満たしている必要があります。

- Classic ZTP を使用してデバイスをオンボードしている場合は、Cisco Crosswork とデバイスが安全なネットワークドメインにあることを確認してください。
- ZTP にデバイスを Cisco NSO へオンボーディングさせる場合は、NSO を Cisco Crosswork プロバイダとして 設定します。必ず NSO プロバイダのプロパティキーを `forward` に、プロパティ値を `true` に設定してください。
- Cisco Crosswork クラスタはデバイスから、クラスタはデバイスから、アウトオブバンド管理ネットワークまたはインバウンド データ ネットワークのいずれかを介して到達可能である必要があります。これらの要件の範囲の一般的な表示については、『*Cisco Crosswork Infrastructure 4.0 and Applications Installation Guide*』の「Network Requirements」の項にあるネットワーク図を参照してください。この種のアクセスを有効にするには、静的ルートを追加し、ファイアウォール設定を変更する必要がある場合があります。

ZTP アセットのアセンブル

クラシック ZTP とセキュア ZTP の両方で、次の ZTP アセットを収集する必要があります。

- **ソフトウェアイメージ**：ネットワークデバイスの機能を可能にする、CiscoIOS-XR などのインストール可能なオペレーティング システム ソフトウェア。シスコは、イメージを TAR、ISO、または RPM ファイルとして配布します。各イメージファイルは、特定のデバイスプラットフォームまたはファミリの特定のネットワーク OS の単一リリースを表します。イメージファイルを一度に 1 つずつ Cisco Crosswork にアップロードし、各ソフトウェアイメージファイルの各 MD5 チェックサムを入力します。Cisco Crosswork は MD5 チェックサムを使用してファイルの整合性を検証します。シスコまたはサードパーティの製造元からデバイスイメージをダウンロードする場合は、チェックサムを必ず記録してください。アップロードするイメージの独自の MD5 チェックサムを生成することもできます。

- **ソフトウェア メンテナンス アップデート（SMU）**：特定のソフトウェアリリースの 1 つまたは複数の重大な問題に対するポイントフィックスを提供するシスコソフトウェアパッケージ。シスコは、関連する問題を説明する readme.txt ファイルを使用してブート不可形式の SMU を配布しています。シスコは、ソフトウェアイメージの次のメンテナンスリリースに SMU のコンテンツを展開します。ソフトウェアイメージのダウンロード中ではなく、構成ファイルを使用して SMU を適用します。SMU を一度に 1 つずつ Cisco Crosswork にアップロードします。

  現在のデバイスと有効なサポート契約を結んでいるシスコのお客様は、Cisco Support & Downloads ページを使用して、シスコのソフトウェアイメージと SMU を検索およびダウンロードできます。

- **設定**：ZTP は設定ファイルを使用して、SMU を使用したソフトウェアのアップグレードなど、特定のデバイスにインストールされているソフトウェアイメージの機能を設定します。設定ファイルは、Linux シェルスクリプト（SH）、Python スクリプト（PY）、または ASCII テキストファイル（TXT）に保存されたデバイスのオペレーティングシステムの CLI コマンドです。組織またはコンサルタントが設定を作成します。Cisco Crosswork に設定ファイルを 1 つずつアップロードします。カスタム設定コードは置換可能なパラメータを使用でき、多くのタスクを完了するために Cisco Crosswork API 呼び出しを使用する必要があります。特に、デバイスが 1 つの ZTP 状態から別の状態に移行したときに、コードで API コールを使用して Cisco Crosswork サーバーに通知する必要があります。これらのパラメータと API 呼び出しの使用方法の例については、サンプルの ZTP 設定ファイルを参照してください。Cisco Crosswork から ZTP 設定例ファイルをダウンロードするには、**[デバイス管理（Device Management）]** > **[ZTP設定ファイル（ZTP Configuration Files）]** を選択し、[サンプルスクリプトのダウンロード（XR）（Download Sample Script (XR)）] をクリックできます。詳細については、次のセクション「デフォルトの置換可能なパラメータ」および「カスタムの置換可能なパラメータの作成」を参照してください。セキュア ZTP を使用すると、事前設定ファイル、設定後ファイル、およびメインまたは Day 0 設定ファイルを読み込むことができます。

- **クレデンシャル**：Cisco Crosswork がデバイスにアクセスして制御するために使用するユーザー名とパスワード。それらをクレデンシャルプロファイルとしてロードすると、Cisco Crosswork はそれらを暗号化された形式で保存します。GUI を使用してクレデンシャルプ

ロファイルを 1 つずつ作成することも、クレデンシャルプロファイルの CSV ファイルをダウンロードして変更することで一括でロードすることもできます。

- **シリアル番号**：ZTP を使用してオンボーディングする予定のデバイスのシリアル番号。クラシックまたはセキュア ZTP を使用して、オンボードする予定の各デバイスのシリアル番号を入力します。デバイスエントリを作成する前に、CSV ファイルをインポートして、シリアル番号を一括でロードします。セキュア ZTP を使用する場合は、所有権バウチャーを要求するときにシリアル番号をシスコに送信してください。

セキュア ZTP の使用を計画している場合は、次の追加の ZTP アセットを組み立てます。

- **所有者証明書**：所有者証明書と所有者キーの両方を Cisco Crosswork にロードして、各デバイスのリーフ証明書を生成できるようにします。
- **固定ドメイン証明書（PDC）**：所有者証明書とともに PDC を Cisco Crosswork にロードします。また、所有権バウチャーを要求するときに PDC をシスコに送信します。
- **所有権バウチャー（OV）**：他の証明書とともに OV をロードします。シスコまたはサードパーティの製造元に OV を要求する場合は、PDC とデバイスのシリアル番号を送信します。シスコは、準備が整った時点で、Tarball 内の 1 つ以上の VCJ ファイルとして OV を返します。この交換は、お客様とお客様のシスコアカウントチームが合意した安全な方法を使用して行われます。サードパーティ製デバイス用のバウチャーを使用している場合、製造元が提供する VCJ ファイルは命名規則 *serial.vcj* に従う必要があります。ここで、*serial* は対応するデバイスのシリアル番号です。Cisco Crosswork では、所有権バウチャーをデバイスにマッピングするために、このファイル命名規則が必要です。
- **SUDI ルート CA 証明書**：他の証明書および OV と同時に SUDI ルート CA 証明書をロードします。Cisco SUDI ルート証明書は、「Cisco PKI: Policies, Certificates, and Documents」ページ（https://www.cisco.com/security/pki/policies/index.html）からダウンロードできます。

一部の組織は、承認された資産のライブラリを維持しています。組織にこのようなライブラリがある場合は、これらのアセットにクライアントマシンから簡単にアクセスできることを確認します。これにより、ZTP の設定を簡単に実行できます。

デフォルトの置換可能なパラメータ

次の表に、カスタム設定ファイルで使用できるデフォルトの置換可能パラメータを示します。実行時に、これらの各プレースホルダを Cisco Crosswork は各デバイスの適切な値に置き換えます。これらのプレースホルダの使用例については、このトピックの前のセクションで説明したように、Cisco Crosswork からサンプル設定スクリプトをダウンロードしてください。

表 *1: ZTP* 設定ファイルのデフォルトパラメータ

Cisco Crosswork が置換するプレースホルダ	**...からの値を使用して...**
{$HOSTNAME}	ZTP デバイスエントリで指定されているデバイスのホスト名。
{$IP_ADDRESS}	DHCP によって割り当てられたデバイスの IP アドレス。

Cisco Crosswork が置換するプレースホルダ	...からの値を使用して...
{$SSH_USERNAME}	クレデンシャルプロファイルの [ユーザー名（User Name）] フィールドの値（[接続タイプ（Connectivity Type）] が [SSH] の場合）。
{$SSH_PASSWORD}	クレデンシャルプロファイルの [パスワード（Password）] フィールドの値（[接続タイプ（Connectivity Type）] が [SSH] の場合）。
{$SSH_ENPASSWORD}	クレデンシャルプロファイルの [イネーブルパスワード（Enable Password）] フィールドの値（[接続タイプ（Connectivity Type）] が [SSH] の場合）。
{$SNMP_READ_COM}	クレデンシャルプロファイルの [読み取りコミュニティ（Read Community）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv2] の場合）。
{$SNMP_WRITE_COM}	クレデンシャルプロファイルの [書き込みコミュニティ（Write Community）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv2] の場合）。
{$SNMP_SEC_LEVEL}	クレデンシャルプロファイルの [セキュリティレベル（Security Level）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv3] の場合）。
{$SNMP_USERNAME}	クレデンシャルプロファイルの [ユーザー名（User Name）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv2] または [SNMPv3] の場合）。
{$SNMP_AUTH_TYPE}	クレデンシャルプロファイルの [ユーザー名（User Name）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv3] で [セキュリティレベル（Security Level）] が [AUTH_NO_PRIV] または [AUTH_PRIV] の場合）。
{$SNMP_AUTH_PASS}	クレデンシャルプロファイルの [ユーザー名（User Name）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv3] で [セキュリティレベル（Security Level）] が [AUTH_NO_PRIV] または [AUTH_PRIV] の場合）。
{$SNMP_PRIV_TYPE}	クレデンシャルプロファイルの [ユーザー名（User Name）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv3] で [セキュリティレベル（Security Level）] が [AUTH_PRIV] の場合）。
{$SNMP_PRIV_PASS}	クレデンシャルプロファイルの [プライバシーパスワード（Priv Password）] フィールドの値（[接続タイプ（Connectivity Type）] が [SNMPv3] で [セキュリティレベル（Security Level）] が [AUTH_PRIV] の場合）。

カスタム置換可能パラメータ

次の例に示すように、独自の置換可能パラメータを設定ファイルに作成できます。

```
!
hostname {$name}
username {$ssh_name}
 group root-lr
 group cisco-support
 secret {$ssh_pwd}
!
tpa
 vrf default
 !
!
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method http
 !
!

interface loopback1
 ipv4 address {$ip1}
interface loopback2
 ipv4 address {$ip2}
```

ZTP アセットのロード

クレデンシャルプロファイルを作成する前に、組み立てた ZTP アセットをアップロードします。

クラシックとセキュア ZTP の両方で、以下をロードする必要があります。

- ソフトウェア イメージ
- SMU
- コンフィギュレーション ファイル
- デバイスのシリアル番号

セキュア ZTP では、次をロードする必要があります。

- 固定ドメイン証明書
- 所有権証明書
- 所有権バウチャー

マップされたネットワークドライブを使用して、ソフトウェアイメージ、SMU、および設定ファイルをアップロードできます。

Cisco Crosswork は、重複するシリアル番号をチェックし、それらを自動的に単一のエントリにマージします。Cisco Crosswork は、アップロードしたすべての所有権バウチャーを既存のシリアル番号に自動的に関連付けます。

