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 （1 ページ）
- ZTP 設定のワークフロー （14 ページ）
- ZTP プロビジョニングのワークフロー （47 ページ）

ゼロタッチプロビジョニングの概念

Cisco Crosswork Zero Touch Provisioning（ZTP）アプリケーションでは、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の場所に出荷し、物理的に設置した後でプロビジョニングすることができます。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。デバイスは自動的にソフトウェアイメージと設定をダウンロードし、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。
- **PnP 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、シスコデバイスに適用するセキュアなプロセス。Cisco Plug and Play（Cisco PnP）を使用してデバイスを検証し、セキュアで暗号化されたチャネルを介して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、デバイスをライセンスなしで 90 日間オンボーディングできます。この評価期間が終了すると、ZTP を使用して以前にオンボーディングしたすべてのデバイスと、予定している今後のニーズをカバーするのに十分なキャパシティを備えたライセンスバンドルを購入してインストールするまでは、ZTP を使用して新しいデバイスをオンボーディングすることはできません。

- **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イメージのサポートされているバージョンです。ソフトウェアイメージのインストールは、ZTP処理ではオプションの部分となります。インストールするように設定されている場合、ZTP プロセスは Cisco Crosswork からデバイスにイメージをダウンロードします。デバイスによってそのインストールが行われます。SMU もインストールする必要がある場合、ZTP はクラシック ZTP とセキュア ZTP の設定処理の一部としてそれらをインストールできます（SMU は PnP ZTP ではサポートされていません）。

- **Cisco Plug and Play（Cisco PnP）**：シスコ独自のゼロタッチ プロビジョニング ソリューションで、ほとんどのIOS ソフトウェアイメージにバンドルされています。Cisco PnP は、ソフトウェア PnPエージェントと PnPサーバーを使用して、デバイスにイメージと設定を配布します。通信の安全を確保するために、サーバーとエージェントは HTTPS を使用して通信します。

- **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。使用する予定のZTPモードに応じて、ファイルは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 コマンドになります（これらのすべてがすべての ZTP モードでサポートされているわけではありません）。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。セキュア ZTP は、最大 3 つの異なる設定ファイルもサポートします。これらの設定ファイルは、事前設定、Day 0、および設定後の順序でオンボーディング中に適用されます。

- **設定の処理方法**：セキュア ZTP ユーザーオプション。新しい設定を既存のデバイス設定にマージするか、または上書きするかを指定できます。セキュア ZTP を実装している場合にのみ使用できます。

- **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、従来の ZTP 処理の通常部分にすぎません。
- **所有者証明書**：組織の認証局（CA）署名入り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。所有者証明書は、セキュア ZTP 処理の一部としてデバイスにインストールします。
- **所有権バウチャー**：所有権バウチャーは、デバイスに保存されている所有者証明書を検証することにより、デバイスの所有者を識別するために使用されます。シスコは、組織からの要求に応じて所有権バウチャーを提供します。
- **Cisco PnP エージェント**：Cisco IOS-XE デバイスに組み込まれたソフトウェアエージェント。PnP エージェントをサポートするデバイスは、スタートアップ設定ファイルなしで初めて電源が投入されるたびに、Cisco PnP サーバーを検索しようとします。このエージェントは、DHCP や DNS など、さまざまな方法でサーバーの IP アドレスを検出できます。
- **Cisco PnP サーバー**：ソフトウェアイメージおよび設定の管理と Cisco PnP 対応デバイスへの配布を行う中央サーバー。Cisco Crosswork ZTP には、HTTPS を使用して PnP エージェントと通信するように設定された PnP サーバーが組み込まれています。
- **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、デバイスの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、検証済みのルータだけです。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。クラシック ZTP とセキュア ZTP では、DHCP ブートファイル URL のソフトウェアイメージファイルの UUID を使用します。
- **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 がイメージと設定ファイルを保存する場所。

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）520 および 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（ただし、このリリースではサポートされていないリリース 7.3.2 と 7.4.1 を除く）。

  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
  - Cisco NCS 1000 ～ 1004 シリーズ
  - Cisco NCS 5500 シリーズ
  - Cisco NCS 8000 シリーズと 8800 シリーズ（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「Secure ZTP：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。

PnP 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 PnP ZTP がサポートされています。

- **ソフトウェア**：Cisco IOS-XE バージョン 16.12、17.4.1、17.5.1. お客様に推奨されるバージョンは、バージョン 16.12.5 です。
- **ハードウェア**：
  - Cisco Network Convergence Systems（NCS）520 シリーズ ルータ
  - Cisco アグリゲーション サービス ルータ（ASR）903
  - Cisco ASR 907
  - Cisco ASR 920

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

PnP ZTP を使用する場合は、ZTP 処理をトリガーする**前**に、各 IOS-XE デバイスの最小ライセンスブートレベルが **metroipaccess** または **advancedmetroipaccess** に設定されていることを確認します。ブートレベルが正しく設定されている場合、デバイスの IOS-XE `#sh run | sec license` CLI コマンドの出力に、2 つのライセンスレベル、`license boot level advancedmetroipaccess` または `license boot level metroipaccess` のいずれかを示すステートメントが含まれている必要があります。コマンド出力に他のライセンスレベル、特にこれらのライセンスレベルより低いライセンスレベルが示されている場合は、Cisco PnP の暗号化機能が有効になりません。これにより、証明書のインストールが失敗して PnP ZTP デバイスのプロビジョニングが失敗します。

セキュア ZTP：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が同じ形式のファイル名で OV ファイルを生成す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

ベストプラクティスとして、使用するデバイスに最も安全な実装を常に選択してください。ただし、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 と PnP 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 （4 ページ）」を参照してください。

- **PnP ZTP を使用する場合**：Cisco PnP プロトコルをサポートする Cisco IOS-XE デバイスのセキュアプロビジョニングの設定が必要な場合は、PnP ZTP を使用します。設定はセキュア ZTP よりも簡単ですが、クラシック ZTP よりは若干複雑です。そのため、ネットワークデバイスがこれらの基本要件を満たしている場合に最適です。
- **イメージデバイスで ZTP を使用**：ZTP モードのいずれかを使用する場合、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。

  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。クラシック ZTP モードとセキュア ZTP モードの両方で設定ファイルをチェーンできますが、追加のスクリプトを実行するクラシック ZTP の機能は、特定のデバイスで許可されるスクリプトの実行のサポートに制限されます。PnP ZTP は CLI コマンドのみを実行でき、スクリプトを実行することはできません。

  いずれの場合も、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された後は、ZTP を使用してデバイスを再設定することは避けてください（詳細については、「オンボーディング済み ZTP デバイスの再設定 （73 ページ）」を参照してください）。
- **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。

  一貫性を強化するもう 1 つの方法は、同様のタイプのデバイス用に基本的なスクリプト設定を作成し、スクリプトロジックを使用して、特別なロールを持つデバイス用の他のスクリプトを呼び出す（チェーンする）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。この例を拡張すると、そのスクリプトは共通の設

定を適用し、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て適用します。セキュア ZTP を使用する場合は、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 の処理ロジック

Cisco Crosswork ZTP の処理は、クラシック ZTP、セキュア ZTP、または PnP ZTP のいずれを実装するかによって異なります。このトピックの次のセクションでは、各 ZTP モードの ZTP 処理の各ステップについて詳しく説明します。

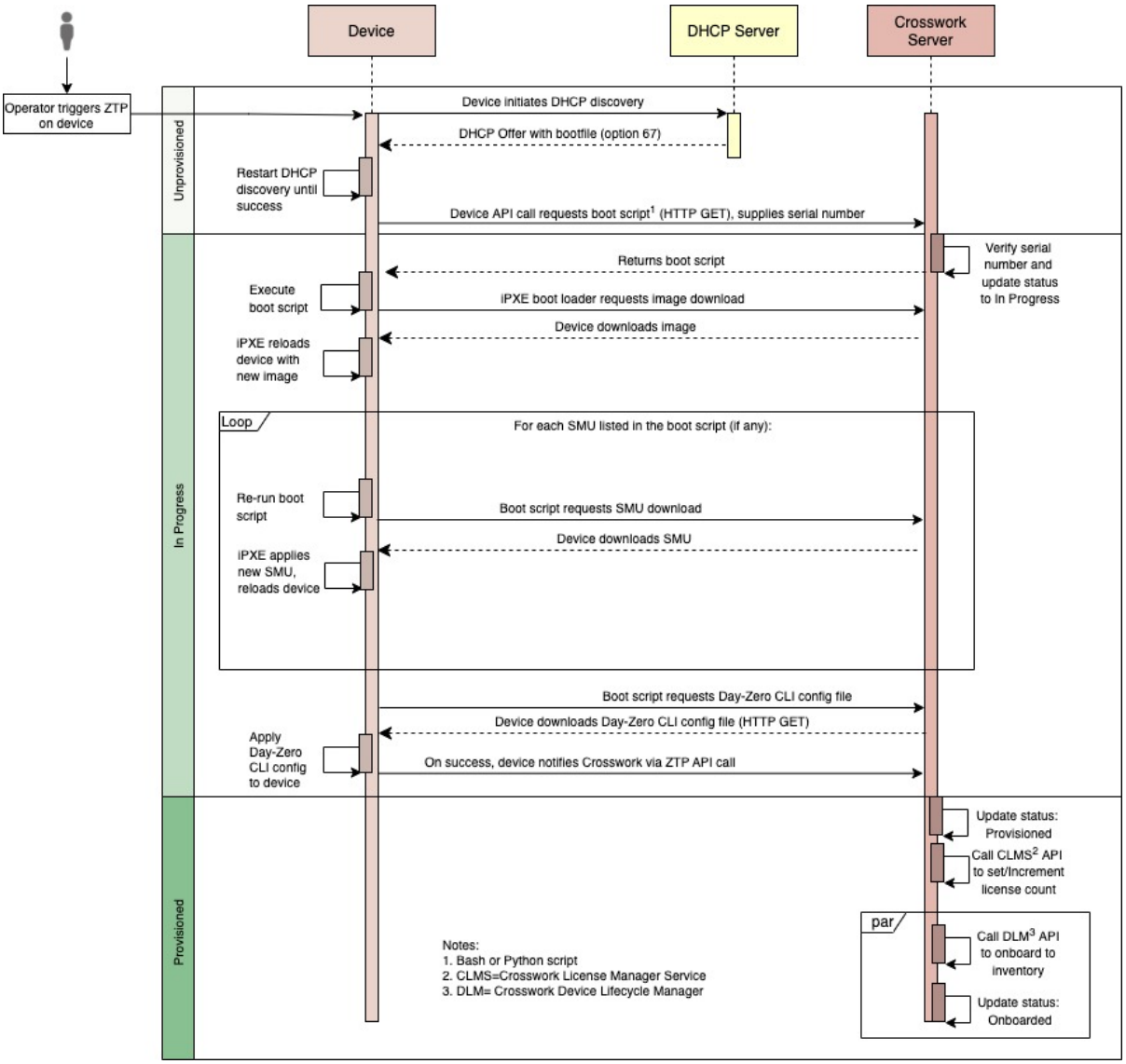
デバイスのリセットまたはリロードによって開始されると、ZTP プロセスは自動的に進行します。また、Cisco Crosswork は、[ゼロタッチデバイス（Zero Touch Devices）] ウィンドウを更新し、各デバイスで処理が完了したときに示すステータスメッセージも表示します。各セクションの図は、これらの状態遷移を、各図の左側にある緑の陰影のブロックで示しています。導入準備状態に到達するのは ZTP 処理の最後にのみ発生するため、導入完了状態への遷移は表示されません。

図で示されているように、ZTP で使用する構成スクリプトは、Cisco Crosswork 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。構成がこれに失敗した場合、Crosswork は発生した状態の変更を登録できず、ZTP の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 の処理

