Zero Touch Provisioning

This section contains the following topics:

Zero Touch Provisioning Concepts

The Cisco Crosswork Zero Touch Provisioning (ZTP) application allows you to provision networking devices remotely. You can ship factory-fresh devices to a branch office or remote site. Local operators can cable these devices to the network without installing an image or configuring them. To use ZTP, you first establish an entry for each device in the DHCP server and in the ZTP application. You can then activate ZTP processing by connecting the device to the network and powering it on or reloading it. ZTP downloads and applies a certified image and one or more configurations to the device automatically (you can also apply configurations only). Once configured, ZTP on boards the new device to the Cisco Crosswork device inventory. You can then use other Cisco Crosswork applications to monitor and manage the device.

Cisco Crosswork ZTP uses the following basic terms and concepts:

- **Classic ZTP**: A process to download and apply software and configuration files to devices. It uses iPXE firmware and HTTP to boot the device and perform downloads. It's not suitable for use over public networks.
- **Secure ZTP**: A secured process to download and apply software images and configuration files to devices. It uses secure transport protocols and certificates to verify devices and perform downloads.
- **Evaluation License Countdown**: Licenses for devices onboarded using ZTP have an evaluation period, normally 90 days. Cisco Crosswork displays a countdown banner throughout the evaluation period. Try to purchase a pool of licenses by the time the evaluation period expires. After expiration, Cisco Crosswork displays a warning banner and blocks onboarding of new devices until you apply purchased licenses.
- **Image file**: A binary software image file, used to install the network operating system on a device. For Cisco devices, these files are the supported versions of Cisco IOS-XR images. When configured to do so, the Classic ZTP process downloads the image from Cisco Crosswork and installs it using the open-source boot firmware iPXE. If you must install SMUs, ZTP applies them as part of configuration processing.
- **Configuration file**: A file used to set the operating parameters of the newly imaged or reimaged device. The file can be a Python script, Linux shell script, or a sequence of Cisco IOS CLI commands stored as

ASCII text. The ZTP process downloads the configuration file to the newly imaged device, which then executes it. ZTP processing requires configuration files.

- **Credential profile**: Collections of passwords and community strings that are used to access devices via SNMP, SSH, HTTP, and other network protocols. Cisco Crosswork uses credential profiles to access your devices, automating device access. All credential profiles store passwords and community strings in encrypted format.
- **Bootfile name**: The explicit path to and name of a software image that is stored in the ZTP repository. For each device you plan to onboard using ZTP, specify the bootfile name as part of the device configuration in DHCP.
- **HTTPS/TLS**: Hypertext Transport Protocol Secure (HTTPS) is a secure form of the HTTP protocol. It wraps an encrypted layer around HTTP. This layer is the Transport Layer Security (TLS) (formerly Secure Sockets Layer, or SSL).
- **iPXE**: The open-source boot firmware iPXE is the popular implementation of the Preboot eXecution Environment (PXE) client firmware and bootloader. iPXE allows devices without built-in PXE support to boot from the network. The iPXE boot process is part of Classic ZTP processing, and isn't part of Secure ZTP processing. However, on-site technicians can still force iPXE boot and then begin Secure ZTP processing.
- **Owner certificate**: The CA-signed end-entity certificate for your organization, which binds a public key to your organization. You install owner certificates on your devices.
- **Ownership Voucher**: Nonceless audit vouchers that verify that devices onboarded with ZTP are bootstrapping into a domain your organization owns. Cisco supplies OVs in response to requests from your organization.
- **PDC**: A Pinned Domain Certificate (PDC) is the CA- or self-signed domain certificate of your organization. The public key of the PDC pins the PDC to the DNS network domain assigned to your organization. The PDC helps your devices verify that images and configurations that are downloaded and applied during ZTP processing come from within your organization.
- **SUDI**: The Secure Unique Device Indentifier (SUDI) is a certificate with an associated key pair. The SUDI contains the product identifier and serial number. Cisco inserts the SUDI and key pair in the device hardware Trust Anchor module (TAm) during manufacturing, giving the device an immutable identity. During Secure ZTP processing, the back-end system challenges the device to validate its identity. The router responds using its SUDI-based identity. This exchange, and the TAm encryption services, permit the back-end system to provide encrypted image and configuration files. Only the specific router can open these encrypted files, ensuring confidentiality in transit over public networks.
- **SUDI Root CA Certificates**: A root authority certificate for SUDIs, issued and signed by a Certificate Authority (CA), used to authenticate subordinate SUDI certificates.
- **UUID**: The Universal Unique Identifier (UUID) uniquely identifies an image file that is uploaded to Cisco Crosswork. You can use the UUID of the software image file in the DHCP bootfile URL. UUIDs are not required for configuration files.
- **ZTP asset**: ZTP requires access to several types of files and information in order to onboard new devices. We refer to these files and information collectively as "ZTP assets." You load these assets as part of ZTP setup, before initiating ZTP processing.
- **ZTP profile**: A Cisco Crosswork storage construct that combines (normally) one image and one configuration into a single unit. Cisco Crosswork uses ZTP profiles to automate imaging and configuration processes. Using ZTP profiles is optional, but we recommended them. They are an easy way to organize