イメージ、設定ファイル、およびシリアル番号を任意の順序でアップロードできます。シリアル番号をロードした後にのみ、証明書と所有権バウチャーをロードします。

ステップ 1 画像と SMU をアップロードします。

a) メインメニューから、**[デバイス管理（Device Management）] > [ソフトウェアイメージ（Software Images）]** を選択し、＋をクリックします。

b) 必要なイメージまたは SMU のファイル情報を入力し、[追加（Add）] をクリックします。

ファイルの MD5 チェックサムを入力する必要があります。

[参照（Browse）] をクリックして、ISO、TAR、または RPM ファイルを選択することもできます。

c) ＋をクリックし、すべてのイメージと SMU ファイルをロードするまで、手順 1b を繰り返します。

ステップ 2 設定ファイルとスクリプトをアップロードします。

a) メインメニューから、**[デバイス管理（Device Management）] > [設定ファイル（Configuration Files）]** を選択し、＋をクリックします。

b) 必要な設定ファイル情報を入力して [追加（Add）] をクリックします。[参照（Browse）] をクリックして PY、SH、または TXT 設定ファイルを選択します。

c) ＋をクリックし、すべての設定ファイルをロードするまで手順 2b を繰り返します。セキュア ZTP を実装する場合は、事前設定前、事前設定後、メイン、または Day 0 設定ファイルを含めます。

ステップ 3 デバイスのシリアル番号をアップロードします。

a) メインメニューから、**[デバイス管理（Device Management）] > [シリアル番号とバウチャー（Serial Number and Voucher）]** を選択し、[シリアル番号の追加（Add Serial Number）] をクリックします。

b) [CSVのアップロード（Upload CSV）] をクリックし、[serialnumber.csv] リンクをクリックして sampleSerialnumber.csv ファイルをダウンロードします。

c) 選択した CSV ファイルエディタを使用して、ZTP を使用してオンボーディングする予定のすべてのデバイスのシリアル番号をテンプレートに入力します。更新した CSV ファイルテンプレートを新しい名前で保存します。

d) [シリアル番号の追加（Add Serial Number）] を再度選択します。[参照（Browse）] をクリックして更新した CSV ファイルを選択し、[シリアル番号の追加（Add Serial Number）] をクリックしてシリアル番号をインポートします。

ステップ 4 セキュア ZTP を実装する場合、次の手順に進みます。

ステップ 5 固定されたドメイン証明書、所有者証明書、および SUDI ルート CA 証明書をアップロードします。

a) メインメニューから、**[管理（Administration）] > [証明書管理（Certificate Management）]** を選択し、＋をクリックします。

b) [証明書名（Certificate Name）] に、この証明書グループの名前を入力します。

c) [証明書の役割（Certificate Role）] で、[セキュアZTPプロビジョニング（Secure ZTP Provisioning）] を選択します。

d) [参照（Browse）] をクリックして、[ピン留めされたドメインCA証明書（Pinned Domain CA Certificate）]、[所有者証明書（Owner Certificate）]、および [所有者キー（Owner Key）] ファイルを選択します。
e) **[保存（Save）]** をクリックします。

ステップ 6 所有権バウチャーのアップロード

a) メインメニューから、**[デバイス管理（Device Management）]** > **[シリアル番号とバウチャー（Serial Number and Voucher）]** を選択し、[バウチャーの追加（Add Voucher）] をクリックします。
b) [参照（Browse）] をクリックして、シスコ提供の VCJ ファイル（または、複数のバウチャーがある場合は、所有権バウチャーを含むTARball）を選択します。次に [アップロード（Upload）] をクリックします。

サードパーティのデバイスのバウチャーをアップロードする場合、アップロードされた VCJ ファイルまたは TARball 内のファイルは、命名規則 *serial*.vcj に従う必要があります。この規則では、 シリアルは対応するデバイスのシリアル番号です。Cisco Crosswork では、所有権バウチャーをデバイスにマッピングするために名前を付ける必要があります。

ZTP でのクレデンシャルプロファイルの作成

Cisco Crosswork ZTP では、デバイスにアクセスして設定するのにクレデンシャルプロファイルが必要です。次に、CSV ファイルを使用して一括でクレデンシャルプロファイルを追加する方法を示します。クレデンシャルプロファイルを 1 つずつ追加するには、**[デバイス管理（Device Management）]** > **[クレデンシャルプロファイル（Credential Profiles）]**を選択し、⊞をクリックします。

デバイスで有効になっている SNMP のバージョンに対してのみ、SNMP クレデンシャルプロファイルを作成することをお勧めします。例：デバイス設定で SNMPv2 のみが有効になっている場合は、プロファイルに SNMPv3 クレデンシャルを含めないでください。

ステップ 1 メインメニューから **[デバイス管理（Device Management）]** > **[クレデンシャルプロファイル（Credential Profiles）]** を選択します。

ステップ 2 ⎘をクリックします。

ステップ 3 [「Credential template（*.csv）」サンプルファイルのダウンロード（Download sample 'Credential template (*.csv)' file）] リンクをクリックし、CSV ファイルテンプレートをローカルに保存します。

ステップ 4 任意のエディタを使用して CSV テンプレートを開きます。作成するクレデンシャルプロファイルごとに 1 行ずつファイルに行を追加します。

これを行う場合は、次のガイドラインに従います。

- クレデンシャルプロファイルの [パスワード（Password）] 列が空白の場合、CSV ファイルをインポートできません。必要に応じて、これらのフィールドに実際のパスワードを入力できます。Cisco Crosswork は暗号化された形式でこれらのパスワードを保存します。この方法を選択した場合は、アップロード後すぐに CSV ファイルを破棄してください。CSV ファイルの [パスワード（Password）] 列にアスタリスクを入力してインポートすることをお勧めします。インポートが成功したら、Cisco Crosswork の

GUI を使用して各プロファイルを編集し、次の手順で説明するように実際のパスワードを入力できます。

- 同じフィールド内で複数のエントリを区切るには、セミコロンを使用します。
- 複数のエントリをセミコロンで区切る場合は、各フィールドに値を入力する順序が重要であることに注意してください。1つの列の最初のエントリは次の列の最初のエントリにマッピングされます。例：[パスワードタイプ（Password Type）] に、パスワードタイプのリスト、**ROBOT_USERPASS_SSH;ROBOT_USERPASS_TELNET;ROBOT_USERPASS_NETCONF** を入力します。次に、[ユーザー名（User Name）] 列に **Tom;Dick;Harry;**、[パスワード（Password）] 列に **root;MyPass;Turtledove;** と入力します。これらの 3 つの列の入力順序によって、入力した値間の結果のマッピングが決まります。
  - ROBOT_USERPASS_SSH: Tom : root
  - ROBOT_USERPASS_NETCONF: Dick : MyPass
  - ROBOT_USERPASS_TELNET: Harry : Turtledove
- ファイルを保存する前に、サンプルデータ行を必ず削除してください。列ヘッダー行は無視できます。

ステップ 5 完了したら、CSV ファイルを新しい名前で保存します。

ステップ 6 必要に応じて、**[デバイス管理（Device Management）]**>**[クレデンシャルプロファイル（Credential Profiles）]** を再度選択し、 をクリックします。

ステップ 7 [参照（Browse）] をクリックして CSV ファイルまで移動し選択します。

ステップ 8 CSV ファイルを選択した状態で、[インポート（Import）] をクリックします。

ステップ 9 インポートが完了したら、次の手順を実行します。

a) [クレデンシャルプロファイル（Credential Profiles）] ウィンドウの左側から、更新するプロファイルを選択し、 をクリックします。

b) クレデンシャルプロファイルのパスワードとコミュニティ文字列を入力し、[保存（Save）]をクリックします。

c) すべてのパスワードとコミュニティ文字列を入力するまで、必要に応じてこれらの手順を繰り返します。

ZTP プロファイルの作成

Cisco Crossworkは、ZTPプロファイルを使用して、イメージ化プロセスと設定プロセスを自動化します。ZTPプロファイルはオプションですが、作成することを強くお勧めします。ZTPイメージ化と設定プロセスを簡素化するのに役立ちます。ZTPプロファイルを使用すると、特定のクラスのまたはデバイスファミリ内のデバイスに適用できる、定義済みのイメージファイルと設定ファイルのセットを整理できます。

クラシック ZTP を実装する場合、各 ZTP プロファイルには 1 つのイメージファイルと、1 つの設定ファイルのみを関連付けることができます。セキュア ZTP を使用すると、事前設定ファイル、設定後ファイル、およびメインまたは Day 0 設定ファイルを指定できます。

ZTP プロファイルでは、イメージファイルを指定する必要はありません。

ZTP プロファイルはいくつでも作成できます。デバイスファミリごと、ユースケースごと、またはネットワークロールごとに 1 つの ZTP プロファイルのみを作成することをお勧めします。

ステップ 1 メインメニューから **[デバイス管理（Device Management）] > [ゼロタッチプロファイル（Zero Touch Profiles）]** を選択します。

ステップ 2 [+ 新しいプロファイル（+ New Profile）] をクリックします。

ステップ 3 新しい ZTP プロファイルに必要な値を入力します。プロファイルのソフトウェアイメージを指定する必要はありません。

ステップ 4 セキュア ZTP を実装している場合、[セキュア ZTP を有効にする（Enable Secure ZTP）] スライダを調整し、事前および事後の設定ファイルの名前を入力します。

ステップ 5 [保存（Save）] をクリックして新しい ZTP プロファイルを作成します。

ZTP デバイスエントリファイルの作成

Cisco Crosswork は、ZTP デバイスエントリを使用して、プロビジョニングするデバイスの IP アドレス、プロトコル、およびその他の情報を事前に指定できます。Cisco Crosswork は、ZTP 処理が正常に完了すると、これらのインポートされたエントリに詳細情報を入力します。

デバイスエントリの CSV ファイルをインポートすることで、ZTP デバイスエントリを一括で作成できます。

次のトピックでは、デバイスエントリ CSV ファイルのテンプレートをダウンロードする方法について説明します。また、適切にフォーマットされた ZTP デバイスエントリを作成する方法についても説明します。

慣れるまでは、デバイスエントリの CSV ファイル形式を試すことをお勧めします。テンプレートのコピーに 1 つまたは 2 つのデバイスエントリのみを追加し、インポートします。その後、必要な結果が得られるかどうかを確認できます。

また、次のトピックで説明するように、Cisco Crosswork の UI を使用して、ZTP デバイスエントリを 1 つずつ作成することもできます。

ZTP デバイス エントリ テンプレートのダウンロードと編集

1. メインメニューから **[デバイス管理（Device Management）] > [デバイス（Devices）]** を選択します。
2. [ゼロタッチデバイス（Zero Touch Devices）] タブをクリックします。
3. をクリックします。