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。

図 *1* : クラシック *ZTP* の処理

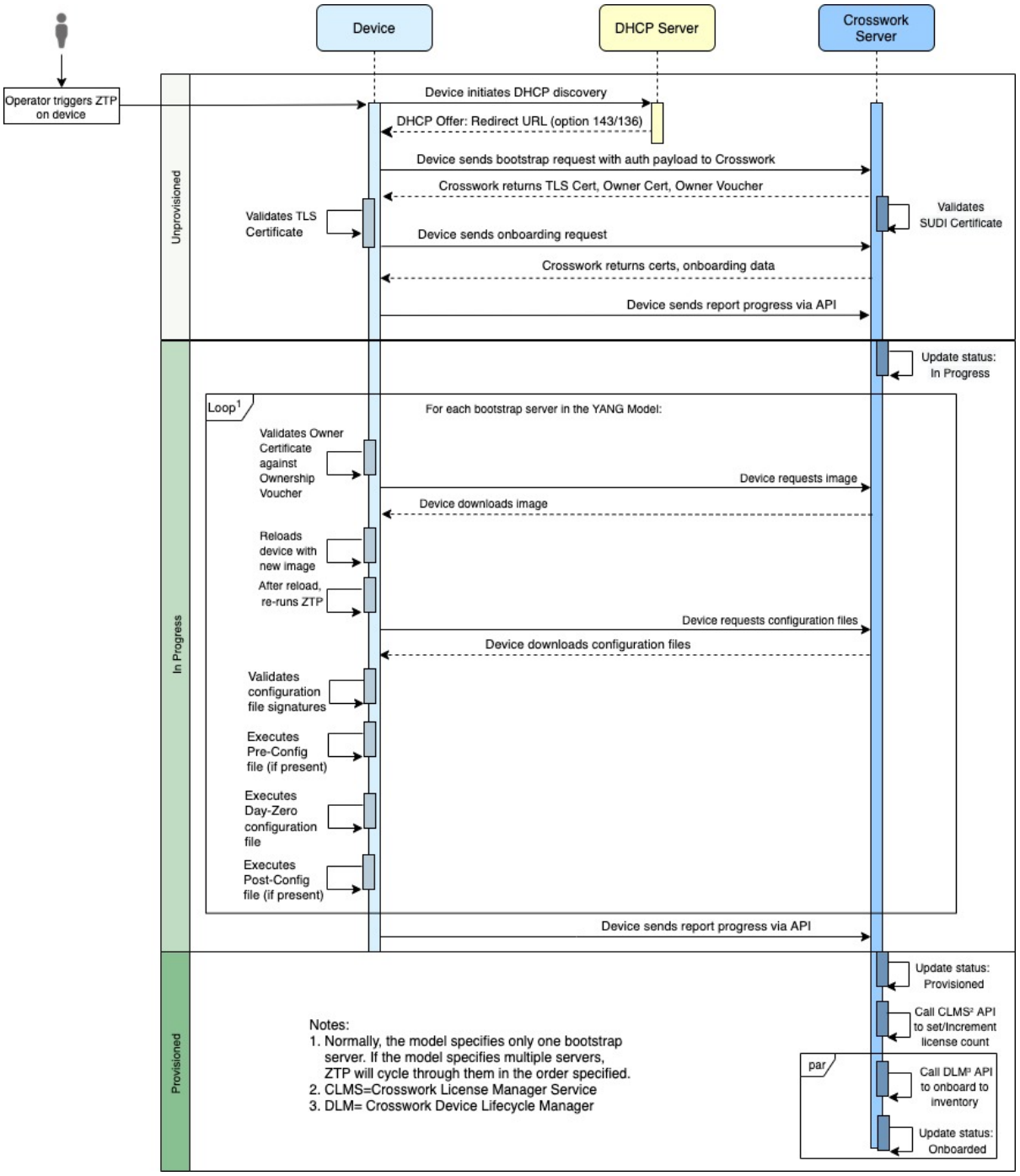


DHCPサーバーは、デバイスのシリアル番号に基づいてデバイスのアイデンティティを確認してから、ブートファイルとイメージのダウンロードを提供します。ZTPがデバイスをイメージ化すると、デバイスは設定ファイルをダウンロードし、実行します。

セキュア ZTP の処理

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。

図 *2:* セキュア *ZTP* の処理



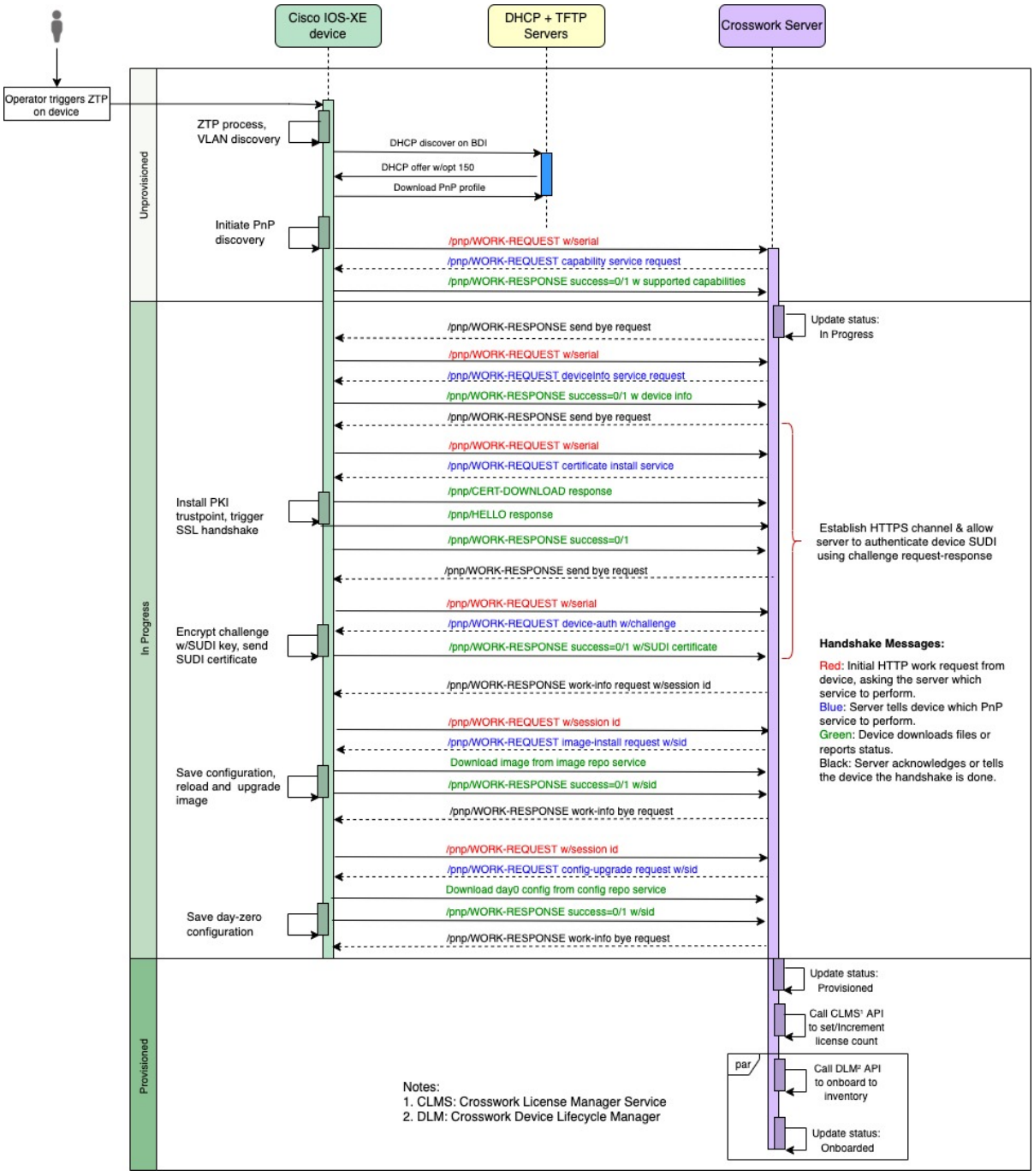
デバイスと ZTP ブートストラップサーバーは TLS/HTTPS を介してデバイスとサーバー証明書でセキュアな一意のデバイス識別子（SUDI）を使用し、相互に認証します。セキュアな HTTPS チャネルを介して、ブートストラップサーバーはデバイスに署名付きイメージと設定アーティファクトをダウンロードさせます。これらのアーティファクトは、RFC 8572 YANG スキーマ（https://tools.ietf.org/html/rfc8572#section-6.3）に準拠する必要があります。デバイスは新しい

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。

PnP ZTP の処理

次の図に、PnP 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。

図 *3 : PnP ZTP* の処理



オペレータが PnP ZTP 処理をトリガーすると、デバイスは VLAN 検出を実行し、DHCP 検出が開始される BDI インターフェイスを作成します。DHCP 検出の一部として、デバイスは DHCP オプション 150 設定を使用して外部 TFTP サーバーの IP アドレスも取得します。デバイスは、認証なしで TFTP サーバーから PnP プロファイルをダウンロードし、デバイスの実行コンフィギュレーションにコピーします。PnP プロファイルは CLI テキストファイルです。プロファイ

ルは、デバイスの PnP エージェントをアクティブにし、ポート 30620 上で組み込み Crosswork PnP サーバーに作業要求を HTTP 経由で送信します。次に、PnP サーバーはデバイスのシリアル番号を Crosswork のシリアル番号の「許可」リスト（以前に Crosswork にアップロードしたもの）と照合して検証し、PnP 機能サービス要求を開始します。デバイスからの PnP 作業応答が成功すると、デバイスのプロビジョニングステータスが [プロビジョニングなし（Unprovisioned）] から [進行中（In Progress）] に変更されます。その後、PnP サーバーは、デバイス情報、証明書のインストール、イメージのインストール、設定のアップグレードなどの要求を含む一連のサービス要求を開始します。これらの各サービス要求には、PnP サーバーと PnP エージェント間の 4 ウェイハンドシェイクが含まれます。証明書インストール要求の一部として、Crosswork PnP サーバーはその証明書をデバイスと共有します。デバイスにこのトラストポイントを正常にインストールすると、PnP プロファイル設定が変更され、Crosswork で HTTPS とポート 30603 の使用が開始されます。後続のイメージと設定のダウンロード要求は、HTTPS を使用してトランザクションを保護します。現在、デバイスでは SUDI 証明書認証はサポートされていません。デバイスが新しいイメージ（存在する場合）をダウンロードしてインストールし、リロードすると、PnP プロセスは引き続き CLI 設定ファイルをダウンロードし、デバイスの実行コンフィギュレーションに適用します。デバイスのステータスが [プロビジョニング済み（Provisioned）] に設定され、ライセンス数が Crosswork で更新されます。デバイスのステータスは [オンボーディング済み（Onboarded）] に設定され、デバイスは PnP サーバーとの通信を停止します。

ZTP と評価ライセンス

すべての Cisco Crosswork アプリケーションは、ライセンスなしで 90 日間使用できます。ユーザーがシステムにログインするたびに、Crosswork はトライアル期間の残りの日数を示すバナーを表示します。トライアルが期限切れになると、バナーにその旨が表示されます。その時点で、それ以上のデバイスでは ZTP オンボーディングプロセスを完了できなくなります。ZTP ライセンスは、ブロック単位で販売されるライセンスによる消費ベースモデルに従います。ZTP を使用してデバイスをオンボーディングする機能を取り戻すには、トライアル期間中にオンボーディングしたデバイスの数と、今後 ZTP でオンボーディングする予定の新しいデバイスの数の両方をカバーするライセンスブロックをインストールする必要があります。たとえば、トライアル中に 10 台のデバイスをオンボーディングしてから、91 日目に10 台のデバイスのライセンスバンドルをインストールした場合、使用できるライセンスは残りません。別のデバイスをオンボーディングする前に、少なくとも 1 つのライセンスブロックをインストールする必要があります。必要に応じて、ライセンスブロックを追加できます。オペレータは、ライセンスの消費をモニターして、予期せぬライセンス不足を回避する必要があります。使用済みのライセンスの数と、まだ使用可能なライセンスの数を確認するには、Cisco Smart Licensing のサイトを確認します。