ZTP images and configurations around device families, classes, and roles, and help maintain consistent ZTP use.

- **ZTP repository**: The location where Cisco Crosswork stores ZTP image and configuration files.

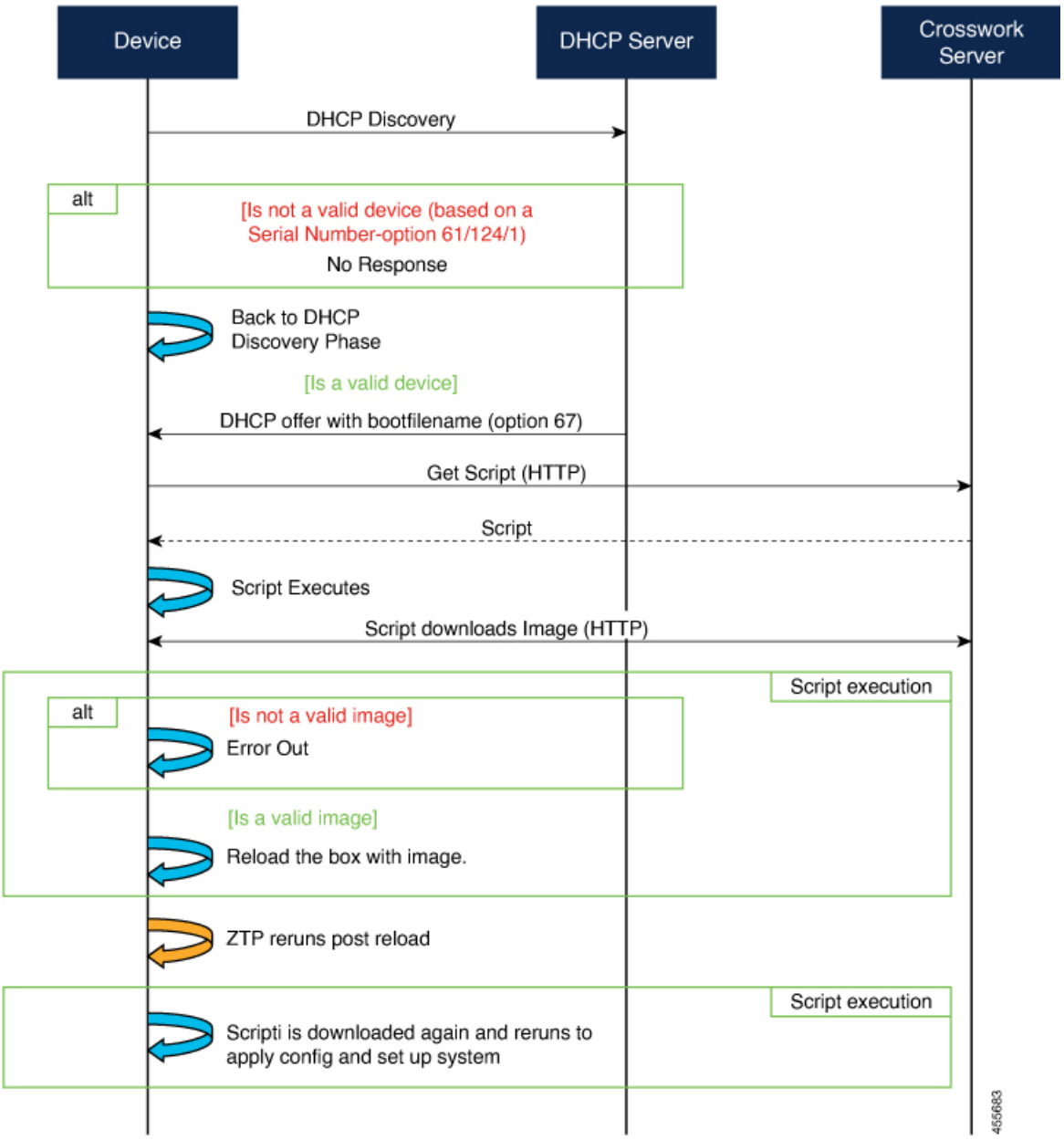
ZTP Processing Logic

Cisco Crosswork ZTP processing differs depending on whether you choose to implement Classic ZTP or Secure ZTP.