4. [「devices import」テンプレート（.csv）のダウンロード（Download 'devices import' template (.csv)）] リンクをクリックし、[保存（Save）] をクリックしてローカルストレージリソースに保存します。[キャンセル（Cancel）] をクリックしてダイアログボックスをクリアします。
5. 選択したアプリケーションで CSV テンプレートを開き、新しい名前で保存します。各行で、ZTP を使用してオンボーディングする予定の各デバイスのエントリを作成します。各列に入力する値については、次のトピックの項を参照してください。

ZTP デバイスエントリの CSV テンプレートリファレンス

次の表で、テンプレート内の列の使用方法について説明します。エントリを必要とする列については、列名の横にアスタリスク（*）を付けて示しています。

4 つの [接続（Connectivity）] 列では複数のエントリが許可されているため、1 台のデバイスに複数の接続プロトコルを指定できます。このオプションを使用する場合は、エントリ間にセミコロンを使用し、次の 3 つの列に同じ順序で値を入力します。たとえば、[接続プロトコル（Connectivity Protocol）] 列に **`SSH;NETCONF;`** と入力するとします。[接続ポート（Connectivity Port）] 列に **`23;830;`** と入力した場合、2 つの列のエントリは次のようにマッピングされます。

- SSH：22
- NETCONF：830

表 *2 : ZTP* デバイス エントリ テンプレートの列リファレンス

カラム	使用方法
UUID	自分で生成して入力することを選択しない限り、Cisco Crosswork はランダムな UUID を割り当てます。デバイスに割り当てられた 128 ビットの汎用一意識別子を入力します。
ホスト名（Host Name）*	デバイスに割り当てるホスト名を入力します。
シリアル番号（Serial Number）*	デバイスのシリアル番号を入力します。同じデバイスに対して最大 3 つのシリアル番号を入力できます。これらは、以前に Cisco Crosswork にロードした各デバイスのシリアル番号と同じである必要があります。 ZTP では、通常のすべての展開にシリアル番号のエントリが必要です。DHCP Option 82 を使用してリレーエージェントを実装する場合は、このフィールドを空白のままにすることもできますが、デバイスを識別するためにリモート ID と回線 ID は指定する必要があります。
MAC アドレス（MAC Address）	デバイスの MAC アドレスを入力します。

カラム	使用方法
IPアドレス（IP Address）	デバイスのIPアドレス（IPv4またはIPv6）と、そのサブネットマスクをスラッシュ表記で入力します。
クレデンシャルプロファイル（Credential Profile）*	Cisco Crosswork がデバイスにアクセスして設定するために使用するクレデンシャルプロファイルの名前を入力します。クレデンシャルプロファイルを使用する場合にのみ必要です。
OS プラットフォーム（OS Platform）*	デバイスの OS プラットフォームを入力します。例：IOS-XR。
バージョン（Version）*	デバイスプラットフォームイメージのOSプラットフォームのバージョンを入力します。プラットフォームのバージョンは、プロビジョニングに使用するイメージファイルと設定ファイルに指定されているものと同じバージョンである必要があります。現在、ZTPはIOS-XRバージョン 6.6.3、7.0.1、7.0.2、および 7.0.12 をサポートしています。 [プロファイル名（Profile Name）] 列に ZTP プロファイルを指定しない場合にのみ必要です。
デバイスファミリ（Device Family）*	デバイスのデバイスファミリを入力します。デバイスファミリは、ZTPがプロビジョニングに使用するイメージファイルと設定ファイルのデバイスファミリと一致する必要があります。 [プロファイル名（Profile Name）] 列に ZTP プロファイルを指定しない場合にのみ必要です。
イメージ ID（Image ID）	デバイスにインストールするソフトウェアイメージファイルの Cisco Crosswork によって割り当てられた ID を入力します。
設定 ID（Config ID）*	デバイスの設定時に使用する設定ファイルの Cisco Crosswork によって割り当てられた ID を入力します。
プロファイル名（Profile Name）	このデバイスのプロビジョニングに使用するZTPプロファイルの名前を入力します。
設定属性（Configuration Attributes）	デバイスの設定ファイルの置換可能パラメータに Cisco Crosswork で使用する値を入力します。セキュア ZTP を使用している場合は、事前設定前、事前設定後、Day 0 設定ファイルパラメータを含めることができます。

カラム	使用方法
接続プロトコル（Connectivity Protocol）	デバイスをモニターするため、または Cisco Crosswork アプリケーションと機能をサポートするために必要な接続プロトコル。選択できるプロトコルは、**SSH**、**SNMPv2**、**NETCONF**、**TELNET**、**HTTP**、**HTTPS**、**GRPC**、および **SNMPv3** です。
接続 IP アドレス（Connectivity IP Address）*	接続プロトコルの IP アドレス（IPv4 または IPv6）とサブネットマスクを入力します。接続プロトコルの設定を選択した場合にのみ必要です。
接続ポート（Connectivity Port）*	この接続プロトコルに使用するポートを入力します。各プロトコルがポートにマッピングされます。選択したプロトコルにマッピングされるポート番号を必ず入力してください。 次の場合を除き、すべてのデバイスに 1 つ以上のポートとプロトコルを指定します。 • オンボードデバイスのステータスを管理対象外またはダウンに設定します。 • オンボーディングしたデバイスの Cisco Crosswork 到達可能性チェックを無効にします。 デバイスごとに複数のプロトコルとポートを指定する必要がある場合があります。指定するプロトコルとポートの数は、Cisco Crosswork の設定方法と使用している Crosswork アプリケーションによって異なります。次のセクション「Crosswork 接続プロトコルの要件」の表を参照してください。
接続タイムアウト（Connectivity Timeout）	このプロトコルを使用した通信試行がタイムアウトするまでの経過時間を入力します（秒単位）。デフォルト値は 30 秒、推奨されるタイムアウト値は 60 秒です。
プロバイダー名（Provider Name）	新しい ZTP デバイスをオンボーディングするプロバイダの名前を入力します。入力する名前は、デバイス管理プロバイダーの名前と正確に一致する必要があります。
プロバイダータイプ（Provider Type）	プロバイダーのタイプ。例：NSO。
プロバイダーのノード ID（Provider Node ID）	プロバイダーのメインノードの IP アドレスまたは URL。
インベントリ ID（Inventory ID）	デバイスに割り当てるインベントリ ID を入力します。

カラム	使用方法
セキュア ZTP が有効（Secure ZTP Enabled）	セキュア ZTP を使用してデバイスをプロビジョニングする場合は TRUE、そうでない場合は FALSE と入力します。
事前設定 ID（PreConfig ID）	関連する設定ファイルを実行する前に、実行する設定スクリプトの Cisco Crosswork ID を入力します。
設定後 ID（PostConfig ID）	関連する設定ファイルを実行した直後に、実行する設定スクリプトの Cisco Crosswork ID を入力します。
ロケーションが有効（Location Enabled）	ロケーション ID を使用してデバイスを識別する場合は、TRUE と入力します。シリアル番号で識別する場合は、FALSE と入力します。TRUE と入力した場合は、対応する列にリモート ID と回線 ID を入力します。FALSE と入力した場合は、対応する列にシリアル番号を入力します。
リモート ID（Remote ID）*	セキュア ZTP を実装し、Option 82 を使用する場合：ブートストラップサーバーとして機能するリモートホストの名前を識別します。 DHCP Option 82 を使用してリレーエージェントを実装する場合は、このエントリは必須です。デバイスのリモート ID と回線 ID の組み合わせを入力する必要があります。 Option 82 を使用しない場合は、このフィールドを空白のままにできますが、デバイスのシリアル番号は指定する必要があります。
回線 ID（Circuit ID）*	セキュア ZTP を実装し、Option 82 を使用する場合：ブートストラップサーバーが要求を受信するインターフェイスまたは VLAN を識別します。 DHCP Option 82 を使用してリレーエージェントを実装する場合は、このエントリは必須です。デバイスのリモート ID と回線 ID の組み合わせを入力する必要があります。 Option 82 を使用しない場合は、このフィールドを空白のままにできますが、デバイスのシリアル番号は指定する必要があります。
routingInfo.globalospfrouterid	デバイスに OSPF を実装する場合は、デバイスの OSPF ルータ ID を入力します。
routingInfo.globalisissystemid	デバイスに IS-IS を実装する場合は、デバイスの IS-IS システム ID を入力します。
routingInfo.teRouterid	デバイスにトラフィック エンジニアリングを実装する場合は、デバイスの TE ルータ ID を入力します。

Crosswork 接続プロトコルの要件

Cisco Crossworkの機能とアプリケーションでは、デバイスごとにさまざまな接続プロトコルを有効にする必要があります。次の表に、サポートされる各接続プロトコルのこれらの要件を示します。

表 *3:* アプリケーションと機能の接続プロトコルの要件

プロトコル	ポート	アプリケーション	機能
GRPC	9090	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork 最適化エンジン（COE）	• Cisco Crosswork API 通信
HTTP	80	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork 最適化エンジン（COE）	• 3つのアプリケーションすべてでの NSO プロバイダーのオンボーディング
HTTPS	443	• Cisco Crosswork ネットワークコントローラ（CNC）	• NSO プロバイダーのオンボーディング
NETCONF	830	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork Optimization Engine	• 3つのアプリケーションすべてでの NSO プロバイダーのオンボーディング

プロトコル	ポート	アプリケーション	機能
SNMPv2	161	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork Optimization Engine	• SNMPv2 でのデータ収集
SNMPv3	161	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork Optimization Engine	• SNMPv3 でのデータ収集
SSH	22	• Cisco Crosswork ネットワークコントローラ（CNC） • Cisco Crosswork Change Automation と Health Insights（CAHI） • Cisco Crosswork Optimization Engine	• CLIデータ収集、デバイスへの SSH アクセス

単一 ZTP デバイスエントリの作成

ZTP を使用してオンボーディングするデバイスが少数の場合は、デバイスエントリを 1 つずつ作成するほうが簡単な場合があります。単一の ZTP デバイスエントリを作成するには、ZTP ユーザーインターフェイスで次の手順を実行します。

ステップ 1 メインメニューから **[デバイス管理（Device Management）]** > **[デバイス（Devices）]** を選択します。

ステップ 2 [ゼロタッチデバイス（Zero Touch Devices）] タブをクリックします。

ステップ 3 ＋ をクリックします。

ステップ 4 新しい ZTP デバイスエントリの値を入力します。

ZTP がデバイスにオンボードされた後、Cisco Crosswork はさらに多くの属性を表示する場合があります。

ステップ 5 [保存（Save）] をクリックします。

ZTP プロビジョニングのワークフロー

ZTP の設定が完了したら、次のようにデバイスをプロビジョニングして維持できます。

1. ZTP 処理をトリガーした後、Cisco Crosswork がイメージと設定ソフトウェアを安全にダウンロードできるように DHCP を設定します。
2. 作成した ZTP デバイスエントリの CSV ファイルを Cisco Crosswork にアップロードします。ファイルをインポートすると、オンボーディング時に ZTP が入力するデバイスエントリが作成されます。少数の ZTP デバイスのみをオンボーディングする場合は、代わりに ZTP ユーザーインターフェイスを使用してデバイスエントリを作成します。
3. 各デバイスの電源の再投入または CLI の再起動の実行によって ZTP 処理をトリガーします。
4. オンボーディングされるデバイスの情報を入力します。それらを編集し、（たとえば）プロビジョニング時に ZTP が検出できなかった地理的位置情報を入力します。