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 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。ZTP の前提条件を満たす （14 ページ）を参照してください
2. ZTP が処理に必要とするアセットのタイプを組み立てて Crosswork に読み込ませます。使用したい ZTP モードやオンボーディングするデバイスによっては、最低 3 種類から最高 8 種類のアセットを用意する必要があります。ZTP アセットの組み立てと読み込み （15 ページ）を参照してください
3. オプション：ZTP プロファイルを作成します。これは、オンボーディングプロセス中にデバイスのイメージングと構成を簡素化および標準化するのに役立ちます。ZTP プロファイルの作成 （38 ページ）を参照してください
4. ZTP デバイスエントリを作成します。ZTP は、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ときに、これらのデバイスエントリをデータベースの「アンカー」として使用します。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（「ZTP デバイスエントリのアップロード （47 ページ）」を参照）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（「単一 ZTP デバイスエントリの作成 （46 ページ）」を参照） 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。Crosswork API を使用してデバイスをオンボードすることもできます（Cisco Crosswork DevNet ページの ZTP API リファレンスを参照）。

このセクションの残りのトピックでは、これらの各タスクを実行する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。

- 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 クラスタノードはデバイスから、ノードはデバイスから、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 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。このタイプのアクセスを有効にするには、ファイアウォール設定変更が必要な場合があります。
- Crosswork ZTP を使用してオンボーディングする Crosswork クラスタノードとデバイスがまったく異なるサブネットにある場合は、Crosswork ノードからデバイスサブネットへの 1 つ以上の静的ルートを設定する必要があります。メインメニューからこれを行うには、**[管理（Administration）]** > **[設定（Settings）]** > **[静的ルート（Static Routes）]** を選択します。[+] をクリックし、接続先サブネットの IP アドレスとマスク（スラッシュ表記）を入力して、[追加（Add）] をクリックします。
- PnP ZTP の使用を計画している場合は、TFTP サーバーを Cisco Crosswork プロバイダーとして追加する必要があります。TFTP サーバーは、次のような汎用プロファイルを使用して設定できます。

```
pnp profile test-profile
    transport http ipv4 192.168.100.205 port 30620
```

ZTP アセットの組み立てと読み込み

「ZTP アセット」という用語は、次のチェックリストに示されているソフトウェアと構成ファイル、ログイン情報、証明書、およびその他のアセットを指します。準備して Crosswork に読み込ませる必要があるアセットの数は、使用する ZTP モードに必要かどうか、デバイスのオンボーディングを開始したときのデバイスの状態、およびその他の要因によって異なります。

便宜上、これらのアセットをチェックリストに示されている順序で準備して読み込むことをお勧めします。ソフトウェアイメージなどのオプションのアセットを含む各アセットを準備して読み込む方法の詳細については、チェックリストの最後の列にあるリンクされたトピックを参照してください。

多くの組織は、シリアル番号や構成ファイルなどの ZTP アセット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、デスクトップから簡単にアクセスできるようにしてください。これにより、ZTP の設定を簡単に実行できます。

IOS-XR デバイスでセキュア ZTP を使用する背景については、『*System Setup and Software Installation Guide for Cisco NCS 540 Series Routers, IOS XR Release 7.3.x*』の「Securely Provision Your Network Devices」の章を参照してください。

Cisco Crosswork は、IOS-XR デバイス用に、Cisco Crosswork を認証局として独自の TLS 証明書を提供しています。IOS-XR デバイスは Crosswork TLS サーバー証明書で X.509 検証を実行しないため、独自の TLS CA 証明書チェーンを提供またはアップロードする必要はありません。

表 *1: ZTP* 資産チェックリスト

順序	アセット	クラシック ZTP	セキュア ZTP	PnP ZTP	詳細については、次を参照してください。
1	ソフトウェア イメージ	オプション	オプション	オプション	デバイスにソフトウェアイメージがインストールされていない場合は、ソフトウェアイメージが必要です。ソフトウェアイメージの検索と読み込み（17 ページ）
2	設定	**必須**	**必須作業です**。複数の設定をサポートしています。	**必須**	構成ファイルと読み込み（18 ページ）
3	ソフトウェア メンテナンス アップグレード（SMU）	オプション	オプション	サポート対象外	SMU の検索と読み込み（30 ページ）
4	デバイスのクレデンシャル	**必須**	**必須**	**必須**	ZTP でのクレデンシャル プロファイルの作成（30 ページ）
5	シリアル番号	**必須**	**必須**	**必須**	デバイスのシリアル番号の検索と読み込み （32 ページ）
6	ピン留めされたドメイン証明書（PDC）、所有者証明書（OC）、および所有者キー	未使用	**必須**	未使用	PDC、所有者証明書、および所有者キーを更新する （33 ページ）。
7	所有権バウチャー	未使用	**必須**	未使用	所有権バウチャーのリクエストと読み込み （36 ページ）。

順序	アセット	クラシック ZTP	セキュア ZTP	PnP ZTP	詳細については、次を参照してください。
8	SUDI ルート証明書	未使用	必須	IOS-XE デバイスのみに必要	SUDI ルート証明書の準備と読み込み （37 ページ）

ソフトウェアイメージの検索と読み込み

ソフトウェアイメージは、ネットワークデバイスの機能を可能にする、インストール可能なネットワーク オペレーティング システム ソフトウェア（Cisco IOS-XR、または PnP ZTP の場合は Cisco IOS-XE など）を含むファイルです。

ソフトウェアイメージの読み込みは、すべてのZTPモードでオプションですが、オンボーディングしているデバイスにソフトウェアイメージがインストールされていない場合は必須です。すでにイメージ化されているデバイスにソフトウェアイメージを適用する必要はありません。イメージを読み込まずに、構成ファイルをデバイスに適用することもできます。イメージの読み込みは、オンボードするデバイスにイメージがインストールされていない場合、またはデバイスのオンボードと同時にネットワーク OS をアップグレードする場合にのみ必要です。

シスコは、IOS-XR イメージを TAR、ISO、BIN、または RPM ファイルとして配布しています。シスコでは、IOS-XE イメージを BIN ファイルとして配布しています。各シスコイメージ 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。

[シスコサポート & ダウンロード（Cisco Support & Downloads）] ページからソフトウェア イメージ ファイルをダウンロードします。ダウンロード中に、イメージの MD5 チェックサムを記録します。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。Cisco Crosswork は MD5 チェックサムを使用してソフトウェア イメージ ファイルの整合性を検証します。

ソフトウェア イメージ ファイルを一度に 1 つずつ Cisco Crosswork に読み込ませ、読み込み中に各ソフトウェア イメージ ファイルの MD5 チェックサムを入力します。

ソフトウェアイメージを Cisco Crosswork に読み込ませるには、次の手順を実行します。

1. Cisco Crosswork を起動します。
2. 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。
3. をクリックします。 
4. 入力するか、または [参照（Browse）]をクリックし、アップグレードするコンポーネントのソフトウェア イメージ ファイルを選択しますプロンプトが表示されたら、ファイルの MD5 チェックサムを入力します。
5. [追加（Add）] をクリックして、ソフトウェア イメージ ファイルの追加を終了します。

6. 計画された ZTP 実行で使用されるすべてのソフトウェア イメージ ファイルを読み込むまで、必要に応じて繰り返します。

構成ファイルと読み込み

構成ファイルは、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構成するスクリプトファイルです。これらはすべての ZTP モードに必要です。

クラシック ZTP とセキュア ZTP モードで使用される構成ファイルは、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。Cisco IOS-XR デバイスで、クラシック ZTP とセキュア ZTP のみを使用する場合は、構成ファイルを使用して、SMU を使用してインストールされているネットワーク OS ソフトウェアバージョンをアップグレードすることもできます（「SMU の検索と読み込み （30 ページ）」を参照）。

従来の ZTP は、デバイスごとに 1 つの day-zero 構成ファイルのみをサポートします。セキュア ZTP では、オンボーディング時に最大 3 つの構成ファイルを適用できます。1 つは事前設定の準備用、2 つ目は Day 0 設定またはメイン設定、3 つ目は Day 0 設定後に適用される設定後ファイルです。Day 0 設定のみが必須です。アプリケーションの順序は固定されています。

Cisco PnP ZTP は、Cisco ASR 900 デバイスと Cisco NCS 520 デバイスでは Day 0 設定 TXT ファイルのみをサポートします。PnP ZTP 設定ファイルでは、IOS-XE CLI コマンドを使用する必要があります。PnP ZTP は、Linux シェル（SH）または Python（PY）スクリプトファイルをサポートしていません。

Cisco Crosswork に設定ファイルを 1 つずつアップロードします。

組織またはコンサルタントが構成ファイルを作成します。次のセクションでは、ZTP モードのいずれかを使用してデバイスをオンボーディングするときに使用する構成ファイルを準備するためのガイドラインと、これらのファイルを Cisco Crosswork に読み込む方法について説明します。

設定例ファイルのダウンロード

構成スクリプトファイルの内容は、使用するデバイスと組織での使用方法によって大きく異なります。したがって、このドキュメントでは使用可能なすべてのオプションを完全には説明していません。

覚えておくべき主なガイドラインは次のとおりです。

1. カスタム設定コードは、デフォルトとカスタムの両方の置換可能（または「プレースホルダ」）パラメータを使用できます。これにより、デバイスエントリを一括でインポートするとき、または一度に 1 つずつ作成するときに、[設定属性（Configuration Attributes）] フィールドを使用してランタイム時に値を挿入できます。

2. 必要に応じて、新しいカスタム置換可能パラメータを作成できます。既定のパラメータと同じ名前を使用せず、このトピックで説明する変数の命名規則に従っていれば、任意の名前を付けることができます。デフォルトの置き換え可能なパラメータを使用する場合、デバイスエントリの [構成属性（Configuration Attributes）] フィールドで設定した値の代わり

に、このトピックの「設定ファイルでデフォルトの置き換え可能なパラメータを使用する」セクションで説明されているソースからランタイム値が挿入されます。

3. 置換可能パラメータの名前は大文字と小文字が区別され、中カッコとドル記号を含める必要があります。スペースを含めることはできません（代わりにアンダースコアを使用）。
4. すべてのカスタム置換可能パラメータのランタイム値が[設定属性（Configuration Attributes）]フィールドに指定されていることを確認します。ランタイム値を指定しなかったカスタム置換可能パラメータが 1 つでもある場合は、デバイス設定プロセスが失敗します。
5. セキュア ZTP を使用している場合は、Day 0 設定にのみカスタム置換可能パラメータを使用できます。カスタム置換可能パラメータは、事前設定ファイルと設定後ファイルではサポートされていません。
6. 一部のタスクを実行するには、Cisco Crosswork の API コール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。。
7. どの設定ファイルでも別の設定ファイルをコールして実行できますが（デバイスに正常にダウンロードできる場合）、セキュア ZTP でのみ、初期のセキュアなダウンロードの一環として個別の事前設定ファイル、設定後ファイル、および Day 0 設定ファイルを指定できます。
8. 設定ファイル名に複数のピリオドを含めることはできず、また、スペースの代わりにアンダースコアを使用する必要があります。その他のファイルの制限は、以下で説明する設定例ファイルに記載されています。

置き換え可能なパラメータと API コールの使用方法の例については、Cisco Crosswork ZTP アプリケーションに付属の Cisco IOS-XR デバイスの ZTP 構成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)）] をクリックします。サンプル構成スクリプトにはコメントが付けられており、より一般的に使用される API 呼び出しと置き換え可能なパラメータの例が示されています。

置換可能なパラメータの詳細については、以下のセクション「構成ファイルでのデフォルトの置換可能パラメータの使用」および「構成ファイルでのカスタムの置換可能パラメータの使用」を参照してください。

Crosswork API コールの詳細については、Cisco Crosswork の Cisco Developer Network（DevNet）サイトで利用可能な「Crosswork API References」メニューの ZTP デバイスと設定 API に関する項を参照してください。

次のセクション「サンプル ZTP 構成スクリプト」では、置き換え可能なパラメータと API の使用方法の例を示します。

設定ファイルのプレビュー

以前に Cisco Crosswork にアップロードされた設定ファイルの内容をプレビューするには、**[デバイス管理（Device Management）] > [ZTP 設定ファイル（ZTP Configuration Files）]** を選択