Classic ZTP Logic

The following illustration shows the processing logic that Classic ZTP uses to provision and onboard devices. The DHCP server verifies the device identity based on the device serial number, then offers downloads of the boot file and image. Once ZTP images the device, the device downloads the configuration file and executes it.

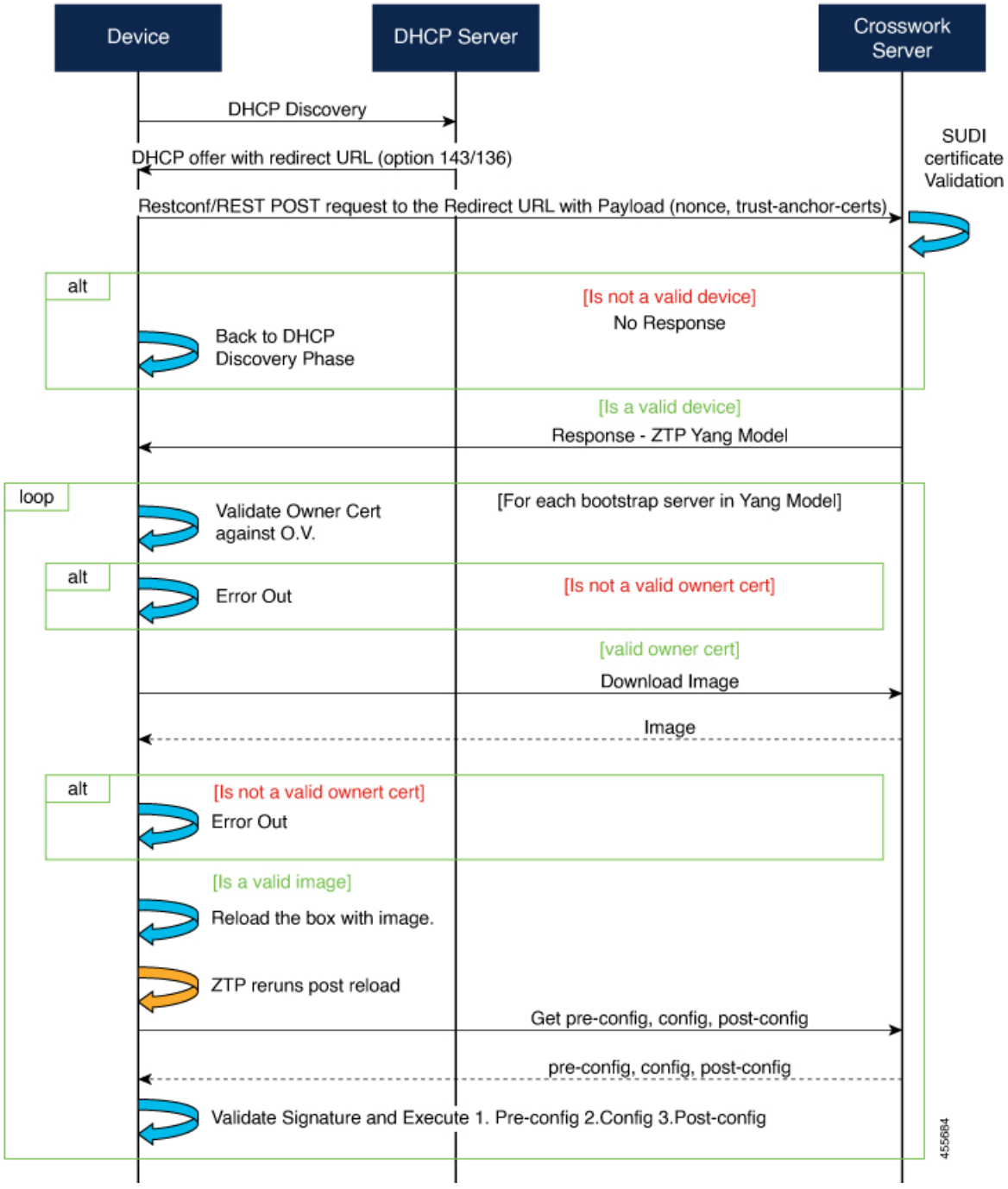
Figure 1: Classic ZTP Processing Logic



Secure ZTP Logic

The following illustration shows the process logic that Secure ZTP uses to provision and onboard devices. The device and the ZTP bootstrap server authenticate each other using the Secure Unique Device Identifier (SUDI) on the device and server certificates over TLS/HTTPS. Over a secure HTTPS channel, the bootstrap server lets the device download signed image and configuration artifacts. These artifacts must adhere to the RFC 8572 YANG schema. Once the device installs the new image (if any) and reloads, the device downloads configuration scripts and executes them.