このコアワークフローを完了すると、次のトピックのアドバイスと方法を使用して、ZTP デバイスの継続的なメンテナンスを実行できます。

- 追加情報で ZTP デバイスを更新します。
- オンボーディング後、他のアプリケーションを使用するか、デバイスを削除して再オンボーディングした後、ZTP デバイスを再設定します。
- デバイスライセンスを消費することなく、ZTP デバイスを廃止または交換します。
- デバイスのオンボーディングに使用した ZTP アセットでハウスキーピングを実行します。
- ZTP 処理およびデバイスの問題をトラブルシューティングします。

この項の残りのトピックでは、これらの各タスクの実行方法について説明します。

ZTP デバイスエントリのアップロード

次に、事前に作成した ZTP デバイスエントリ CSV ファイルをインポートして、複数の ZTP デバイスエントリを作成する手順を示します。

インポートした ZTP デバイスエントリは、[ゼロタッチデバイス（Zero Touch Devices）] タブに常に [ステータスが（Status）] が [プロビジョニングなし（Unprovisioned）] に設定された状態で表示されます。これらは、ZTP 処理をトリガーするまで [プロビジョニングなし（Unprovisioned）] のままになります。

ステップ 1 メインメニューから **[デバイス管理（Device Management）]** > **[デバイス（Devices）]** を選択します。

ステップ 2 [ゼロタッチデバイス（Zero Touch Devices）] タブをクリックします。

ステップ 3 **[デバイスのインポート（Import Devices）]** をクリックします。

ステップ 4 [参照（Browse）] をクリックし、作成した ZTP デバイスエントリ CSV ファイルに移動してそのファイルを選択します。

ステップ 5 CSV ファイルを選択した状態で、[インポート（Import）] をクリックします。

Crosswork ZTP での DHCP の設定

ZTP処理をトリガーする前に、ZTPデバイスとそれらに適用するソフトウェアを特定する情報を使用して DHCP 設定ファイルを更新します。この情報により、Cisco Crosswork と DHCP はZTPデバイスを識別し、ネットワーク接続とファイルのダウンロードの要求に応答できるようになります。

以降のトピックでは、この要件を満たすようにDHCPサーバー設定を更新する例を示します。これらのトピックの例では、次の図に示す DHCP コンテキスト設定を前提としています。図は、Internet Systems Consortium DHCP サーバーの設定例を示しています。セキュア ZTP においてのみ、`sztp-redirect` オプションを有効にする行が必要です。クラシック ZTP を使用している場合は省略してください。

図 *5*: セキュア *ZTP DHCP* コンテキスト

```
#
authoritative;

default-lease-time 7200;
max-lease-time 7200;
# Next line is needed for Secure ZTP only;
option sztp-redirect code 143 = text;

subnet 192.168.100.0 netmask 255.255.255.0 {
  option routers 192.168.100.1;
  option domain-name "cisco.com";
  option domain-name-servers 171.70.168.183;
  option subnet-mask 255.255.255.0;
  range 192.168.100.105 192.168.100.195;
}
```

クラシック ZTP の DHCP 設定

セキュア ネットワーク ドメインのみを介してデバイスをプロビジョニングする場合は、クラシック ZTP を使用することを強くお勧めします。

クラシック ZTP でサポートされているシスコのデバイスでは、HTTP 経由でのみ iPXE ソフトウェアイメージをダウンロードできます。これらの同じデバイスは、HTTP または HTTPS を介した設定ファイルのダウンロードをサポートしています。これらのオプションでは、組織の DHCP サーバー設定に DHCP ブートファイル URL のエントリが必要です。

イメージと設定ファイルのダウンロードの両方に HTTP を使用する場合は、これらの URL で HTTP プロトコルとポート 30604 を指定する必要があります。詳細については、図 1 と 2 の例を参照してください。

設定ファイルのダウンロードのみに HTTPS を使用する場合は、URL で HTTPS プロトコルとポート 30603 を指定する必要があります。URL の HTTPS プロトコルの前に -k オプションを指定します。ヘルプについては、図 3 および 4 の例を参照してください。

ZTP では、設定のダウンロードに DHCP Option 82 を使用できます。Option 82（DHCP リレーエージェント情報オプションとも呼ばれる）は、IP スプーフィングや MAC スプーフィング、または DHCP アドレス枯渇を使用した攻撃からデバイスを保護します。Option 82 を使用すると、オンボーディングしりデバイスとデバイス要求を解決する DHCP サーバー間に配置された中間ルータまたは中継ルータを指定できます。このオプションを使用するには、ロケーション ID を指定します。ロケーション ID は、回線 ID（インターフェイスまたは VLAN ID）とリモート ID（ホスト名）で構成されます。図 2 および 4 の例に示すように、これらの値を設定ダウンロード URL のパラメータとして指定します。Option 82 の詳細については、RFC 3046（http://tools.ietf.org/html/rfc3046）を参照してください。

次の例に従う場合：

- <*CW_HOST_IP*> を Cisco Crosswork サーバーの IP アドレスに必ず置き換えてください。
- <*IMAGE_UUID*> を ZTP リポジトリのソフトウェアイメージファイルの UUID に置き換えます。ブートファイル名と UUID の使用に関するヘルプについては、このトピックの後のセクション「DHCP セットアップ用のブートファイル名と UUID のコピー」を参照してください。
- 設定ファイルには UUID は必要ありません。

図 *6 : HTTP* を使用したクラシック *ZTP DHCP* の設定

```
host cztp1 {
 hardware ethernet 00:a7:42:86:54:f1;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else if exists user-class and option user-class ="exr-config" {
     filename = "http://<CW_HOST_IP>:30604/crosswork/configsvc/v1/file";
  }
}
```

図 *7 : HTTP* と *Option 82* を使用したクラシック *ZTP DHCP* の設定

```
host cztp2 {
 hardware ethernet 00:a7:42:86:54:f2;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else if exists user-class and option user-class ="exr-config" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/configsvc/v1/file?circuitid=Gig001&remoteid=MAR1";
  }
}
```

図 *8 : HTTPS* を使用したクラシック *ZTP DHCP* の設定

```
host cztp3 {
 hardware ethernet 00:a7:42:86:54:f3;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else if exists user-class and option user-class ="exr-config" {
     filename = "-k https://<CW_HOST_IP>:30603/crosswork/configsvc/v1/file";
  }
}
```

図 *9 : HTTPS* と *Option 82* を使用したクラシック *ZTP DHCP* の設定

```
host cztp4 {
 hardware ethernet 00:a7:42:86:54:f4;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else if exists user-class and option user-class ="exr-config" {
     filename = "-k
https://<CW_HOST_IP>:30603/crosswork/configsvc/v1/file?circuitid=Gig001&remoteid=MAR1";
  }
}
```

セキュア ZTP の DHCP 設定

セキュア ZTP を使用すると、セキュアなネットワークドメインとセキュアでないネットワークドメインの両方でデバイスをプロビジョニングできます。設定ファイルのダウンロードに HTTPS を使用し、設定アーティファクトに `option sztp-redirect` を指定します。Option 82 を使用する場合は、リモート ID と回線 ID を追加します。リモート ID はブートストラップサーバーとして機能するリモートホストを識別し、回線 ID はリモートホスト上のインターフェイスまたは VLAN を識別します。図 5 と 6 の例を参照してください。ブートファイル名と UUID の使用に関するヘルプについては、次のセクション「DHCP セットアップ用のブートファイル名と UUID のコピー」を参照してください。

図 *10 : HTTPS* を使用したセキュア *ZTP DHCP* の設定

```
host sztp1 {
 hardware ethernet 00:a7:42:86:54:f4;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else {
     option sztp-redirect
"https://<CW_HOST_IP>:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data";

  }
}
```

図 *11 : HTTPS* と *Option 82* を使用したセキュア *ZTP DHCP* の設定

```
host sztp2 {
 hardware ethernet 00:a7:42:86:54:f5;
  if exists user-class and option user-class = "iPXE" {
     filename =
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";
  } else if exists user-class and option user-class ="exr-config" {
     option sztp-redirect
"https://<CW_HOST_IP>:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data?circuitid=Gig001&remoteid=MAR1";
```

```
  }
}
```

DHCP 設定用のブートファイル名と UUID のコピー

DHCPサーバーの設定ファイルを変更する場合は、各ソフトウェアイメージのブートファイル名と UUID を指定します。すでに Cisco Crosswork にアップロードしたソフトウェアイメージのリストから、両方をクリップボードに直接コピーできます。設定ファイルにはUUIDは必要ありません。

ソフトウェアイメージのブートファイル名と UUID をコピーするには、次の手順を実行します。

1. メインメニューから **[デバイス管理（Device Management）]** > **[ソフトウェアイメージ（Software Images）]** を選択します。
2. コピーする場合は、次の手順を実行します。
   - ソフトウェアイメージのブートファイル名と UUID：[イメージ/SMU名（Image/SMU Name）] 列の をクリックします。
   - ソフトウェアイメージの UUID のみ：[イメージの UUID（Image UUID）] 列の をクリックします。

   Cisco Crosswork によってブートファイル名と UUID がクリップボードにコピーされます。これを DHCP ホストエントリに貼り付けることができます。

   コピーしたファイルパスを使用して DHCP ホストエントリを作成する場合は、*IP* 変数を Cisco Crosswork サーバーの IP アドレスとポートに置き換えます。

Generic Internet Systems Consortium（ISC）DHCP の設定

次の図に、Internet Systems Consortium（ISC）DHCP サーバーの `/etc/dhcp/dhcp.conf` 設定ファイルでクラシック ZTP およびセキュア ZTP デバイスに対して作成するホストエントリのタイプの例を示します。

他のサードパーティ製DHCPサーバーは全体的な実装が異なりますが、多くの場合はこれらの ISC の例と同様のオプションと形式を使用します。

これらの新しいエントリの作成が完了したら、ISC DHCP サーバーを必ずリロードするか、または再起動します。

図 *12*：クラシック *ZTP ISC IPv4 DHCP* の設定例

```
host NCS5k-1
{
    option dhcp-client-identifier "FOC2302R09H";
    hardware ethernet 00:cc:fc:bb:be:6a;
    fixed-address 105.1.1.16;
    if exists user-class and option user-class = "iPXE" {
        filename = "http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/
           <IMAGE_UUID>
    } else if exists user-class and option user-class = "exr-config" {
```

```
        filename = "http://<CW_HOST_IP>:30604/crosswork/configsvc/v1/file";
    }
}
```