し、設定ファイル名をクリックします。ポップアッププレビューには、次の表に示すように、重要なコード機能のコードシンタックスのスタイルが含まれています。

表 *2: ZTP* 設定ファイルプレビューのコードシンタックスの色

対象のコード機能	表示色
句読点、演算子、エンティティ、URL、変数、クラス名、定数	黒色
コメント	グレー
プロパティ、タグ、ブール値、関数名、シンボル	オレンジ
セレクタ、属性名、文字、組み込み、挿入	深緑
機能	パープル
キーワード、属性値	青
正規表現、重要	茶
文字列	緑
番号、イーサネットアドレス、MACアドレス	マゼンタ

設定ファイルでのデフォルトの置換可能パラメータの使用

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、Cisco Crosswork から設定スクリプトの例をダウンロードしてください（**[デバイス管理（Device Management）] > [ZTP 設定ファイル（ZTP Configuration Files）] > [サンプルスクリプトのダウンロード（XR）（Download Sample Script (XR)）]**）。これらのデフォルトの置き換え可能なパラメータの使用方法を示す例については、このトピックの後半のセクション「サンプル ZTP 構成スクリプト」を参照してください。

表 *3: 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	使用される値
{$HOSTNAME}	ZTP デバイスエントリで指定されているデバイスのホスト名。
{$IP_ADDRESS}	ZTP デバイスエントリで指定されているデバイスの IP アドレス。
{$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）。

Cisco Crosswork が置換するプレースホルダ	使用される値
{$SSH_PASSWORD}	クレデンシャルプロファイルの [パスワード（Password）] フィールドの値（[接続タイプ（Connectivity Type）] が [SSH] の場合）。
{$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）。

設定ファイルでのカスタム置換可能パラメータの使用

次の例に示すように、独自のカスタム置換可能パラメータを設定ファイルに作成できます。この例に示すように、同じ設定ファイル内でカスタムとデフォルトの置換可能パラメータを使用できます。

次の条件を満たしている限り、任意の名前をカスタム置換可能パラメータに割り当てることができます。

- 指定された変数定義形式（{$MyParm} など）に従う。
- パラメータ名のスペースをアンダーライン文字に置き換える。
- デフォルトの置換可能パラメータと同じ名前や大文字を再使用しない。
- デバイスエントリファイルの [設定属性（Configuration Attributes）] フィールドに、各カスタムパラメータの値を入力する。次の CLI 設定のサンプルファイルとそのカスタムパラメータを ZTP デバイスエントリファイルで使用するには、ZTP デバイスエントリファイルの各デバイスの [設定属性（Configuration Attributes）] フィールドで **{$LOOPBACK0_IP}** カスタムパラメータの値を指定する必要があります。カスタムパラメータの値を指定し忘れた場合は、設定が失敗します。

セキュア ZTP を使用している場合は、Day 0 設定ファイルのみでカスタム置換可能パラメータがサポートされます。

このスクリプトの例の最初の行は、IOS-XR デバイスの CLI スクリプトで必要です。これにより、ZTP はファイルが CLI スクリプトか bash/Python スクリプトかを確認できます。必要に応じてバージョン番号を更新してください。IOS-XE デバイスの場合、このような行は必要ありません。

図 *4:* 置換可能パラメータが混在する *IOS-XR CLI* 設定スクリプトの例

```
!! IOS XR Configuration 7.3.1
!
hostname {$HOSTNAME}
username {$SSH_USERNAME}
 group root-lr
 group cisco-support
 password 0 {$SSH_PASSWORD}
!
cdp
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 120
!

call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method http
 !
!
```

```
interface Loopback0
 ipv4 address {$LOOPBACK0_IP} 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
 description OOB Management ZTP
 ipv4 address {$IP_ADDRESS}
!
end
```

ZTP 設定スクリプトの例

このセクションでは、ZTP の構成スクリプトの例を示します。

図 *5 : IOS XR* デバイスのクラシック *ZTP Day 0* 構成スクリプト

```
#!/bin/bash

################################################################################
#
# ztpSampleScriptFile.sh
#
# Purpose: This sample script is required to notify Crosswork of the status of
# ZTP processing on an IOS XR device, and to update the device's IP address and
# hostname in Crosswork. It is also used to download a day0 config file from
# Crosswork config repository and apply this initial configuration to the device.
#
# To use: Modify the sample script as needed, following the comment guidance.
# Then upload the modified script to the Crosswork config repository.
# Next, copy the URL of this file from the repository and set that
# value in the DHCP server boot filename for ZTP config download. When ZTP is
# triggered on the device, it will download and run the script, then notify
# Crosswork.
#
# Replace the following variables with valid values & upload to Crosswork config
# repository. Sample values are provided for reference.
# - XRZTP_INTERFACE_NAME: e.g., MgmtEth0/RP0/CPU0/0 interface where ZTP triggered
# - CW_HOST_IP: Crosswork VM management or data network IP address
# - CW_PORT: 30604 for HTTP & 30603 only for HTTPS download of config file
# - CW_CONFIG_UUID: Replace with UUID of day0 config file from Crosswork repo,
#   assuming user has already uploaded device day-0 config file.
#
# This script has been tested and is known to work on Cisco NCS5501, NCS540l,
# ASR9901, and 8800 routers.
#
################################################################################


export LOGFILE=/disk0:/ztp/customer/user-script.log

XRZTP_INTERFACE_NAME="MgmtEth0/RP0/CPU0/0"
# ZTP helper library is assumed to be installed in IOS-XR linux shell
source /pkg/bin/ztp_helper.sh
interfacedata=$(xrcmd "show interface ${XRZTP_INTERFACE_NAME}")

CW_HOST_IP="<EnterIPv4AddressHere>"
CW_PORT="30604"
CW_CONFIG_UUID="e04661f8-0169-4ad3-82b8-a7c26c4f2565"

# Send logging information to log file on device disk0:/ztp/user-script.log
function ztp_log() {

    echo "$(date +"%b %d %H:%M:%S") "$1 >> $LOGFILE
}
```

```
#
# Get chassis serial number of the device, required by ZTP process.
# This works on Cisco NCS5501, NCS540l, 8800 series routers.
#
function get_serialkey(){

    local sn=$(dmidecode | grep -m 1 "Serial Number:" | awk '{print $NF}');
    if [ "$sn" != "Not found" ]; then
           ztp_log "Serial $sn found.";
           # The value of $sn from dmidecode should be same as serial number
           # of XR device chassis.
           DEVNAME=$sn;
           return 0
    else
        ztp_log "Serial $sn not found.";
        return 1
    fi
}

#
# Get chassis serial number of the device, required by ZTP process.
# This is tested and works on Cisco ASR 9901, but not other devices.
#
function get_serialkey_asr9901(){

     udi=$(xrcmd "show license udi")
     sn="$(cut -d':' -f4 <<<"$udi")"
     pid="$(cut -d':' -f3 <<<"$udi")"
     pid="$(cut -d',' -f1 <<<"$pid")"
     echo "Serial Number $sn"
     echo "product id $pid"
}

#
# Get IP address and subnet mask from device. IP address is assigned from DHCP
# server on interface where ZTP was triggered.
#
function get_ipaddress(){

    local ipvar=($(echo $interfacedata | awk -F "Internet address is " '{sub(/
.*/,"",$2);print $2}'));
    local ipv4addr=$(xrcmd "sh run interface ${XRZTP_INTERFACE_NAME} | i ipv4 address"
| awk '{print $3}')
    local ipv6addr=$(xrcmd "sh run interface ${XRZTP_INTERFACE_NAME} | i ipv6 address"
| awk '{print $3}')
    local ipaddress=($(echo $ipvar | awk -F "/" '{sub(/ .*/,"",$1);print $1}'));
    local mask=($(echo $ipvar | awk -F "/" '{sub(/ .*/,"",$2);print $2}'));
    local maskv6=($(echo $ipv6addr | awk -F "/" '{sub(/ .*/,"",$2);print $2}'));

    ztp_log "### Value of interfacedata => $interfacedata ###"
    ztp_log "### Value of ipvar => $ipvar ###"
    ztp_log "#####IPv4 address $ipaddress and mask $mask found. #####";

    IPADDR=$ipaddress
    MASK=$mask
    MASKV6=$maskv6

    return 0
}

#
# Fetch hostname from device configuration.
#
```

```
function get_hostname(){

    hostnamedata=$(xrcmd "show running-config hostname")
    local hostname=($(echo $hostnamedata | awk -F "hostname " '{sub(/ .*/,"",$2);print
$2}'));

    ztp_log "#####hostname $hostname found.";
    HOSTNAME=$hostname;
    return 0;
}

#
# Download day-0 config file from Crosswork config repository using values
# set for CW_HOST_IP, CW_PORT and CW_CONFIG_UUID.
# The MESSAGE variable is optional, can be used to display a suitable message
# based on the ZTP success/failure log.
#
function download_config(){

    ztp_log "### Downloading system configuration ::: ${DEVNAME} ###";
    ztp_log "### ip address passed value ::: ${IPADDR} ###";
    ip netns exec global-vrf /usr/bin/curl -k --connect-timeout 60 -L -v --max-filesize
 104857600
http://${CW_HOST_IP}:${CW_PORT}/crosswork/configsvc/v1/configs/device/files/${CW_CONFIG_UUID}
 -H X-cisco-serial*:${DEVNAME} -H X-cisco-arch*:x86_64 -H X-cisco-uuid*: -H
X-cisco-oper*:exr-config -o /disk0:/ztp/customer/downloaded-config 2>&1

    if [[ "$?" != 0 ]]; then
        STATUS="ProvisioningError"
        ztp_log "### status::: ${STATUS} ###"
        ztp_log "### Error downloading system configuration, please review the log ###"
        MESSAGE="Error downloading system configuration"
    else
        STATUS="Provisioned"
        ztp_log "### status::: ${STATUS} ###"
        ztp_log "### Downloading system configuration complete ###"
        MESSAGE="Downloading system configuration complete"
    fi
}

#
# Apply downloaded configuration to the device and derive ZTP status based on
# success/failure of ZTP process. The MESSAGE variable is optional, can be used
# to display a suitable message based on the ZTP success/failure log.
#
function apply_config(){
    ztp_log "### Applying initial system configuration ###";
   xrapply_with_reason "Initial ZTP configuration" /disk0:/ztp/customer/downloaded-config
 2>&1 >> $LOGFILE;
    ztp_log "### Checking for errors ###";
    local config_status=$(xrcmd "show configuration failed");
    if [[ $config_status ]]; then
        echo $config_status  >> $LOGFILE
        STATUS="ProvisioningError"
        ztp_log "### status::: ${STATUS} ###"
        ztp_log "!!! Error encountered applying configuration file, please review the
log !!!!";
        MESSAGE="Error encountered applying configuration file, ZTP process failed"
    else
       STATUS="Provisioned"
       ztp_log "### status::: ${STATUS} ###"
       ztp_log "### Applying system configuration complete ###";
       MESSAGE="Applying system configuration complete, ZTP process completed"
   fi
```

```
}

#
# Call Crosswork ZTP API to update device ZTP status, IP address, hostname.
# Without this function, device status will remain in "In Progress" and not
# be updated in Crosswork.
#
# Using this API, device SSH/SNMP connectivity details can also be updated.
# Values for connectivity details values can be added as part of
# "connectivityDetails" array in below curl command. Sample snippet provided:
#
#   "connectivityDetails": [{
#     "protocol": "SSH",
#     "inetAddr": [{
#       "inetAddressFamily": "IPV4/IPV6",
#       "ipaddrs": "<ssh/snmp ipaddress>",
#       "mask": <ipaddress mask(Integer).>,
#       "type": "CONNECTIVITYINFO"
#     }],
#     "port": <ssh/snmp port(Integer)>,
#     "timeout": <ssh/snmp timeout(Integer). default to 60sec>
#   }]
#
function update_device_status() {

     echo "'"$IPADDR"'"
     echo "'"$MASK"'"
     echo "'"$DEVNAME"'"
     echo "'"$STATUS"'"
     echo "'"$HOSTNAME"'"
     echo "'"$MESSAGE"'"


    curl -d '{
       "ipAddress":{
            "inetAddressFamily": "IPV4",
            "ipaddrs": "'"$IPADDR"'",
            "mask":  '$MASK'
        },
       "serialNumber":"'"$DEVNAME"'",
       "status":"'"$STATUS"'",
       "hostName":"'"$HOSTNAME"'",
       "message":"'"$MESSAGE"'"
   }' -H "Content-Type: application/json" -X PATCH
http://${CW_HOST_IP}:${CW_PORT}/crosswork/ztp/v1/deviceinfo/status
}


# ==== Script entry point ====
STATUS="InProgress"
get_serialkey;
#get_serialkey_asr9901; // For Cisco ASR9901, replace get_serialkey with
get_serialkey_asr9901.
ztp_log "Hello from ${DEVNAME} !!!";
get_ipaddress;
ztp_log "Starting autoprovision process...";
download_config;
apply_config;
get_hostname;
update_device_status;

ztp_log "Autoprovision complete...";
exit 0
```