Figure 2: Secure ZTP Processing Logic



ZTP State Transitions

Once initiated by a device reset or reload, the ZTP process proceeds automatically. Cisco Crosswork also updates the Zero Touch Devices window with status messages showing which stage of the process each device

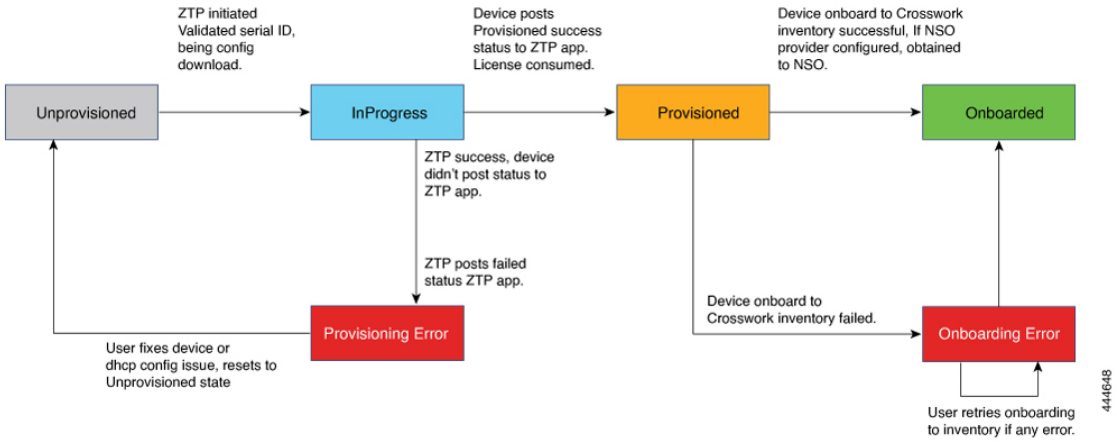
has reached. The states and their transitions differ between Classic and Secure ZTP, as explained in the next two sections.

The configuration scripts you use with ZTP must report device state changes to Cisco Crosswork using Cisco API calls. Failure to do so means Cisco Crosswork can't register state changes when they occur, resulting in failed provisioning and onboarding. To see examples of these calls, select **Device Management** > **ZTP Configuration Files**, then click **Download Sample Script**.

Classic ZTP State Transitions

The following figure shows the state changes for Classic ZTP processing.