図 *13 :* クラシック *ZTP ISC IPv6 DHCP*の設定例

```
host 5501
{
    host-identifier option dhcp6.client-id
00:02:00:00:00:09:46:4f:43:32:33:30:38:52:30:53:33:00;
    fixed-address6 fc00:15:2::36;
    if exists dhcp6.user-class and substring(option dhcp6.user-class, 2, 4) = "iPXE" {
      option dhcp6.bootfile-url
"http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/
        <IMAGE_UUID>";
    } else {if exists dhcp6.user-class and substring(option dhcp6.user-class, 0, 10) =
"exr-config" {
      option dhcp6.bootfile-url
"http://<CW_HOST_IP>:30604/crosswork/crosswork/configsvc/v1/file";
    }
}
```

図 *14 :* セキュア *ZTP ISC IPv4 DHCP* の設定例

```
authoritative;
option sztp-redirect code 143 = text;

default-lease-time 7200;
max-lease-time 7200;

subnet 105.1.1.0 netmask 255.255.255.0 {
  option routers 105.1.1.254;
  option domain-name "cisco.com";
  option domain-name-servers 171.70.168.183;
  option subnet-mask 255.255.255.0;
  range 105.1.1.40 105.1.1.140;
  if exists user-class and option user-class = "iPXE" {
         filename =
"http://105.1.2.100:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-db2fb355-de5b-4c13-8290-346c4daaa577";

      } else {
option sztp-redirect
"https://105.1.2.100:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data";

  }

}
```

図 *15 :* セキュア *ZTP ISC IPv6 DHCP* の設定例

```
default-lease-time 2592000;
preferred-lifetime 604800;
option dhcp-renewal-time 3600;
option dhcp6.user-class code 15 = string;
option dhcp6.bootfile-url code 59 = string;
option dhcp-rebinding-time 7200;
allow leasequery;
option dhcp6.name-servers 3ffe:501:ffff:100:200:ff:fe00:3f3e;
option dhcp6.domain-search "cisco.com";
option sztp-redirect code 136 = text;

option dhcp6.info-refresh-time 21600;
subnet6 fc00::/64 {
 range6 fc00::10:10:101 fc00::10:10:105;
```

```
}
 host CW14-NCS {

      host-identifier option dhcp6.client-id
00:02:00:00:00:09:46:4f:43:32:32:32:31:52:31:39:4e:00;
      fixed-address6 fc00::10:10:100;
      if exists dhcp6.user-class and substring(option dhcp6.user-class, 2, 4) = "iPXE"
{
  option dhcp6.bootfile-url
"http://[fc00::10:11:97]:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-db2fb355-de5b-4c13-8290-346c4daaa577";

      } else {
option sztp-redirect
"https://[fc00::10:11:20]:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data";

 }

 }
```

次の表に、IPv4 ISC DHCP デバイスエントリの例内の各行と、使用される値のソースを示します。説明は、IPv4 のクラシック ZTP とセキュア ZTP の両方に適用されます。IPv6 の例のエントリの説明は同じですが、IPv6 のアドレッシング方式に適合させています。

表 *4 : ISC IPv4 DHCP* 設定ホストのエントリと値

IPv4 エントリ	説明
`host NCS5k-l`	デバイスエントリのホスト名。ホスト名は、実際に割り当てられたホスト名と同じにすることができますが、同じである必要はありません。
`option dhcp-client-identifier`	デバイスエントリの一意の ID。クラシック ZTP と IPv4 の例に示されている値「`FOC2302R09H`」は、デバイスのシリアル番号です。シリアル番号はデバイスのシャーシで確認できます。デバイスに物理的にアクセスできない場合は、IOS-XR の `show inventory` コマンドでシリアル番号が表示されます。
`hardware ethernet 00:cc:fc:bb:be:6a`	デバイスのイーサネット NIC ポートの MAC アドレス。このアドレスは、ZTP プロセスをトリガーするアドレスです。Cisco Crosswork から到達可能なアドレスであれば、管理ポートまたはデータポートを指定できます。
`fixed-address 105.1.1.16`	設定時にデバイスに割り当てられる IP アドレス。この例は静的 IP の場合ですが、標準の DHCP IP のプール割り当てコマンドを使用することもできます。
`option user-class = "iPXE"` and `filename =`	この行は、着信 ZTP 要求に「`iPXE`」オプションが含まれていることを確認します。クラシック ZTP では、このオプションを使用してデバイスをイメージ化します。要求にこのオプションが含まれている場合、デバイスは、`filename =` パラメータで指定された UUID とパスに一致するイメージファイルをダウンロードします。

IPv4 エントリ	説明
クラシック ZTP：`option user-class = "exr-config"` および `ffl filename =` セキュア ZTP：`option sztp-redirect code 143=text`	この行は、着信 ZTP 要求に「`exr-config`」オプションが含まれていることを確認します。ZTP はこのオプションを使用してデバイスを設定します。要求にこのオプションが含まれている場合、デバイスは `filename =` パラメータで指定されたパスに一致する設定ファイルをダウンロードします。

Cisco Prime Network Registrar（CPNR）でのクラシック ZTP DHCP の設定スクリプト

次に示すのは、ZTP デバイス、イメージ、および設定ファイルのエントリを CPNR DHCP サーバーの設定ファイルに追加できるスクリプトの 2 セットです。IPv4 用に 3 つのスクリプトが 1 セット、IPv6 用に 5 つのスクリプトがもう 1 セットあります。これらのスクリプトを使用するには、次の手順を実行します。

1. スクリプトの内容をコピーして、ここに示す名前のローカルテキストファイルに貼り付けます。
2. スクリプトのコメントで説明されているように、`ztp-v4-setup-vi-nrcmd.txt` スクリプトまたは `ztp-v6-setup-vi-nrcmd.txt` スクリプトのデバイス、イメージ、および設定エントリを必要に応じて変更します。
3. 使用するスクリプトファイルをローカル CPNR サーバーのルートフォルダにコピーします。
4. 次のコマンドを使用して、CPNR サーバーでスクリプトを実行します。

   ```
   [root@cpnr-local ~]#/opt/nwreg2/local/usrbin/nrcmd -N username -P password
   <ztp-IPVersion-setup-via-nrcmd.txt
   ```

   ここで、

   - *username* は、CPNR サーバーで管理者権限を持つユーザー ID の名前です。
   - *password* は、対応する CPNR 管理者のユーザー ID のパスワードです。
   - *IPVersion* は IPv4 バージョンのスクリプトの場合は `v4`、IPv6 バージョンのスクリプトの場合は `v6` です。

（注） 次のスクリプトは、クラシック ZTP 専用です。セキュア ZTP では使用できません。

図 *16 : IPv4* スクリプト *1/3* : *ztp-v4-setup-vi-nrcmd.txt*

```
#
# Create the scope
#
scope ztp-ncs-5501-mgmt create 192.0.20.0/24

# Add the dynamic range
scope ztp-ncs-5501-mgmt addrange 200 225
```

```
# Default the routers option. Note: No need to do subnet-mask. It is automatically
provided.
scope-policy ztp-ncs-5501-mgmt setoption routers 10.10.10.1

# Set the lease time for clients on this scope
scope-policy ztp-ncs-5501-mgmt setoption dhcp-lease-time 216000
#
# Load the option 43 definitions
import option-set ztp-v4-option-set.txt
#
# Set the client classing expression and enable use of client-class
dhcp set client-class-lookup-id=@ztp-v4-client-class-expr.txt
dhcp enable client-class
#
# Load the client classes - these are used to lookup the correct client details
# depending on whether an iso or script is requested by the client.
client-class ztp-iso create
client-class ztp-iso set client-lookup-id="(or (try (concat (as-string
    (request get option 61)) \"-iso\")) (request macaddress-string))"
#
client-class ztp-script create
client-class ztp-script set client-lookup-id="(or (try (concat (as-string
    (request get option 61)) \"-script\")) (request macaddress-string))"
#
# Clients that are not ztp will fall into the ztp-none class
# and should not be offered service so they are excluded.
#
client-class ztp-none create
client-class ztp-none set action=exclude
#
# Create a default client that will prevent service to unknown clients.
client default create
client default set action=exclude
#
# Create some ZTP clients
#
# For each ZTP client we create two clients based on their serial number.
# (See above for the client-lookup-id expressions.)
# One has "-iso" added to the end that will be used when the client's
# request includes "iPXE" in option 77.
# The other has "-script" added to the end that will be used when the
# client's request includes "exr-config" in option 77.
#

### Device-1 Settings ####
client <device-1-serial-num>-iso create
client-policy <device-1-serial-num>-iso set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-d3930e13-b081-4905-b2e5-051249d9b0cb"


client <device-1-serial-num>-script create
client-policy <device-1-serial-num>-script set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/configsvc/v1/configs/device/files/d1d7b441-3a27-47d1-aef0-39c3087d34c1"
client-policy <device-1-serial-num>-script setvendoroption 43 Cisco-ZTP "(1 exr-config)(2
 0)"

### Device-2 Settings ####
client <device-2-serial-num>--iso create
client-policy <device-2-serial-num>-iso set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-d3930e13-b081-4905-b2e5-051249d9b0cb"
```

```
client <device-2-serial-num>-script create
client-policy <device-2-serial-num>-script set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/configsvc/v1/configs/device/files/d1640deb-8252-47b6-aab1-a843c0c7757b"
client-policy <device-2-serial-num>-script setvendoroption 43 Cisco-ZTP "(1 exr-config)(2
 0)"

#
# Create more as needed using the above as models.
# Note: For those that need option 67 (boot file), you can use:
#   client-policy <name> setoption boot-file "<file-url>"
#
# The next line is optional. Uncomment it if you want to log what the script is doing.
# dhcp set log-settings=+incoming-packet-detail,outgoing-packet-detail,client-detail

# Assure that the server is up-to-date with this configuration
dhcp reload
```