図 *6 :* セキュア *ZTP* : シンプルな *Day-Zero* 構成スクリプト

```
!! IOS XR
!
hostname ztpdevice1
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address dhcp
!
```

図 *7 :* セキュア *ZTP* : 置き換え可能なパラメータを使用した *Day-Zero* 構成スクリプト

```
!! IOS XR
!
hostname {$hname}
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address {$mgmt_ipaddr} {$mgmt_subnet_mask}
!
```

図 *8 :* セキュア *ZTP* : 構成後スクリプト

```
#!/bin/bash

################################################################################
#
#SZTP post script to update hostname and ipaddress for the device
# input -  serial key and crosswork host and port
#
################################################################################


export LOGFILE=/disk0:/ztp/customer/user-script.log

XRZTP_INTERFACE_NAME="MgmtEth0/RP0/CPU0/0"
# ZTP helper library is assumed to be installed in IOS-XR linux shell
source /pkg/bin/ztp_helper.sh
interfacedata=$(xrcmd "show interface ${XRZTP_INTERFACE_NAME}")

CW_HOST_IP="<EnterIPv4AddressHere>"   #update from the post script prepare code
CW_PORT="30603"           #update from the post script prepare code


# Send logging information to log file on device disk0:/ztp/user-script.log
function ztp_log() {

    echo "$(date +"%b %d %H:%M:%S") "$1 >> $LOGFILE
}


#
# Get IP address and subnet mask from device. IP address is assigned from DHCP
# server on interface where ZTP was triggered.
#
function get_ipaddress(){

    local ipvar=($(echo $interfacedata | awk -F "Internet address is " '{sub(/
.*/,"",$2);print $2}'));
    local ipv4addr=$(xrcmd "sh run interface ${XRZTP_INTERFACE_NAME} | i ipv4 address"
| awk '{print $3}')
    local ipv6addr=$(xrcmd "sh run interface ${XRZTP_INTERFACE_NAME} | i ipv6 address"
| awk '{print $3}')
    local ipaddress=($(echo $ipvar | awk -F "/" '{sub(/ .*/,"",$1);print $1}'));
    local mask=($(echo $ipvar | awk -F "/" '{sub(/ .*/,"",$2);print $2}'));
    local maskv6=($(echo $ipv6addr | awk -F "/" '{sub(/ .*/,"",$2);print $2}'));
```

```
    ztp_log "### Value of interfacedata => $interfacedata ###"
    ztp_log "### Value of ipvar => $ipvar ###"
    ztp_log "#####IPv4 address $ipaddress and mask $mask found. #####";

    IPADDR=$ipaddress
    MASK=$mask
    MASKV6=$maskv6

    return 0
}

#
# Fetch hostname from device configuration.
#
function get_hostname(){

    hostnamedata=$(xrcmd "show running-config hostname")
    local hostname=($(echo $hostnamedata | awk -F "hostname " '{sub(/ .*/,"",$2);print
$2}'));

    ztp_log "#####hostname $hostname found.";
    HOSTNAME=$hostname;
    return 0;
}



#
# Call Crosswork ZTP API to update device ZTP status, IP address, hostname.
# Without this function, device status will remain in "In Progress" and not
# be updated in Crosswork.
#
# Using this API, device SSH/SNMP connectivity details can also be updated.
# Values for connectivity details values can be added as part of
# "connectivityDetails" array in below curl command. Sample snippet provided:
#
#   "connectivityDetails": [{
#     "protocol": "SSH",
#     "inetAddr": [{
#       "inetAddressFamily": "IPV4/IPV6",
#       "ipaddrs": "<ssh/snmp ipaddress>",
#       "mask": <ipaddress mask(Integer).>,
#       "type": "CONNECTIVITYINFO"
#     }],
#     "port": <ssh/snmp port(Integer)>,
#     "timeout": <ssh/snmp timeout(Integer). default to 60sec>
#   }]
#
function update_device_status() {

     echo "'"$IPADDR"'"
     echo "'"$MASK"'"
     echo "'"$SERIAL_KEY"'"
     echo "'"$HOSTNAME"'"


    curl -d '{
       "ipAddress":{
            "inetAddressFamily": "IPV4",
            "ipaddrs": "'"$IPADDR"'",
            "mask":  '$MASK'
        },
       "serialNumber":"'"$SERIAL_KEY"'",
```

```
       "hostName":"'"$HOSTNAME"'",
       "message":"Post config script updated succssfully"
   }' -H "Content-Type: application/json" -X PATCH
http://${CW_HOST_IP}:${CW_PORT}/crosswork/ztp/v1/deviceinfo/status
}

function get_sudi_serial() {
   local rp_card_num=`ip netns exec xrnns /pkg/bin/show_platform_sysdb | grep Active  |
 cut -d ' ' -f 1`
   echo $rp_card_num
   xrcmd "show platform security tam all location $rp_card_num" > tamfile.txt
   local sudi_serial=$(sed -n -e '/Device Serial Number/ s/.*\- *//p' tamfile.txt)
   echo $sudi_serial
   SERIAL_KEY=$sudi_serial
   return 0
}

function ztp_disable()
{
 xrcmd "ztp disable noprompt"
}

function ztp_enable()
{
 xrcmd "ztp enable noprompt"
}

# ==== Script entry point ====
get_sudi_serial;
ztp_log "Hello from ${SERIAL_KEY} !!!";
get_ipaddress;
get_hostname;
update_device_status;

ztp_log "Autoprovision complete...";
ztp_log "Disabling secure mod"
ztp_disable;
exit 0
```

構成ファイルの読み込み

構成ファイルを Cisco Crosswork に読み込むには、次の手順を実行します。

1. Cisco Crosswork を起動します。
2. メインメニューから、**[デバイス管理（Device Management）]** > **[ZTP構成ファイル（ZTP Configuration Files）]** を選択ます。
3. をクリックします。 ＋
4. [参照（Browse）] をクリックして設定ファイルを選択します。
5. 必要な構成情報を入力します。

   セキュア ZTP を実装する場合は、[タイプ（Type）] ドロップダウンを使用して、追加する構成ファイルが [事前設定（Pre-config）] か、[Day 0設定（Day0 config）] か、または [設定後（Post-config）] かを指定します。

   クラシック ZTP と PnP ZTP の場合は、常に [Day 0設定（Day0-config）] [タイプ（Type）] ドロップダウンを選択します。

6. [追加（Add）] をクリックして、構成ファイルの追加を終了します。
7. 計画された ZTP 実行で使用されるすべての構成ファイルをロードするまで、必要に応じて繰り返します。

SMU の検索と読み込み

ソフトウェア メンテナンス アップデート（SMU）は、シスコ ネットワーク オペレーティング システム ソフトウェア イメージの特定のリリースにおける重大な問題の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 ファイルです。シスコは、SMU に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。

ZTP オンボーディング中にデバイスに SMU を適用することは、クラシック ZTP およびセキュア ZTP でのみサポートされ、その後は構成ファイルの適用中にのみサポートされます（「構成ファイルと読み込み （18 ページ）」を参照）。SMU は、Cisco IOS-XE デバイスまたは PnP ZTP ではサポートされていません。

ソフトウェアイメージと同様に、[Cisco Support & Downloads（シスコサポート & ダウンロード）] ページから SMU ファイルをダウンロードします。ダウンロード中に、SMU ファイルの MD5 チェックサムを記録します。Cisco Crosswork は MD5 チェックサムを使用して SMU ファイルの整合性を検証します。一度に 1 つずつ SMU を Cisco Crosswork に読み込ませ、読み込み中に各 SMU ファイルの MD5 チェックサムを入力します。

SMU を Cisco Crosswork に読み込ませるには、次の手順を実行します：

1. Cisco Crosswork を起動します。
2. 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。
3. をクリックします。
4. 入力するか、または [参照（Browse）]をクリックし、アップグレードするコンポーネントの SMU ファイルを選択しますプロンプトが表示されたら、ファイルの MD5 チェックサムを入力します。
5. [追加（Add）] をクリックして、SMU の追加を終了します。
6. 計画された ZTP 実行で使用されるすべての SMU ファイルを読み込むまで、必要に応じて繰り返します。

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 つずつ追加することもできます。これを行うには、**[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。

クレデンシャルプロファイルを使用すると、デバイスがサポートするプロトコルごとに異なるクレデンシャルを指定できます。SNMP クレデンシャルを含んでいるデバイス クレデンシャル プロファイルを作成する場合は、プロファイルにはデバイスで実際に有効になっている SNMP のバージョンのクレデンシャルと、そのバージョンのみを含めることをお勧めします。たとえば、デバイス設定で SNMPv3 が有効になっていない場合は、そのデバイスのクレデンシャルプロファイルに 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。これらの列に入力する順序によって、入力した 3 つのパスワードタイプ、3 つ のユーザー名、および 3 つのパスワードの間に次のマッピングが設定されます。
  - 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。

デバイスのシリアル番号の検索と読み込み

デバイスのシリアル番号は、すべての ZTP モードで必要です。

ほとんどの組織は、非販売在庫レコードの一部としてネットワークデバイスのシリアル番号のデータベースを維持しています。ネットワークに新しいデバイスを追加する場合、通常、購入時に新しいデバイスのシリアル番号を同じデータベースに追加します。これは、ZTP を使用してオンボードする予定のデバイスのシリアル番号を探す最初の場所です。

新しく購入したデバイスのシリアル番号を取得するには、シスコサポートに連絡することもできます。

最後の手段として、すでにイメージが作成されている Cisco IOS デバイスの場合は、デバイスコンソールにログインして、`show inventory CLI` コマンドを実行します。コマンド出力で、次の図に示すようなデバイス名と説明のセクションを探します。（この例に示すように）ラインカードまたはその他のオプションを備えたデバイスの場合、シャーシとカードの両方のシリアル番号を読み込む必要があります。

```
RP/0/RP0/CPU0:ios#sh inv
Wed May 18 13:33:53.674 UTC
NAME: "0/RP0", DESCR: "NC5501 w/o TCAM Route Processor Card"
PID: NCS-5501          , VID: V01, SN: FOC23297HGS

NAME: "Rack 0", DESCR: "NCS5501 w/o TCAM 1RU Chassis"
PID: NCS-5501          , VID: V01, SN: FOC2332R014
...
```

デバイスのシリアル番号を Cisco Crosswork に読み込ませるには、次の手順を実行します。

1. Cisco Crosswork を起動します。
2. 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）]** を選択します。
3. [シリアル番号を追加（Use Serial Number）] をクリックします。

4. [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` テンプレートファイルをダウンロードします。
5. 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。
6. [シリアル番号の追加（Add Serial Number）] を再度選択します。
7. [参照（Browse）] をクリックして、更新された CSV ファイルを選択します。
8. [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。

PDC、所有者証明書、および所有者キーを更新する

ピン留めされたドメイン証明書、所有者証明書、および所有者キーは、セキュア ZTP にのみ必要です。これらは、Classic ZTP および PnP ZTP では使用されません。

テスト環境では、ZTP が最初にインストールされたときに Cisco Crosswork が生成するデフォルトのピン留めドメイン証明書（PDC）、所有者証明書（OC）、および所有者キーを使用できます。これらの資格情報は、認証局（CA）としてシスコに依存しており、製品テストの便宜のためにのみ提供されています。シスコは、これらのデフォルトのログイン情報を使用している場合、ネットワークをセキュリティリスクにさらさない保護された「サンドボックス」環境で Cisco Crosswork をテストしていると想定しています。