Figure 3: Classic ZTP Device State Transitions



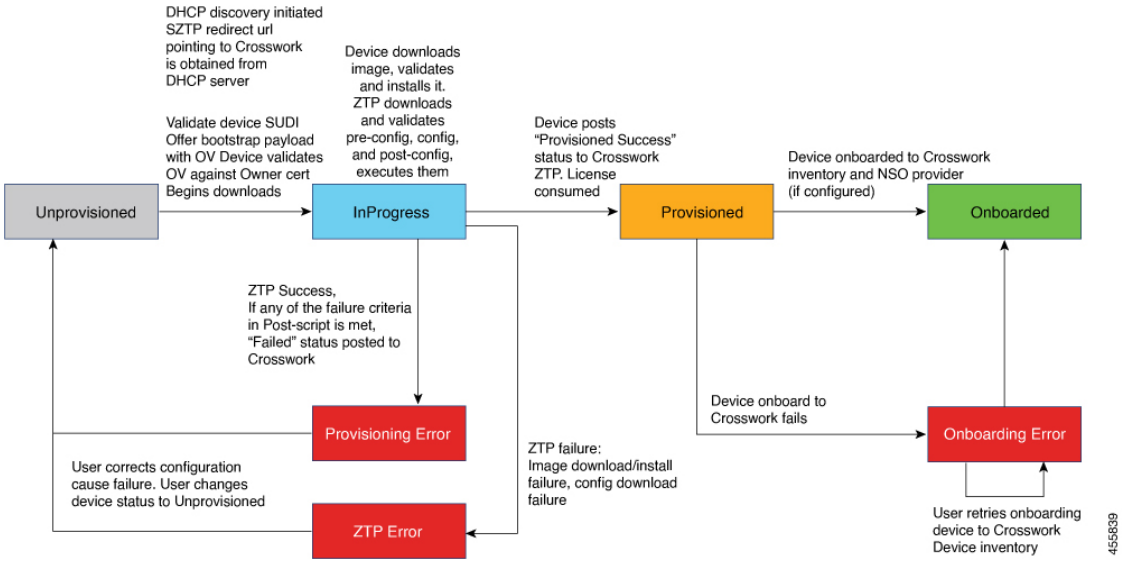
Classic ZTP device entries start out in the **Unprovisioned** state. After you initiate ZTP, devices transition to the **InProgress** state, when they connect to the network and start downloading image and configuration files. The device remains in the **InProgress** state until it reports either that it ran into a **Provisioning Error** or that it's **Provisioned**. If provisioning is successful, the device transitions to the **Provisioned** state. Once provisioned, Cisco Crosswork onboards the device. If Cisco NSO is a Cisco Crosswork provider, Cisco NSO also onboards the device. If onboarding is successful, the device state changes to **Onboarded**. It's now part of inventory and you can monitor and manage it like any other Cisco Crosswork network device.

Classic ZTP is successful when the device loads its image and/or configuration code successfully, connects with Cisco Crosswork, and reports a **Provisioned** status. This status change causes one license to be counted for that device serial number. Because the license is tied to the serial number, later transition to the **Onboarded** state, or any further ZTP processing, doesn't affect the license count.

Secure ZTP State Transitions

The following figure shows the state changes for Secure ZTP processing.

Figure 4: Secure ZTP State Transitions



Secure ZTP device entries start out in the **Unprovisioned** state. After you initiate ZTP, the device and the bootstrap server validate each other and the payload. The two use the device SUDI, ownership vouchers and device owner certificates to validate over HTTP/TLS. After validation, the device entry transitions to the **InProgress** state, when it connects to the network and starts downloading image and configuration files. The device remains in the **InProgress** state until it posts a **Provisioning Error**, **ZTP Error** or **Provisioned** status to Cisco Crosswork. If the provisioning is successful, the device transitions to the **Provisioned** state. Once provisioned, Cisco Crosswork onboards the device. If Cisco NSO is a Cisco Crosswork provider, Cisco NSO also onboards the device. If onboarding is successful, the device state changes to **Onboarded**. It's now part of inventory and you can monitor and manage it like any other Cisco Crosswork network device.

If any of the validation steps fail, Secure ZTP posts a **Provisioning Error**. If the image or configuration code fails verification or installation, Secure ZTP posts a **ZTP Error** instead. Like Classic ZTP, Secure ZTP is successful when the device loads its image and/or configuration code successfully, connects with Cisco Crosswork, and posts a **Provisioned** status. License consumption is the same as in Classic ZTP.

ZTP and Evaluation Licenses

All licenses start with an evaluation period of 90 days. When the evaluation period expires, Cisco Crosswork displays a banner warning users that evaluation licenses have expired. While ZTP displays this banner, it blocks some operations, including configuration downloads. When your organization enrolls in smart licensing and applies licenses to some of the onboarded devices, ZTP removes the block. ZTP displays the warning banner until you license all your onboarded devices.

Your onboarded ZTP devices are always associated with either:

- A serial number, or
- The values of the Option 82 location ID attributes (remote ID and circuit ID).

Serial numbers and location IDs form an "allowed" list. ZTP uses this list when deciding to onboard a device and assign it a license. If you delete an onboarded ZTP device from inventory, and then onboard it again later, use the same serial number or location ID. If you use a different serial number or location ID, you may consume

an extra license. The current release provides no workaround for this scenario. In any case, you can't have two different ZTP devices with the same serial number or location ID active at the same time.

Platform Support for ZTP

This topic details Cisco Crosswork Zero Touch Provisioning support for Cisco and third-party software and devices.