図 *17 : IPv4* スクリプト *2/3* : *ztp-v4-setup-vi-nrcmd.txt*

```
#
# Create the scope
#
scope ztp-ncs-5501-mgmt create 192.0.20.0/24

# Add the dynamic range
scope ztp-ncs-5501-mgmt addrange 200 225

# Default the routers option. Note: No need to do subnet-mask. It is automatically
provided.
scope-policy ztp-ncs-5501-mgmt setoption routers 10.10.10.1

# Set the lease time for clients on this scope
scope-policy ztp-ncs-5501-mgmt setoption dhcp-lease-time 216000
#
# Load the option 43 definitions
import option-set ztp-v4-option-set.txt
#
# Set the client classing expression and enable use of client-class
dhcp set client-class-lookup-id=@ztp-v4-client-class-expr.txt
dhcp enable client-class
#
# Load the client classes - these are used to lookup the correct client details
# depending on whether an iso or script is requested by the client.
client-class ztp-iso create
client-class ztp-iso set client-lookup-id="(or (try (concat (as-string
    (request get option 61)) \"-iso\")) (request macaddress-string))"
#
client-class ztp-script create
client-class ztp-script set client-lookup-id="(or (try (concat (as-string
    (request get option 61)) \"-script\")) (request macaddress-string))"
#
# Clients that are not ztp will fall into the ztp-none class
# and should not be offered service so they are excluded.
#
client-class ztp-none create
client-class ztp-none set action=exclude
#
# Create a default client that will prevent service to unknown clients.
client default create
client default set action=exclude
#
# Create some ZTP clients
```

```
#
# For each ZTP client we create two clients based on their serial number.
# (See above for the client-lookup-id expressions.)
# One has "-iso" added to the end that will be used when the client's
# request includes "iPXE" in option 77.
# The other has "-script" added to the end that will be used when the
# client's request includes "exr-config" in option 77.
#

### Device-1 Settings ####
client <device-1-serial-num>-iso create
client-policy <device-1-serial-num>-iso set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-d3930e13-b081-4905-b2e5-051249d9b0db"


client <device-1-serial-num>-script create
client-policy <device-1-serial-num>-script set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/configsvc/v1/configs/device/files/d1d7b441-3a27-47d1-aef0-39c3087d34c1"
client-policy <device-1-serial-num>-script setvendoroption 43 Cisco-ZTP "(1 exr-config)(2
 0)"

### Device-2 Settings ####
client <device-2-serial-num>--iso create
client-policy <device-2-serial-num>-iso set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-d3930e13-b081-4905-b2e5-051249d9b0db"


client <device-2-serial-num>-script create
client-policy <device-2-serial-num>-script set packet-file-name=

"http://<cw-ipv4-address>:30604/crosswork/configsvc/v1/configs/device/files/d1640deb-8252-47b6-aab1-a843c0c7757b"
client-policy <device-2-serial-num>-script setvendoroption 43 Cisco-ZTP "(1 exr-config)(2
 0)"

#
# Create more as needed using the above as models.
# Note: For those that need option 67 (boot file), you can use:
#   client-policy <name> setoption boot-file "<file-url>"
#
# The next line is optional. Uncomment it if you want to log what the script is doing.
# dhcp set log-settings=+incoming-packet-detail,outgoing-packet-detail,client-detail

# Assure that the server is up-to-date with this configuration
dhcp reload
```

図 *18 : IPv4* スクリプト *3/3* : *ztp-v4-client-class-expr.txt*

```
(or
   (if (equal (as-string (request get-blob option 77)) "iPXE") "ztp-iso")
   (if (equal (as-string (request get-blob option 77)) "exr-config") "ztp-script")
   "ztp-none"
)
```

図 *19 : IPv6* スクリプト *1/5* : *ztp-v6-setup-vi-nrcmd.txt*

```
#
# create prefix for mgmt
prefix prefix-for-mgmt create 2001:DB8:10e:201a::/64
#
# Set the client classing expression and enable use
```

```
# of client-class
#
dhcp set v6-client-class-lookup-id=@ztp-v6-client-class-expr.txt
dhcp enable client-class
#
# Load the client classes - these are used to lookup the correct
# client details depending on whether an iso or script is requested
# by the client.
#
client-class ztp-iso create
client-class ztp-iso set v6-client-lookup-id=@ztp-v6-iso-lookup-expr.txt
#
client-class ztp-script create
client-class ztp-script set v6-client-lookup-id=@ztp-v6-script-lookup-expr.txt
client-class-policy ztp-script set v6-reply-options=17
#
# Delete option set (may not exist and ok if fails)
#
option-set dhcp6-cisco-custom delete
#
import option-set ztp-v6-options.txt
#
# Clients that are not ztp will fall into the ztp-none class
# and should not be offered service so they are excluded.
#
client-class ztp-none create action=exclude
#
# Create a default client that will prevent service to
# unknown clients.
#
client default create
client default set action=exclude
#
# Create some ZTP clients
#
# For each ZTP client we create two clients based on their mac-address.
# One has "-iso" added to the end that will be used when the client's
# request does not include the "exr-config" in option 77.
# The other has "-script" added to the end that will be used when the
# client's request does include "exr-config" in option 77.
#
client <device-serial-no>-iso create
# Set the vendor options using blob format as option definitions are for different data
client-policy <device-serial-no>-iso setV6VendorOption 17 dhcp6-cisco-custom "(1
exr-config)(2 0)"
# Escape the [ and ] as nrcmd (which uses tcl interpreter) will otherwise fail command
client-policy <device-serial-no>-iso setv6option bootfile-url

"http://\[cw-ipv6-address\]:30604/crosswork/imagesvc/v1/device/files/cw-image-uuid-aec596

      a1-7847-4254-966a-2456aa5"
#
client <device-serial-no>-script create
# Set the vendor options using blob format as option definitions are for different data
client-policy <device-serial-no>-script setV6VendorOption 17 dhcp6-cisco-custom "(1
exr-config)(2 0)"
# Escape the [ and ] as nrcmd (which uses tcl interpreter) will otherwise fail command
client-policy <device-serial-no>-script setv6option bootfile-url
   "http://\[cw-ipv6-address\]:30604/crosswork/configsvc/v1/configs/device/files/8eb6b7e1

      -bd54-40bb-84e0-89f11a60128b"
#

# Assure the server is up-to-date with this configuration
```

```
dhcp reload
```

図 *20 : IPv6* スクリプト*2/5* : *ztp-v6-client-class-expr.txt*

```
(or (try (if (equal (as-string (request get option 15)) "exr-config") "ztp-script"))
    (try (if (equal (as-string (request get option 15)) "iPXE") "ztp-iso"))
   "ztp-none"
)
```

図 *21 : IPv6* スクリプト *3/5* : *ztp-v6-iso-lookup-expr.txt*

```
(let (id)
  (setq id (request get option 1))
  (or
# First try extracting the serial number from DUID
      (try (if (equali (substring id 0 6) 00:02:00:00:00:09)
                (concat (as-string (substring id 6 128)) "-script")
           )
      )
# If that fails, use normal client-id (DUID) lookup
     (concat (to-string id) "-iso")

  )
)
```

図 *22 : IPv6* スクリプト *4/5* : *ztp-v6-script-lookup-expr.txt*

```
(let (id)
  (setq id (request get option 1))
  (or
# First try extracting the serial number from DUID
      (try (if (equali (substring id 0 6) 00:02:00:00:00:09)
                (concat (as-string (substring id 6 128)) "-script")
           )
      )
# If that fails, use normal client-id (DUID) lookup
     (concat (to-string id) "-script")
  )
)
```

図 *23 : IPv6* スクリプト *5/5* : *ztp-v6-options.txt*

```
# Option Definition Set Export/Import Utility
# Version: 1
#
{
  ( name = dhcp6-cisco-custom )
  ( desc = Cisco Systems, Inc. )
  ( vendor-option-enterprise-id = 9 )
  ( id-range = 2 )
  ( option-list = [
    {
      ( name = cisco-17 )
      ( id = 17 )
      ( base-type = AT_VENDOR_OPTS )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
      ( option-list = [
        {
          ( name = clientID )
```

```
  ( id = 1 )
  ( base-type = AT_NSTRING )
  ( sepstr = , )
  ( desc = ZTP - clientID )
}
{
  ( name = authCode )
  ( id = 2 )
  ( base-type = AT_INT8 )
  ( sepstr = , )
  ( desc = ZTP - authCode )
}
{
  ( id = 3 )
  ( name = md5sum )
  ( base-type = AT_NSTRING )
  ( desc = ZTP - md5sum )
}
{
  ( name = cnr-leasequery )
  ( id = 13 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = oro )
      ( id = 1 )
      ( base-type = AT_SHORT )
      ( flags = AF_IMMUTABLE )
      ( repeat = ZERO_OR_MORE )
      ( sepstr = , )
    }
    {
      ( name = dhcp-state )
      ( id = 2 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = data-source )
      ( id = 3 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = start-time-of-state )
      ( id = 4 )
      ( base-type = AT_TIME )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = base-time )
      ( id = 5 )
      ( base-type = AT_DATE )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = query-start-time )
      ( id = 6 )
```

```
  ( base-type = AT_DATE )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = query-end-time )
  ( id = 7 )
  ( base-type = AT_DATE )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = client-class-name )
  ( id = 8 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = partner-last-transaction-time )
  ( id = 9 )
  ( base-type = AT_TIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = client-creation-time )
  ( id = 10 )
  ( base-type = AT_TIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = limitation-id )
  ( id = 11 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = binding-start-time )
  ( id = 12 )
  ( base-type = AT_TIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = binding-end-time )
  ( id = 13 )
  ( base-type = AT_STIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = fwd-dns-config-name )
  ( id = 14 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = rev-dns-config-name )
  ( id = 15 )
  ( base-type = AT_NSTRING )
```

```
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = lookup-key )
  ( id = 16 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = user-defined-data )
  ( id = 17 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = prefix-name )
  ( id = 18 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = failover-state-serial-number )
  ( id = 19 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = reservation-key )
  ( id = 20 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = failover-partner-lifetime )
  ( id = 21 )
  ( base-type = AT_STIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = failover-next-partner-lifetime )
  ( id = 22 )
  ( base-type = AT_STIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = failover-expiration-time )
  ( id = 23 )
  ( base-type = AT_STIME )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = client-oro )
  ( id = 24 )
  ( base-type = AT_SHORT )
  ( flags = AF_IMMUTABLE )
```

```
      ( repeat = ZERO_OR_MORE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = failover )
  ( id = 21 )
  ( base-type = AT_BLOB )
  ( flags = AF_NO_CONFIG_OPTION,AF_SUPPORTS_ENCAP_OPTION,AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = server-state )
      ( id = 1 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = server-flags )
      ( id = 2 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = binding-status )
      ( id = 3 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = binding-flags )
      ( id = 4 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = start-time-of-state )
      ( id = 5 )
      ( base-type = AT_DATE )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = state-expiration-time )
      ( id = 6 )
      ( base-type = AT_DATE )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = failover-expiration-time )
      ( id = 7 )
      ( base-type = AT_DATE )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = bndupd-serial )
      ( id = 8 )
```

```
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = bndack-serial )
  ( id = 9 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = client-flags )
  ( id = 10 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = vpn-id )
  ( id = 11 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = lookup-key )
  ( id = 12 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = type )
      ( id = 0 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = data )
      ( id = 0 )
      ( base-type = AT_BLOB )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = user-defined-data )
  ( id = 13 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = reconfigure-data )
  ( id = 14 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = time )
```

```
      ( id = 0 )
      ( base-type = AT_DATE )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = key )
      ( id = 0 )
      ( base-type = AT_BLOB )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = requested-fqdn )
  ( id = 15 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = flags )
      ( id = 0 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = domain-name )
      ( id = 0 )
      ( base-type = AT_DNSNAME )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = forward-dnsupdate )
  ( id = 16 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = reverse-dnsupdate )
  ( id = 17 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = partner-raw-cltt )
  ( id = 18 )
  ( base-type = AT_DATE )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = client-class )
  ( id = 19 )
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
```

```
}
{
  ( name = status-code )
  ( id = 20 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = status-code )
      ( id = 0 )
      ( base-type = AT_SHORT )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = status-message )
      ( id = 0 )
      ( base-type = AT_NSTRING )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = dns-info )
  ( id = 21 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = flags )
      ( id = 0 )
      ( base-type = AT_SHORT )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = host-label-count )
      ( id = 0 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = name-number )
      ( id = 0 )
      ( base-type = AT_INT8 )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = base-time )
  ( id = 22 )
  ( base-type = AT_DATE )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = relationship-name )
  ( id = 23 )
```

```
  ( base-type = AT_NSTRING )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = protocol-version )
  ( id = 24 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = mclt )
  ( id = 25 )
  ( base-type = AT_INT )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
}
{
  ( name = dns-removal-info )
  ( id = 26 )
  ( base-type = AT_BLOB )
  ( flags = AF_IMMUTABLE )
  ( sepstr = , )
  ( option-list = [
    {
      ( name = host-name )
      ( id = 1 )
      ( base-type = AT_RDNSNAME )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = zone-name )
      ( id = 2 )
      ( base-type = AT_DNSNAME )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = flags )
      ( id = 3 )
      ( base-type = AT_SHORT )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = forward-dnsupdate )
      ( id = 4 )
      ( base-type = AT_NSTRING )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
    {
      ( name = reverse-dnsupdate )
      ( id = 5 )
      ( base-type = AT_NSTRING )
      ( flags = AF_IMMUTABLE )
      ( sepstr = , )
    }
  ] )
}
{
  ( name = max-unacked-bndupd )
```

```
                ( id = 27 )
                ( base-type = AT_INT )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = receive-timer )
                ( id = 28 )
                ( base-type = AT_INT )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = hash-bucket-assignment )
                ( id = 29 )
                ( base-type = AT_BLOB )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = partner-down-time )
                ( id = 30 )
                ( base-type = AT_DATE )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = next-partner-lifetime )
                ( id = 31 )
                ( base-type = AT_DATE )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = next-partner-lifetime-sent )
                ( id = 32 )
                ( base-type = AT_DATE )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
              {
                ( name = client-oro )
                ( id = 33 )
                ( base-type = AT_SHORT )
                ( flags = AF_IMMUTABLE )
                ( repeat = ZERO_OR_MORE )
                ( sepstr = , )
              }
              {
                ( name = requested-prefix-length )
                ( id = 34 )
                ( base-type = AT_INT8 )
                ( flags = AF_IMMUTABLE )
                ( sepstr = , )
              }
            ] )
          }
        ] )
      }
    ] )
}
```