本番環境で使用する場合は、ドメイン証明書をピン留めし、中間 OC を生成し、所有者キーに署名する必要があります。その後、次のセクション「デフォルトの PDC、OC、および所有者キーを更新する」の手順を使用して、これらのログイン情報のデフォルトバージョンを更新できます。

独自の証明書管理スタッフと手順を持つ組織は、選択した CA を使用して PDC、OC、および所有者キーを生成する方法に精通しています。これらのタスクでさらに支援が必要な組織は、このトピックの後半のセクション「ドメイン証明書のピン留め、所有者証明書の生成、所有者キーの署名」の例とアドバイスを参照してください。

デフォルトの PDC、OC、および所有者キーを更新する

デフォルトのピン留めドメイン証明書（PDC）、所有者証明書（OC）、および所有者キーを更新するには、次の手順を実行します。

1. Crosswork を起動します。
2. メインメニューから、**[管理（Administration）]** > **[証明書管理（Certificate Management）]** を選択します。
3. [証明書（Certificates）] で、[Crosswork ZTP 所有者（Crosswork-ZTP-Owner）] の横にある ⋯ をクリックし、[証明書の更新（Update Certificate）] をクリックします。
4. [参照（Browse）] をクリックして、固定ドメイン証明書（PEM ファイルまたは CRT ファイル）を選択します。ファイルを選択した状態で、[保存（Save）] をクリックします。

5. [参照（Browse）] をクリックして、所有者証明書（PEM ファイルまたは CRT ファイル）を選択します。ファイルを選択した状態で、[保存（Save）] をクリックします。
6. [参照（Browse）] をクリックして、所有者キー（PEM ファイル、KEY ファイル、CRT ファイル）を選択します。ファイルを選択した状態で、[保存（Save）] をクリックします。
7. [保存（Save）] をクリックして、デフォルトの証明書とキーを更新します。

ドメイン証明書をピン留めし、所有者証明書を生成し、所有者キーに署名します

次の手順では、OpenSSL と Linux Bash シェルを使用して、独自の認証局を使用して PDC、OC、および署名された所有者キーを生成する方法を示す一連の例を示します。このプロセスの追加の説明と例は、次の公開リソースで見つけることができます：OpenSSL Certificate Authority（認証局）。これらのログイン情報を生成したら、前のセクション「既定の PDC、OC、および所有者キーを更新する」の手順に従います。

1. 使用または生成する証明書とその他のファイルを管理するための一連のディレクトリを作成します。次に例を示します。

```
#!/bin/sh
mkdir ./ca
mkdir ./ca/certs
mkdir ./ca/crl
mkdir ./ca/newcerts
mkdir ./ca/private
chmod 700 ./ca/private
touch ./ca/index.txt
echo 1000 > ./ca/serial
mkdir ./ca/intermediate
mkdir ./ca/intermediate/certs
mkdir ./ca/intermediate/crl
mkdir ./ca/intermediate/csr
mkdir ./ca/intermediate/newcerts
mkdir ./ca/intermediate/private
chmod 700 ./ca/intermediate/private
touch ./ca/intermediate/index.txt
echo 1000 > ./ca/intermediate/serial
echo 1000 > ./ca/intermediate/crlnumber
```

2. ルートキーを生成します。次に例を示します。

```
#!/bin/bash
cd ca
openssl genrsa -aes256 -out private/ca.key.pem 4096
chmod 400 ./private/ca.key.pem
```

3. ルート証明書を生成します。次に例を示します。

```
#!/bin/bash
cd ca
##-subj "/C=us/ST=nc/L=rtp/O=cisco/OU=cx/CN=cisco.com" \
openssl req -config openssl.cnf -key ./private/ca.key.pem -new -x509 -days 7300
-sha256 -subj "/C=us/ST=nc/L=rtp/O=cisco/OU=cx/CN=cisco.com" -extensions v3_ca -out
 certs/ca.cert.pem
chmod 444 ./certs/ca.cert.pem
```

4. ルート証明書を確認します。次に例を示します。

```
#!/bin/bash
cd ca
openssl x509 -noout -text -in certs/ca.cert.pem
```

5. 中間キーを生成します。次に例を示します。

```
#!/bin/bash
cd ca
openssl genrsa -aes256 -out intermediate/private/intermediate.key.pem 4096
chmod 400 ./intermediate/private/intermediate.key.pem
```

6. 中間証明書を生成します。次に例を示します。

```
#!/bin/bash
cd ca
##-subj "/C=us/ST=nc/L=rtp/O=cisco/OU=cx/CN=cisco.com" \
openssl req -config intermediate/openssl.cnf -new -sha256 \
      -key intermediate/private/intermediate.key.pem \
      -out intermediate/csr/intermediate.csr.pem \
      -subj "/C=us/ST=nc/L=rtp/O=cisco/OU=cx/CN=intermediate.cisco.com"
chmod 444 ./certs/ca.cert.pem
© 2022 GitHub, Inc.
```

7. 中間鍵に署名します。次に例を示します。

```
#!/bin/bash
cd ca
openssl ca -config openssl.cnf -extensions v3_intermediate_ca \
      -days 3650 -notext -md sha256 \
      -in intermediate/csr/intermediate.csr.pem \
      -out intermediate/certs/intermediate.cert.pem
chmod 444 ./intermediate/certs/intermediate.cert.pem
```

8. 中間証明書を確認します。次に例を示します。

```
#!/bin/bash
cd ca
openssl x509 -noout -text -in intermediate/certs/intermediate.cert.pem
```

9. 証明書チェーンを作成します。次に例を示します。

```
#!/bin/bash
cd ca
cat intermediate/certs/intermediate.cert.pem \
      certs/ca.cert.pem > intermediate/certs/ca-chain.cert.pem
chmod 444 intermediate/certs/ca-chain.cert.pem
```

10. 証明書失効リスト（CRL）に署名します。次に例を示します。

```
#!/bin/bash
mycsr=$1
myip=$2
export SAN="IP:${myip}"
echo $SAN
cd ca
openssl ca -config intermediate/openssl.cnf \
      -extensions usrSrv_cert -days 750 -notext -md sha256 \
      -in intermediate/csr/${mycsr}.csr.pem \
      -out intermediate/certs/${mycsr}.cert.pem
chmod 444 intermediate/certs/${mycsr}.cert.pem
```

所有権バウチャーのリクエストと読み込み

所有権証明書（OV）は、セキュア ZTP にのみ必要です。提供方法に応じて、一度に 1 つずつ読み込むことも、まとめて読み込むこともできます。

シスコは、要求に応じて OV を VCJ または TAR ファイルの形式で提供します。

Secure ZTP を使用してサードパーティのデバイスをオンボードする場合は、サードパーティの製造元に VCJ ファイルを要求する必要があります。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。サードパーティメーカーのバウチャーに関する制限の背景については、「#unique_173 unique_173_Connect_42_SecureZTPGuidelinesThird （5 ページ）」を参照してください。

シスコからの所有権バウチャーのリクエスト

Secure ZTP を使用してオンボードする予定のシスコデバイスの OV を要求するには、[シスコサポートにお問い合わせください（Contact Cisco Support）]。OV を要求するときは、以下を提供する必要があります。

- [ピン留めされたドメイン証明書（Pinned Domain Certificate）]：認証局（CA）によって発行され、ユーザーがピン留めした信頼できるデジタル証明書。PDC のピン留めの詳細については、「PDC、所有者証明書、および所有者キーを更新する （33 ページ）」を参照してください。
- Secure 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（「デバイスのシリアル番号の検索と読み込み （32 ページ）」を参照）。

単一のデバイスに対するリクエストの例を次に示します。

```
{
    "expires-on": "2016-10-21T19:31:42Z",
    "assertion": "verified",
    "serial-number": "JADA123456789",
    "idevid-issuer": "base64encodedvalue==",
    "pinned-domain-cert": "base64endvalue==",
    "last-renewal-date": "2017-10-07T19:31:42Z"
 }
```

シスコサポートは、VCJ ファイルを送信することにより、OV 要求に応答します。複数のデバイスの OV をリクエストした場合、単一の VCJ ファイルではなく、TAR ファイルで複数の VCJ を受け取ります。シスコサポートと合意した安全な方法を使用して、VCJ または TAR ファイル交換を実行することをお勧めします。

個々の VCJ ファイルには、送信元が何であれ、ファイル名としてデバイスのシリアル番号が必要であることに注意してください。ステップ 1 で指定された要求例に従って、シスコは次の名前のファイルを返します：JADA123456789.VCJ。

所有権バウチャーの読み込み

所有権バウチャーの読み込むには、次の手順を実行します。

1. Cisco Crosswork を起動します。
2. 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）]** を選択します。
3. [バウチャーの追加（Add Voucher）] をクリックします。
4. アップロードする VCJ または TAR ファイルの名前を入力するか、参照します。
5. [アップロード（Upload）] をクリックして、OV のアップロードを完了します。

デフォルトの所有権バウチャー証明書の更新

デフォルトの所有権バウチャー証明書を更新するには、次を実行します。

1. メインメニューから、**[管理（Administration）]**>**[証明書管理（Certificate Management）]** を選択します。
2. [証明書の更新（Update Certificate）] をクリックします。
3. [参照（Browse）] をクリックして、デフォルトの所有権バウチャーの更新に使用する TAR ファイルまたは VCJ ファイルを選択します。
4. [証明書の更新（Update Certificate）] をクリックします。
5. [保存（Save）] をクリックします。

SUDI ルート証明書の準備と読み込み

SUDI ルート証明書は、IOS-XE デバイスのオンボーディング時にセキュア e ZTP および PnP ZTP に必要です。Classic ZTP には使用されません。

「SUDI 証明書」には次の 2 種類があります。

- デバイスの **SUDI 証明書**（トラストアンカー証明書とも呼ばれます）。すべての Cisco IOS-XR および IOS-XE デバイスには、SUDI 証明書がデバイスに保存されています。デバイスの SUDI 証明書は変更できません。
- **SUDI ルート証明書**。これは、各デバイスで SUDI 証明書を有効にするルート認証局です。

SUDI ルート証明書を Crosswork にアップロードすると、セキュア ZTP プロセス（および IOS-XE デバイスの場合は PnP ZTP プロセス）が、SUDI ルート証明書をデバイスに保存されている SUDI 証明書と比較することによって、各デバイスを認証できるようになります。これは、PnP ZTP または Secure ZTP プロセスがデバイスにブートストラップ情報を提供する前に必要です。

SUDI ルート証明書を準備して Cisco Crosswork にアップロードするには、次の手順を実行します。

1. 「Cisco Root CA 2048」および「Cisco Root CA 2099」ファイルを PEM 形式で、Cisco PKI：ポリシー、証明書、およびドキュメント（https://www.cisco.com/security/pki/policies/index.html）からダウンロードします。

2. 次の例のように、ASCIIテキストエディタを使用して、ダウンロードした2つのPEMファイルを1つのPEMファイルに結合します。

```
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KCtU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
....
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDITCCAgmgAwIBAgIJAZozWHjOFsHBMA0GCSqGSIb3DQEBCwUAMC0xDjAMBgNV
...
PKkmBlNQ9hQcNM3CSzVvEAK0CCEo/NJ/xzZ6WX1/f8Df1eXbFg==
-----END CERTIFICATE-----
```

3. Cisco Crosswork を起動します。
4. メインメニューから、**[管理（Administration）]**>**[証明書管理（Certificate Administration）]** を選択します。
5. ＋をクリックして、次のようにフィールドに入力します。

   [証明書名（Certificate Name）]：Crosswork-ZTP-Device-SUDI

   [証明書の役割（Certificate Role）]：ZTP SUDI

   [Cisco M2 CA証明書（Cisco M2 CA Certificate）]：アップロードする PEM ファイルの名前を入力するか、参照します。
6. [保存（Save）] をクリックします。Crosswork は SUDI ルート証明書を保存します。