ZTP デバイスブートストラップのトリガー

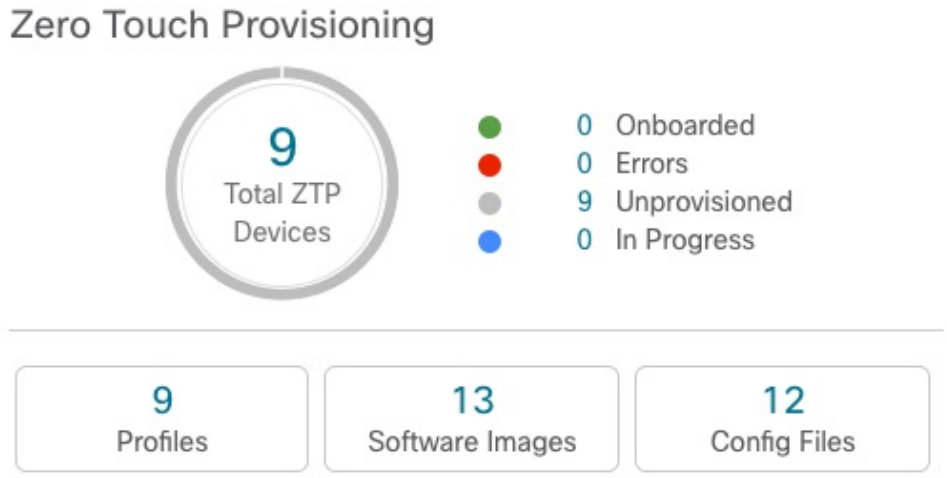
Cisco Crosswork にインポートされたデバイスエントリと DHCP が設定されている場合は、各デバイスを再起動することで ZTP 処理を開始できます。

ステップ 1 次のいずれかのオプションを使用して、ZTP 処理を開始します。

- デバイスの電源を再投入して再起動します。
- ピンを使用して、デバイスの背面にあるシャーシリセットボタンを押します。15 秒間、またはデバイスの電源ライトが点滅し始めるまで押します。
- 以前にイメージ化したデバイスの場合は、Telnet 経由でデバイスに接続し、**ztp initiator** コマンドを発行します。

このセッション中にプロビジョニングする予定のデバイスごとに、必要に応じてこの手順を繰り返します。単一のセッションでデバイスエントリとしてアップロードしたすべてのデバイスを再起動する必要はありません。

ステップ 2 次の図に示すゼロ タッチプロビジョニング ステータス タイルを使用して、ZTP の進行状況を監視します。タイルを表示するには、メインメニューの [ホーム（Home）] アイコンをクリックします。



タイルには、現在の ZTP 処理ステータスの概要ビューが表示されます。現在使用中のすべての ZTP プロファイル、イメージ、および設定ファイルの数を示します。また、タイルには、可能性がある ZTP 処理状態ごとのデバイスの数も表示されます。

オンボーディング済み ZTP デバイス情報の入力

ZTP デバイスは、オンボーディングされると、自動的に Cisco Crosswork の共有デバイスインベントリに組み込まれます。他のデバイスと同様に編集できます。次の手順では、ZTP を使用してオンボーディングされたデバイスに情報を追加する 2 つの方法について説明します。

デバイスを編集する前に、変更するデバイスの CSV バックアップをエクスポートすることをお勧めします。これは、手順 2 で説明するエクスポート機能を使用して実行できます。

始める前に

完全なデバイス インベントリ レコードに必要な一部の情報が不要であるか、または自動化によって利用できません。たとえば、地理的データで、デバイスが建物内の特定の住所または GPS 座標のセットにあることを示すデータなどです。このようなロケーションデータは、アクティブなネットワークを持つほとんどの組織の要件であり、人間のオペレータによってのみ追加できます。

その他の種類のインベントリ情報は、他のアプリケーションを使用してネットワークを管理する場合に役立ちます。たとえば、Cisco Crosswork タグを使用すると、Cisco Crosswork Health Insights の b KPI を特定のデバイスに簡単に適用できます。同様に、SRE ポリシーをデバイスに関連付けると、Cisco Crosswork Network Controller または Cisco Crosswork Optimization Engine をより簡単に使用できるようになります。Cisco NSO などの一部の Cisco Crosswork プロバイダは、この種の拡張デバイス情報に基づいて便利な機能を提供します。すべては人間による更新が必要です。

他の Cisco Crosswork アプリケーションとプロバイダの機能を使用して、このような情報を追加できます。このトピックの詳細については、アプリケーションのユーザーズマニュアルを参照してください。Cisco Crosswork ZTP を使用して、情報の多くを追加することもできます。

ステップ 1 ZTP デバイスのインベントリレコードを更新するには、次の手順を実行します。

a) メインメニューから **[デバイス管理（Device Management）]** > **[ネットワークデバイス（Network Devices）]** を選択します。
b) [ZTP デバイス（ZTP Devices）] タブをクリックします。
c) 変更するデバイスを選択し、 をクリックします。
d) [ステータス（Status）] フィールドの値を [プロビジョニングなし（Unprovisioned）] に変更します。
e) 必要に応じて、デバイスに設定されている他の値を編集します。
f) [保存（Save）] をクリックします。

ステップ 2 ZTP を使用してオンボーディングされたデバイスを含め、デバイスのインベントリレコードを一括で更新するには、次の手順を実行します。

a) メインメニューから **[デバイス管理（Device Management）]** > **[デバイス（Devices）]** を選択します。
b) をクリックします。CSV ファイルを保存します。
c) 選択したアプリケーションで CSV テンプレートを開き、追加または更新するデバイス情報を編集します。更新しないデバイスの行を削除することをお勧めします。
d) 完了したら、編集した CSV ファイルを保存します。

e) 必要に応じて、**[デバイス管理（Device Management）]** > **[デバイス（Devices）]** を選択し、[ゼロタッチデバイス（Zero Touch Devices）] タブをクリックします。
f) をクリックします。
g) [参照（Browse）] をクリックし、作成した CSV ファイルに移動してそのファイルを選択します。
h) CSV ファイルを選択した状態で、[インポート（Import）] をクリックします。

オンボーディング済み ZTP デバイスの再設定

Cisco Crosswork ZTP の目的は、新しいデバイスのエキスパートを現場に配置することなく、新しいデバイスを迅速かつ簡単にオンボーディングすることです。ZTP は、そのタスクの一部としてイメージ化と設定を実行し、デバイス設定の一部としてスクリプトを実行します。ただし、汎用のデバイス設定ユーティリティとして設計されていないため、このような使い方はしないでください。

ZTP を使用してオンボーディングしたデバイスを再設定する必要がある場合は、次を使用します。

- Cisco Crosswork Change Automation Playbook。オンデマンドでデバイスに設定変更を展開できます。
- Cisco Network Services Orchestrator（Cisco NSO）または使用している Cisco Crosswork の他のプロバイダの設定変更機能。
- デバイスとデバイスの OS コマンド ライン インターフェイスへの直接接続。

これらの方法のいずれも使用できない場合は、デバイスを削除するのが最善の方法です。正しい設定を使用すれば、デバイスを再度オンボーディングできます。

ZTP デバイスを削除するには、**[デバイス管理（Device Management）]** > **[デバイス（Devices）]** > **[ゼロタッチデバイス（Zero Touch Devices）]** を選択し、テーブル内のデバイスを選択して をクリックします。

ZTP を使用してオンボーディングしたデバイスの廃止と交換

ZTP を使用してオンボーディングされたシスコのデバイスの廃止が必要な場合があります。デバイスライセンスは、オンボーディング時に入力したデバイスのシリアル番号に関連付けられます。ZTP では、1 台のデバイスを最大 3 つの異なるシリアル番号に関連付けることができます。この事実を使用して、ネットワークと Cisco Crosswork インベントリから障害が発生したデバイスまたは古いデバイスを削除できます。追加のライセンスを消費することなく、後で置き換えることができます。

このルールは、シャーシを備えたデバイスだけでなく、ラインカードやその他の着脱可能なデバイスモジュールにも適用されます。これらの各モジュールには、独自のシリアル番号があります。モジュールの RMA が必要な場合は、古いライセンスを新しいモジュールのシリアル番

号に関連付けます。ただし、次の手順に従って、インベントリから古いラインカードとそのシリアル番号を削除します。

1. **[デバイス管理（Device Management）] > [デバイス（Devices）] > [ゼロタッチデバイス（Zero Touch Devices）]** を選択します。
2. テーブルで古いデバイスを見つけ、そのシリアル番号を記録します。
3. デバイスを選択し、🗑 をクリックして削除します。

   デバイスを削除した後も、Cisco Crosswork はこのシリアル番号に関連付けられたライセンスを消費済みとしてカウントします。新しいデバイスまたは RMA 交換デバイスの購入の一部としてこのライセンスを追跡し、アクティブな使用のために古いデバイスのライセンスを戻すことができます。

   Cisco Crosswork では、同じライセンスを持つアクティブなデバイスを 2 台設定することはできません。新しいデバイスまたは交換用デバイスをオンボーディングする前に、古いデバイスを削除する必要があります。
4. 新しいデバイスをオンボーディングする場合は、次の手順を実行します。
   1. 新しいデバイスの ZTP デバイスエントリを作成する場合は、新しいシリアル番号と古いシリアル番号の両方を入力します。
   2. セキュア ZTP を使用している場合は、新しいデバイスの所有権バウチャー要求とともに、古いデバイスと新しいデバイスの両方のシリアル番号を送信します。シスコは、再生成された所有権バウチャーの使用中のライセンスに、古いシリアル番号と新しいシリアル番号を関連付けます。
   3. 他の ZTP デバイスと同様に、新しいデバイスをオンボーディングします。古いデバイスライセンスのみが使用されます。

ZTP アセットのハウスキーピング

ZTP によるデバイスのオンボーディングが完了したら、アセンブルした ZTP アセットの一部のオフラインコピーを削除できます。組織のポリシーとベストプラクティスに応じて、他のユーザーを保持します。推奨事項：

- [ZTP プロファイル（ZTP profiles）]：通常は、オンボーディングの完了後に ZTP プロファイルを削除しても安全です。ZTP プロファイルを削除するには、**[デバイス管理（Device Management）] > [ゼロタッチプロファイル（Zero Touch Profiles）]** を選択します。削除する ZTP プロファイルを表すタイルで、⋯ をクリックし、ドロップダウンメニューから [削除（Delete）] を選択します。
- [ZTP デバイスエントリ CSV ファイル（ZTP device entry CSV file）]：このファイルのオフラインコピーを保持してテンプレートとして使用することができます。このファイルは、同じネットワークアーキテクチャとデバイスタイプを共有するブランチオフィスが多数ある場合に便利です。それ以外の場合は、ファイルシステムから削除できます。CSV ファイルテンプレートはいつでもダウンロードできます。オンボーディング後に入力したデータ

を含む、ZTP デバイスのすべてのデータが含まれているバックアップ CSV ファイルをエクスポートすると便利な場合があります。CSV デバイスのバックアップをエクスポートするには、**[デバイス管理（Device Management Devices）]** > **[デバイス（Devices）]** > **[ゼロタッチデバイス（Zero Touch Devices）]** を選択します。次に、 をクリックして CSV ファイルを保存します。

- [ソフトウェアイメージと SMU（Software images and SMUs）]：これらのファイルの実稼働バージョンをオフラインで保存し、組織のポリシーに従って古いバージョンを削除します。同じファミリの複数のデバイスをイメージ化するために使用する場合は、アップロードしたイメージファイルを Cisco Crosswork から削除しないでください。古いイメージを削除するには、**[デバイス管理（Device Management）]** > **[ソフトウェアイメージ（Software Images）]** を選択し、テーブル内のファイルを選択して、 をクリックします。
- [設定ファイル（Configuration files）]：すでに Cisco Crosswork にアップロードしている設定を保持する必要はありませんが、組織のポリシーが異なる場合があります。ZTP を使用して同じファミリのデバイスをさらに設定する場合は、アップロードした設定ファイルを削除しないでください。設定が変更された場合は、保存されているバージョンを簡単に更新できます。新しい設定ファイルまたはスクリプトを作成し、**[デバイス管理（Device Management）]** > **[設定ファイル（Configuration Files）]** を選択し、テーブル内のファイルを選択して、 をクリックします。次に、作成した新しいスクリプトファイルを参照し、新しい設定をコピーして貼り付けることができます。設定が古くなった場合は削除します。**[デバイス管理（Device Management）]** > **[設定ファイル（Configuration Files）]** を選択し、テーブル内のファイルを選択して、 をクリックします。
- [クレデンシャルプロファイル（Credential profiles）]：インポートしたクレデンシャルプロファイルの CSV ファイルはすぐに削除できます。アップロードされているクレデンシャルプロファイルは削除しないでください。ユーザー名とパスワードを変更した場合は、クレデンシャルプロファイルを更新します。**[デバイス管理（Device Management）]** > **[クレデンシャル（Credentials）]** を選択し、テーブル内のクレデンシャルプロファイルを選択して、 をクリックします。

ZTP の問題のトラブルシューティング

Cisco Crosswork ZTP のプロビジョニングとオンボーディングは迅速かつ自動的に行われますが、エラーや問題が発生します。次のトピックでは、一般的な問題を解決する方法について説明します。

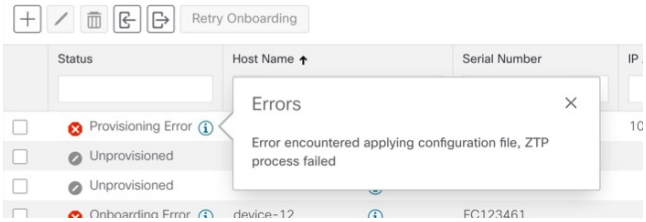
Cisco Crosswork ZTP を使用してオンボーディングできるサードパーティ製デバイスは、セキュア ZTP RFC に 100% 準拠しているサードパーティ製デバイスのみです。

ステータスエラーの検査

[ゼロタッチデバイス（Zero Touch Devices）] ウィンドウの [ステータス（Status）] 列には、ZTP 処理が [プロビジョニングエラー（Provisioning Error）]、[オンボーディングエラー（Onboarding Error）]、または（セキュア ZTP の場合のみ）[ZTP エラー（ZTP Error）] で終了したすべての

デバイスエントリの横に ⓘ が表示されます。アイコンをクリックすると、エラーに関する情報を示すポップアップウィンドウが表示されます。次に例を示します。ポップアップウィンドウの表示が終了したら、✕をクリックして閉じます。

図 *24 : [プロビジョニングエラー（Provisioning Error）]* ポップアップウィンドウ



イメージファイルのアップロード時のエラー

ファイルの MD5 チェックサムが正しいことを確認します。ファイル情報が正しい場合でも、ネットワーク接続が遅いためにイメージのアップロードが失敗する可能性があります。この問題が発生している場合は、アップロードを再試行します。

ZTP デバイスエントリまたは ZTP プロファイルの作成時に、アップロードされたイメージと設定ファイルがドロップダウンメニューに表示されない

ドロップダウンメニューでは、デバイスエントリまたは ZTP プロファイルで指定したデバイスファミリとリリース番号に基づいてイメージと設定ファイルを選択します。ファイル情報が、使用しているデバイスエントリまたはプロファイルの情報と一致していることを確認します。

デバイスのインポート時のエラー

インベントリ内のデバイスにインポートするデバイスと同じシリアル番号がある場合は、インポートする前にデバイスが [プロビジョニングなし（Unprovisioned）] 状態であることを確認します。CSV ファイルを使用してインポートしたすべてのデバイスのステータスは、インポート時に [プロビジョニングなし（Unprovisioned）] に設定されます。インポートする前に、CSV ファイルに記載されている設定、イメージ、および ZTP プロファイルが存在することを確認します。デバイスの CSV ファイルをエクスポートし、変更を加えて再インポートすることで、デバイスイメージファイルと設定ファイルを編集できます。この編集方法を使用する場合は、インポート前に CSV ファイルに正しい UUID があることを確認します。

イメージファイルのダウンロードに失敗した

Cisco Crosswork とデバイス間にネットワーク接続があることを確認します。デバイスが IP アドレスを DHCP サーバーから取得していることを確認します。DHCP サーバーの設定ファイルで指定されたソフトウェアイメージの UUID が正しいことを確認します。設定ファイルで指定されたイメージ UUID を修正する必要がある場合は、ZTP 処理を再度開始する前に DHCP サーバーを再起動してください。

設定ファイルのダウンロードに失敗した

Cisco Crosswork とデバイス間にネットワーク接続があることを確認します。デバイスが IP アドレスを DHCP サーバーから取得していることを確認します。DHCP サーバーの設定ファイルで指定されたソフトウェアイメージの UUID が正しいことを確認します。DHCP 設定ファイルで指定されたイメージ UUID を修正する必要がある場合は、ZTP 処理を再度開始する前に DHCP サーバーを再起動してください。デバイスのシリアル番号がデバイスのシャーシのシリアル番号と一致していることを確認します。ZTP 処理を開始する前に、デバイスのステータスが [プロビジョニングなし（Unprovisioned）] か、または [進行中（In Progress）] であることを確認します。デバイスが他の状態である限り、設定のダウンロードは失敗し続けます。

デバイスの状態が [オンボーディング済み（Onboarded）] と表示され、[プロビジョニング済み（Provisioned）] と表示されない

[プロビジョニング済み（Provisioned）] は、ZTP 処理の中間状態です。デバイスの状態が [プロビジョニング済み（Provisioned）] に変わると、Cisco Crosswork はすぐにデバイスのオンボーディングを試みます。ステータスが [オンボーディング済み（Onboarded）] か、または [オンボーディングエラー（Onboarding Error）] に変わります。

オンボーディングエラー

デバイスを一意に識別するためのデフォルトの Cisco Crosswork デバイスライフサイクル管理（DLM）ポリシーは、IP アドレスです。既存のデバイスと一致する IP アドレスを持つ新しいデバイスをインポートすると、デバイスのステータスが [プロビジョニング済み（Provisioned）] に変わり、その後、[オンボーディングエラー（Onboarding Error）] に変わります。新しいデバイスの IP アドレスが空白の場合、同じ結果が得られます。インストールで OSPF ID、ISIS ID、またはその他の DLM ポリシーを使用してデバイス ID を決定する場合も、同じ問題が発生します。オンボーディングは、すべての DLM ポリシーフィールドに一意の空白以外の値を入力した場合にのみ成功します。オンボーディングが失敗した場合は、ポップアップエラーメッセージを調べて、対応するフィールドを更新し、オンボーディングを再試行します。