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 では、事前設定ファイル、設定後ファイル、および 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 を実装する場合は、[セキュア ZTP（Secure ZTP）] のスライダを [有効（Enabled）] に移動します。次に、事前設定ファイルと設定後ファイルの名前を入力します。

OS バージョンとして IOS-XE を選択した場合、セキュア ZTP は使用できません。

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。

複数の ZTP デバイスエントリを作成する最も簡単な方法は、デバイスエントリの CSV ファイルを使用して、それらをまとめてインポートすることです。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。その後で、必要な結果を取得する方法を確認できます。

次のトピックでは、デバイスエントリの CSV ファイルをダウンロードして使用し、適切な形式の ZTP デバイスエントリを一括で作成する方法について説明します。

また、単一 ZTP デバイスエントリの作成 （46 ページ）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 CSV 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 *4: ZTP* デバイス エントリ テンプレートの列リファレンス

テンプレートの列	使用方法
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。
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。

テンプレートの列	使用方法
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。
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。入力する名前は、Cisco Crossworkで指定されているクレデンシャルプロファイルの名前と一致する必要があります。
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。たとえば、IOS XR などです。Cisco IOS プラットフォーム名は、ハイフンではなくスペースを使用して入力する必要があることに注意してください。
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。 [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。
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。Cisco Crosswork は、アップロード時にすべての設定ファイルに一意の ID を割り当てます。 [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。
[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。 ZTP プロファイルを使用して設定 ID、イメージ ID、OS プラットフォームなどを指定する場合にのみ必要です。
[製品 ID（Product ID）] *	デバイスハードウェアにコード化された、シスコによって割り当てられた PID（製品 ID）を入力します。PID は、工場出荷時にすべてのシスコ ネットワーキング デバイスに貼付されているラベルに印刷された UDI（一意のデバイス識別子）情報から取得できます。 このリリースでは、PID の検証は行われないことに注意してください。将来の要件に備えて、正しい PID を指定することをお勧めします。

テンプレートの列	使用方法
UUID	オンボーディング時にデバイスに割り当てる汎用一意識別子（UUID）を生成して指定することができます。このオプションを選択した場合は、この列に 128 ビット UUID を入力します。それ以外の場合は、このフィールドを空白のままにしておくと、Cisco Crosswork はデバイスのオンボーディング時にランダムな UUID を割り当てます。
[MACアドレス（MAC Address）]	デバイスの MAC アドレスを入力します。
[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。
[設定属性（Configuration Attributes）]	デバイスの設定ファイルのカスタム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。デフォルトの置換可能パラメータのみを使用する場合は、このフィールドを空白のままにします。セキュア ZTP を使用している場合、カスタムの置換可能なパラメータは、Day 0 設定ファイルのパラメータにのみ使用できます。これらのパラメータの使用方法については、次を参照してください。
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。プロトコルの正しい組み合わせを選択するには、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 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。

テンプレートの列	使用方法
[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。 • オンボーディングしたデバイスのステータスを管理対象外またはダウンとして設定します。 • 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。プロトコルの正しい組み合わせを選択するには、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、Cisco Crosswork で指定されているデバイス管理プロバイダの名前と正確に一致する必要があります。
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。
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。
[セキュア ZTP が暗号化済み（Secure ZTP Encrypted）]	現在サポートされていません。FALSE と入力します。
[イメージ ID（Image ID）]	Cisco Crosswork は、アップロード時にすべてのソフトウェア イメージ ファイルに一意の ID を割り当てます。 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。 オンボーディング時にソフトウェアイメージのインストールを含める必要があり、[プロファイル名（Profile Name）] 列にこのソフトウェアイメージを含む ZTP プロファイルを指定しなかった場合にのみ必要です。

テンプレートの列	使用方法
[事前設定 ID（PreConfig ID）]	Cisco Crossworkは、アップロード時にすべての設定ファイルに一意のIDを割り当てます。 [設定 ID（Config ID）] 列に指定した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。 オンボーディング時に事前設定ファイルを実行する場合にのみ必要です。
[設定後 ID（PostConfig ID）]	Cisco Crossworkは、アップロード時にすべての設定ファイルに一意のIDを割り当てます。 [設定 ID（Config ID）] 列に指定した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。 オンボーディング時に設定後ファイルを実行する場合にのみ必要です。
[SZTP 設定モード（SZTP Config Mode）]	セキュア ZTP で、[設定 ID（Config ID）] 列、[事前設定 ID（PreConfig ID）] 列、および [設定後 ID（PostConfig ID）] 列で指定した設定ファイルをデバイス上の既存の設定とマージする場合は、**merge** と入力します。指定した設定ファイルの内容で既存の設定を上書きする場合は、この列を空白のままにします（この列を空白のままにすることで、上書きがデフォルトになります）。
バージョン ID（Version ID）	設定のバージョン ID。 オンボーディング時に実行する事前設定ファイルと設定後ファイルを指定した場合にのみ必要です。
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。これ以外の場合は、このフィールドは空白のままにしておきます。
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。これ以外の場合は、このフィールドは空白のままにしておきます。
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。これ以外の場合は、このフィールドは空白のままにしておきます。

Crosswork 接続プロトコルの要件

Cisco Crosswork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。この表に示されているアプリケーションを使用する場合は、デバイスでこれらのプロトコルを有効にしてください。オンボーディングするには、各デバイスでこれらのプロトコルの少なく

とも 1 つを有効にする必要があります。これらのプロトコルが 1 つもなければ、デバイスをオンボーディングできません。

表 *5:* 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	Crosswork アプリケーション	アプリケーション機能
GRPC	9090	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	Cisco Crosswork API 通信
HTTP	80	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	Cisco Network Services Orchestrator へのデバイスのオンボーディング
HTTPS	443	• Cisco Crosswork Network Controller	Cisco Network Services Orchestrator へのデバイスのオンボーディング
NETCONF	830	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	Cisco Network Services Orchestrator へのデバイスのオンボーディング
SNMPv2	161	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	SNMPv2 でのデータ収集

プロトコル	ポート	Crosswork アプリケーション	アプリケーション機能
SNMPv3	161	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	SNMPv3 でのデータ収集
SSH	22	• Cisco Crosswork Network Controller • Cisco Crosswork Change Automation and Health Insights • Cisco Crosswork Optimization Engine	• CLI でのデータ収集 • 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。

各デバイスエントリに必要な情報については、「ZTP デバイスエントリファイルの作成 （39 ページ）」のテンプレートリファレンスを参照してください。

ZTP でデバイスをオンボーディングすると、Cisco Crosswork はデバイスの地理的位置など、デバイスに関する詳細情報を要求するフィールドを表示します。「オンボーディング済み ZTP デバイス情報の入力 （71 ページ）」の説明に従って、デバイスのインベントリレコードを編集して、この追加情報を提供する必要があります。

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）]** > **[ネットワークデバイス（Network 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 [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、Cisco Crosswork がデバイスと通信してダウンロード要求に応答できるように、DHCP（および PnP ZTP の場合は TFTP）サーバー設定を更新する必要があります。

次のトピックでは、この要件を満たすようにサーバー設定を更新する例を示します。次の手順と例は、使用する ZTP モードによって異なります。

- クラシック ZTP については、「クラシック ZTP での DHCP の設定 （48 ページ）」を参照してください。
- セキュア ZTP については、「セキュア ZTP での DHCP の設定 （52 ページ）」を参照してください。
- PnP ZTP については、「PnP ZTP での DHCP と TFTP の設定 （54 ページ）」を参照してください。
- クラシック ZTP と Cisco PNR の設定スクリプトのセットについては、「Cisco Prime Network Registrar（CPNR）でのクラシック ZTP DHCP の設定スクリプト （54 ページ）」を参照してください。

クラシック 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。

図 *9:* クラシック *ZTP DHCP* コンテキスト

```
#
authoritative;

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

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;
}
```

例：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 クラスタの 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 *10 : 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 *11 : 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 *12 : 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 *13 : 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";
  }
}
```

例：クラシック ZTP での 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 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 *14 :* 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 *15 :* 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/vl/file";
    }
}
```

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。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 *6 : ISC IPv4 DHCP* 設定のホストエントリと値（クラシック *ZTP*）

IPv4 エントリ	説明
`host NCS5k-1`	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。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。

IPv4 エントリ	説明
`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。
`option user-class = "exr-config"` and `ffl filename =`	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。

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。

セキュア ZTP での DHCP の設定

セキュア 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 構成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。

次に、この要件を満たすようにDHCPサーバー構成ファイルを更新する方法を示す例を示します。この例では、インターネット システム コンソーシアム（ICS）DHCP サーバーを使用していることを前提としています。セキュア ZTP には、`sztp-redirect` オプションを有効にする行が必要です。

デバイスはオプション 143 とともにユーザークラスオプション `xr-config` を送信するため、これはホストブロックの一部として示されているように設定する必要があることに注意してください。

図 *16*：セキュア *ZTP DHCP* 構成ファイル

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, it will be used as the
# configuration file instead of this file.
#

# option definitions common to all supported networks...
option domain-name "cisco.com";
option domain-name-servers 192.168.100.101, 171.70.168.183;
option sztp-redirect code 143 = text;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
INTERFACES="ens192";

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none'), since DHCP v2 does not
# have support for DDNS.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, uncomment the "authoritative" directive below.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.100.0 netmask 255.255.255.0 {
 option routers 192.168.100.100;
 range 192.168.100.105 192.168.100.150;
}

host sztpdevice {
 hardware ethernet 08:4f:a9:0e:43:c8;
 fixed-address 192.168.100.153;
  if exists user-class and option user-class ="xr-config" {
# If you want to use a remote circuit ID to identify a remote host
# comment out the first option line and uncomment the second.
     option sztp-redirect
"https://<CrossworkHostIP>:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data";

     #option sztp-redirect
"https://<CrossworkHostIP>:30617/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrap-data?remoteid=MAR1&circuitid=Gig001";
```

```
  }
}
```

PnP ZTP での DHCP と TFTP の設定

PnP ZTP 処理をトリガーする前に、次の手順を実行する必要があります。

1. ASR 900 デバイスと NCS 520 デバイスから到達可能な外部 TFTP サーバーを設定します。
2. PnP プロファイルを外部 TFTP サーバーにアップロードします。
3. Cisco Crosswork PnP サーバーの場所を示す情報で DHCP 設定ファイルを更新します。

この情報により、Cisco Crosswork が許可されます。

以降のトピックでは、これらの各タスクを実行する例を示します。

外部 TFTP サーバーの設定

サポートされているすべての Cisco ASR 900 シリーズと NCS 520 シリーズのルータには、外部 TFTP サーバが必要です。サーバーはポート 69 UDP でアクティブである必要があります。

TFTP への PnP プロファイルのアップロード

PnPプロファイルは、単純な汎用設定ファイルです。TFTPリポジトリの設定サービスへのPnPプロファイルのアップロードは、1 回限りのアクティビティです。

プロファイルの内容で、Crosswork クラスタの仮想データポートの使用を指定する必要があります。この例では、IP アドレス 192.168.100.211 は組み込み Cisco Crosswork PnP サーバーのデータ VIP であり、30620 は PnP サーバーの外部ポートです。

図 *17 :* 例：汎用 *PnP* プロファイル

```
pnp profile cwpnp-data
transport http ipv4 192.168.100.211 port 30620
```

DHCP サーバーの設定

DCHCP エントリは、デバイス上の PnP エージェントから外部 TFTP サーバーの IP アドレスにトラフィックをリダイレクトします。

図 *18 : PnP ZTP DHCP* の設定例

```
option tftp code 150 = text;
host cztp1 {
 hardware ethernet 00:a7:42:86:54:f1;
  option tftp150 "192.168.100.205":
   }
```

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、クラシック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。

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

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。

図 *19 : 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-b2e5-051249d9b0db"


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 *20 : 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-b2e5-051249d9b0cb"


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-b2e5-051249d9b0db"


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 *21 : 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 *22 : 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 *23 : 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 *24 : 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 *25 : 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 *26 : 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。

始める前に

いずれかのデバイスで ZTP ブートストラップをトリガーする前に、次の作業が完了していることを確認します。

- 「ZTP 設定のワークフロー （14 ページ）」で説明されているすべての予備設定タスク。
- ZTP デバイスエントリファイルの作成 （39 ページ） または 単一 ZTP デバイスエントリの作成 （46 ページ） の説明に従ったブートストラップするデバイスの ZTP デバイスエントリの作成。

- Crosswork ZTP での DHCP の設定 （48 ページ） の説明に従った ZTP モードとサーバーの選択に適した DHCP の設定

セキュア ZTP を使用している場合：

1. オンボードする各デバイスのコンソールに Telnet で接続します：`telnet <device IP><userID><password>`。
2. デバイスで Secure ZTP が有効になっているかどうかを確認します。
   1. IOS-XR バージョン 7.5.2 以前の場合：Bash 実行モードに入り、次のコマンドを発行します：`[xr-vm_node:~]$pyztp2 --ztp-mode ZTP モード：セキュア`
   2. 7.5.2 以降の IOS-XR バージョンの場合：IOS CLI コマンドプロンプトに移動し、次のコマンド `show ztp information` を入力します。
3. ログと構成を消去するには、次のコマンドを発行します。

```
ios#ztp clean

ios#config terminal

ios(config)#commit 置換

ios(config)#end
```

PnP ZTP を使用する場合は、ZTP 処理をトリガーする**前に**、各 IOS-XE デバイスの最小ライセンスブートレベルが **metroipaccess** または **advancedmetroipaccess** に設定されていることを確認します。ブートレベルが正しく設定されている場合、デバイスの IOS-XE `#sh run | sec license` CLI コマンドの出力に、2 つのライセンスレベル、`license boot level advancedmetroipaccess` または `license boot level metroipaccess` のいずれかを示すステートメントが含まれている必要があります。コマンド出力にこれらの 2 つより低い他のライセンスレベルが表示された場合、Cisco PnP 暗号化機能が有効になりません。これにより、証明書のインストールが失敗して PnP ZTP デバイスのプロビジョニングが失敗します。

ステップ 1 使用している ZTP モードに適した ZTP 処理を開始します。

- クラシック ZTP の場合は、次のいずれかのオプションを使用します。
  - 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 経由でデバイスコンソールに接続し、**ztp initiator** コマンドを発行します。
- セキュア ZTP の場合は、次のいずれかのオプションを使用します。
  - 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。
- 以前にイメージ化されたデバイスの場合：Telnet 経由でデバイスコンソールに接続し、次のコマンドを発行します（ここで指定された `ztp initiate interface` 値により、デバイス管理ポートでセキュア ZTP が開始されます）：

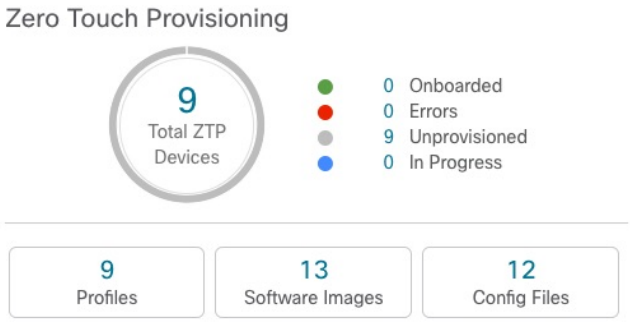
```
ztp enable noprompt

ztp initiate debug verbose interface MgmtEth 0/RP0/CPU0/0
```

- PnP ZTP の場合は、デバイスに適したオプションを使用します。
  - Cisco ASR 903、ASR 907、および NCS 520デバイスの場合は、Telnet 経由で接続し、**write erase** コマンドを発行してから、**reload** コマンドを実行します。
  - Cisco ASR 920 デバイスの場合は、シャーシの ZTP ボタンを 8 秒間押します。

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。1 回のセッションの間に、すべてのデバイスまたは必要な数のデバイスを再起動できます。

ステップ 2 次の図に示すように、[ゼロタッチプロビジョニング（Zero Touch Provisioning）] ステータスタイルを使用して、ZTP 処理の進行状況をモニターします。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

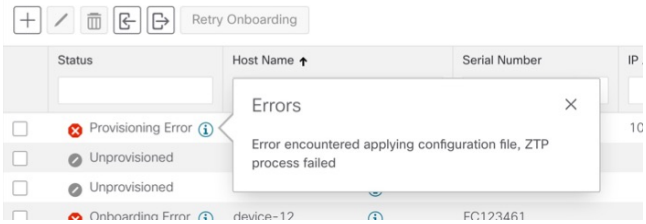
通常、Cisco Crosswork ZTP のプロビジョニングとオンボーディングは迅速かつ自動的に行われます。問題はときどき発生するため、次のトピックでは、一般的な問題と ZTP モードに固有の問題の両方を含む、問題を診断および修正する方法について説明します。

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。

ステータス列を使用して ZTP の問題を診断する

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



次の 2 つのセクションで説明するように、ZTP エラーログを使用して問題を診断することもできます。

エラーログを使用して ZTP の問題を診断する

Crosswork を実行している 1 つ以上の仮想マシンと、その VM で実行されている Crosswork ZTP サービス Kubernetes ポッドのインスタンスの 1 つに SSH ログインすることにより、ZTP エラーログ ファイルに直接アクセスできます。手順は次のとおりです。

1. 次のような Secure Shell コマンドを使用して VM にログインします。

```
ssh admin@VMIP
```

それぞれの説明は次のとおりです。

- `admin` は Crosswork 管理者 ID です。例：cw-admin。
- `VMIP` は、Crosswork を実行している仮想マシンの IP アドレスです。例：192.168.100.102。

2. 次のようなコマンドを使用して、cw-ztp-service Kubernetes ポッドにアクセスします。

```
# kubectl exec -it PodID# bash
```

`PodID#` は、cw-ztp-service Kubernetes ポッドの ID です。アクセスするポッドの番号と一致するように、必要に応じてポッド ID 番号を変更します（ポッド 0 が常に最初です）。例：`cw-ztp-service-0`、`cw-ztp-service-1`、`cw-ztp-service-2` 等

次のようなコマンドでログフォルダに移動します：`cd /var/log/robot/`。その後、フォルダ内の次の ZTP 固有のファイルのいずれかを開くことができます。

- `cw-image-service_stdout.log`
- `cw-image-service_stderr.log`
- `cw-config-service_stdout.log`
- `cw-config-service_stderr.log`

ZTP エラーログの要求

Crosswork ユーザーインターフェイスを使用して、ZTP エラーログファイルのコピーを要求できます。手順は次のとおりです。

1. 管理者権限を持つ ID を使用して、Crosswork ユーザーインターフェイスにログインします。

2.**[管理（Administration）] > [Crosswork Manager]**を選択します。

3. [Crossworkの概要（Crosswork Summary）] ページが表示されたら、[ゼロタッチプロビジョニング（Zero Touch Provisioning）] タイルをクリックします。Crosswork は、ZTP アプリケーションの詳細を表示します。

4. アプリケーションの詳細が表示されたら、**[Showtechのオプション（Showtech Options）] > [リクエストロゴ（Request Logs）]**の順に選択します。次に、**[Showtechリクエスト（Showtech Requests）]** を選択します。 リクエストが完了すると、ダッシュボードからログファイルを取得できます。

（注） 処理のオンボーディングフェーズで問題が発生した場合は、ZTP のログに加えて、Crossworkインベントリ マネージャ アプリケーション（dlminvmgr）のログを要求することができます。上記の手順 3 で、[ゼロタッチプロビジョニング（Zero Touch Provisioning）] の代わりに**[プラットフォームインフラストラクチャ（Platform Infrastructure）]** を選択することで、これを行うことができます。

共通の ZTP 問題のトラブルシューティング

以下は、ZTP モードのいずれかで発生する可能性のある一般的な問題の解決策を示しています。

表 *7:* 一般的な *ZTP* の問題と修正

フェーズ	問題	症状	Remedy
[設定（Setup）]	イメージ、構成、または SMU ファイルのアップロードが失敗する	アップロード中にユーザーインターフェイスに表示されるエラーメッセージ	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	ドロップダウンリストにないファイル	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 IOS リリース番号に基づいてファイルを選択します。ファイル情報が、作成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。
	デバイスエントリの CSV ファイルのインポート中にエラーが発生しました	異なります。エラーログを参照	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。
プロビジョニングされていない	DHCP が応答しないか、オファーの実行に失敗する	ZTP 処理の停止	ping および同様のツールを使用して、Cisco Crosswork サーバーから DHCP サーバーへのアクセスをテストします。

フェーズ	問題	症状	Remedy
進行中（In Progress）	イメージまたは SMU ファイルのダウンロードに失敗した	ZTP 処理の停止	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	ログに記録されたエラー	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。

フェーズ	問題	症状	Remedy
オンボード済み	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	ステータス列に**プロビジョニング済み**が表示されませんでした	[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	ステータス列に**オンボーディングエラー**が表示される	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。

クラシック ZTP の問題のトラブルシューティング

次の表は、Classic ZTP 処理で発生する可能性のある問題の解決策を示しています。

表 *8 :* クラシック *ZTP* の問題と修正

フェーズ	問題	症状	Remedy
プロビジョニングされていない	Crossworkはデバイスのシリアル番号を確認できません	ステータス列に「進行中」と表示されない	ZTP は、追加するデバイスの数に関係なく、複数のシリアル番号の追加をサポートします。デバイスエントリを作成するときは、正しいシリアル番号を割り当ててください。ZTP はシリアル番号に基づいて開始され、接続されたデバイスエントリはそれに基づいて状態の変化を表示し始めます。
進行中（In Progress）	ブートスクリプトの実行に失敗する	処理が停止します。エラーログを参照してください。	ブートスクリプトにエラーがないか調べて修正し、再試行してください。
	iPXEのリロードが失敗する	処理が停止します。エラーログを参照してください。	これは、デバイスの一時的な問題が原因である可能性があります。再度お試しください。プロセスが繰り返し失敗する場合は、シスコデバイスサポートチームに連絡してください。
プロビジョニングされていない、進行中	デバイス進捗レポートAPI 呼び出しが失敗する	処理が停止します。エラーログを参照してください。	API 呼び出しの形式が正しく、値が正しいことを確認してください。それらを修正して、再試行してください。ネットワークの問題が原因で一時的に接続が失われた結果である可能性もあります。

PnP ZTP の問題のトラブルシューティング

次の表は、PnP ZTP 処理で発生する可能性がある問題の解決策を示しています。処理の各段階でのアクティビティの詳細については、[ZTP処理のトピックへのリンク（Link to ZTP Processing topic）] を参照してください。

表 *9 : PnP ZTP* の問題と修正

フェーズ	問題	症状	Remedy
プロビジョニングされていない	PnP プロファイルのダウンロードが失敗する	デバイスがプロビジョニングされていない状態のままになる	パケットのドロップまたは同様のネットワークトラフィックの問題により、ダウンロードが失敗した可能性があります。まず、PnP プロファイルに正しいファイル名、プロトコル、IP アドレス、およびポートが指定されていることを確認します。TFTP サーバーが稼働していて到達可能であることを確認します。次に、デバイスから ZTP を再度トリガーしてみてください。
プロビジョニングされていない、進行中	機能サービスリクエストが失敗する	ZTP デバイスエントリは、「サービス機能チェックに失敗しました」というメッセージとともにエラー状態に移行します。理由：デバイスが最低限必要な機能をサポートしていません。	PnP ZTP が機能するには、プロビジョニングされる XE デバイスが次の最小機能をサポートしている必要があります。 • device-info • 証明書のインストール • image-install • config-upgrade • バックオフ この要件に問題がある場合は、シスコデバイスサポートチームにお問い合わせください。
進行中（In Progress）	証明書インストールに失敗しました	ZTP デバイスがエラー状態になり、「証明書のインストールサービスに失敗しました」というメッセージが表示されます。	まず、XEデバイスにログインし、トラストポイント「CrossworkPnP」がすでに存在する場合はクリーンアップします。次に、Crosswork GUI からデバイスを UnProvisioned 状態に戻し、ZTP を最初から再トリガーします。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。