Platform Support for Classic ZTP

The following platforms support Classic ZTP:

- **Software**: Cisco IOS-XR versions 6.6.3, 7.0.1, 7.0.2, 7.0.12, and 7.3.1 or later.
- **Hardware**:
  - Cisco Network Convergence Systems (NCS) 540 Series Routers
  - Cisco NCS 1000-1004 Series Routers
  - Cisco NCS 5500 Series Routers
  - Cisco NCS 8000 and 8800 Series Routers (Spitfire fixed mode)

Classic ZTP doesn't support third-party devices or software.

Platform Support for Secure ZTP

The following platforms support Secure ZTP:

- **Software**: Cisco IOS-XR version 7.3.1 or later.

  You can upgrade from IOS-XR 6.6.3 to 7.3.1 as a single image installation.
- **Hardware**:
  - Cisco Network Convergence Systems (NCS) 540 Series Routers
  - Cisco NCS 1000-1004 Series Routers
  - Cisco NCS 5500 Series Routers
  - Cisco NCS 8000 and 8800 Series Routers (Spitfire fixed mode)

Secure ZTP supports provisioning for third-party devices only if the third-party devices:

- Are 100-percent compliant with the Secure ZTP RFC 8572(https://tools.ietf.org/html/rfc8572).
- Match Cisco format guidelines for serial numbers in device certificates and ownership vouchers. For details, see the following section, "Guidelines for Third-Party Device Certificates and Ownership Vouchers."

Guidelines for Third-Party Device Certificates and Ownership Vouchers

Secure ZTP processing for any device starts with a successful HTTPS/TLS handshake between the device and Cisco Crosswork. After the handshake, Secure ZTP must extract a serial number from the device certificate.

Secure ZTP then validates the extracted serial number against its internal "allowed" list of serial numbers. You create the allowed list by uploading device serial numbers to Cisco Crosswork. A similar serial-number validation step occurs later, when validating downloads using ownership vouchers.

Unlike Cisco IOS-XR devices, the format of the serial number in third-party vendors' device certificates is not standardized across vendors. Typically, a third-party vendor's device certificate has a `Subject` field or section. The `Subject` contains multiple key-value pairs that the vendor decides upon. One of the key-values pairs is usually a `serialNumber` key. This key's value contains the actual device serial number as a string, which is preceded by the string `SN:`. For example: Let's suppose that the third-party device certificate's `Subject` section contains the following key and value: `serialNumber = PID:NCS-5501 SN:FOC2331R0CW`. Secure ZTP will take the value after the `SN:` string and match that to one of the serial numbers in the allowed list.

If the third-party vendor's device certificate has a different format, validation failures can occur. The degree of failure depends on the degree of difference. The vendor certificate may not match this format at all. The certificate's `Subject` field may not contain a `serialNumber` key with a value that contains the `SN:` string. In this case, Secure ZTP processing falls back to using the whole string value of the `serialNumber` key (if present) as the device serial number. It will then try to match that value to one in the allowed list of serial numbers. These two methods – string matching and the fallback – are the only means Secure ZTP has for determining the third-party device's serial number. If the vendor certificate differs from this expectation sufficiently, Secure ZTP may be unable to validate the device at all.

Secure ZTP has similar format expectations for ownership vouchers. Cisco tools generate ownership vouchers with filenames in the format *`SerialNumber`*`.vcj`, where *`SerialNumber`* is the device's serial number. Secure ZTP extracts the serial number from the filename and then attempts to match it to one in the allowed list. For multivendor support, we assume that third-party vendor tools generate OV files in the same format. If this expectation isn't met, validation failures are likely.

ZTP Implementation Decisions

ZTP offers a range of implementation choices and cost vs. benefit tradeoffs worth considering in advance:

- **When to Use Classic ZTP**: Classic ZTP is easier to implement than Secure ZTP. It needs no PDC, owner certificates, or ownership vouchers. It’s less subject to processing errors, as device and server verification is less stringent and setup is less complex. It’s your only choice if your Cisco devices run IOS XR versions earlier than 7.3.1, as Secure ZTP doesn’t support them. Although Classic ZTP new includes a device serial-number check, it remains insecure at the transport layer. It's not recommended if routes to your remote devices cross a metro or otherwise unsecured network.
- **When to Use Secure ZTP**: Use Secure ZTP when you must traverse public networks and you have devices that support Secure ZTP. The additional security that it provides requires a more complex setup than Classic ZTP. This complexity can make processing error-prone if you’re new to the setup tasks. Secure ZTP setup also requires a certificates and ownership vouchers from the device manufacturer. Use it if your devices are from third-party manufacturers, as Classic ZTP doesn't support third-party hardware. Third-party devices and their software must be 100-percent compliant with RFCs 8572 and 8366. Device certificates for third-party devices must contain the device serial number. Third-party ownership vouchers must be in a format that uses the device serial number as the filename. Cisco can’t guarantee Secure ZTP compatibility with all third-party devices. For more details on third-party device support, see Platform Support for ZTP, on page 8.
- **Use ZTP With Imaged Devices**: There’s no need to specify a software image when you use Classic or Secure ZTP. This feature allows you the option of shipping to your remote location one or more devices on which you have already installed a software image. You can then connect to these devices and trigger ZTP processing remotely. Depending on how you set up things, you can apply:

- A configuration only
- One or more images or SMUs, with more configurations.

All licenses start with an evaluation period of 90 days. When the evaluation period expires, Cisco Crosswork displays a banner warning users that evaluation licenses have expired. While ZTP displays this banner, it blocks some operations, including configuration downloads. When your organization enrolls in smart licensing and applies licenses to some of the onboarded devices, ZTP removes the block. ZTP displays the warning banner until you license all your onboarded devices.

Secure ZTP offers more flexibility with preimaged devices because it offers preconfiguration, day-zero, and postconfiguration script execution capability. But both ZTP modes can chain configuration files that load images, SMUs and configurations.

In both cases, the result is to onboard the device. Once onboarded to Cisco Crosswork, you can't use ZTP to configure the device.

- **Organize Configurations**: Keep your configurations as consistent as possible across devices. Consistency makes solving problems easier. It minimizes the amount of extra configuration you must perform to bring new devices online. It also reduces the number of "special" things to keep in mind when it comes time to reconfigure or upgrade your devices. Start by ensuring that all devices from the same device family and with similar roles have the same or similar basic configurations.

  How you define the role that a device plays depends on your organization, its operational practices, and the complexity of your network environment. For example: Suppose that your organization is a financial services enterprise. It has three types of branches: Sidewalk ATMs, retail branches open during standard business hours, and private trading offices. You could define three sets of basic profiles covering all the devices at each type of branch. You can map your configuration files to each of these profiles.

  Another method of enforcing consistency is to develop basic script configurations for similar types of devices, then use the script logic to call other scripts. If you're using Classic ZTP, the script is in the specified configuration file. That script downloads the basic configuration, then downloads other scripts depending on the branch type. If using Secure ZTP, you have even more flexibility, as you can specify preconfiguration and postconfiguration scripts in addition to the main or day-zero configuration script.

ZTP Setup Workflow

Zero touch provisioning requires you to complete the following setup tasks first, before you can trigger ZTP boot and configuration:

1. Make sure that your environment meets ZTP prerequisites for security, provider configuration, and device connectivity.
2. Assemble the assets ZTP needs for processing. These assets include:
   - The software image to install.
   - The configurations to apply.
   - Credentials to access the device.
   - Device serial numbers.

   If you're using Secure ZTP, these assets also include device owner certificates, the PDC, and ownership vouchers.

3. Load into Cisco Crosswork the ZTP assets you have assembled.
4. Create credential profiles using the credential assets that you assembled.
5. Prepare ZTP device entry files. These files create the Cisco Crosswork device entries that ZTP uses to onboard the devices to the Cisco Crosswork device inventory. If you have many devices to onboard, create the entries in bulk by importing a CSV file. If you have only a few devices to onboard, it's more convenient to prepare these entries one by one, using the Cisco Crosswork UI.

The remaining topics in this section discuss how to perform each of these tasks.

Meet ZTP Prerequisites

For compatibility with ZTP, your Cisco Crosswork installation must meet the following prerequisites:

- If you're using Classic ZTP to onboard any device, ensure that Cisco Crosswork and the devices are in a secure network domain.
- If you want ZTP to onboard your devices to Cisco NSO, configure NSO as a Cisco Crosswork provider. Be sure to set the NSO provider property key to `forward` and the property value to `true`.
- The Cisco Crosswork cluster must be reachable from the devices, and the cluster from the devices, over either an out-of-band management network or an in-band data network. For a general indication of the scope of these requirements, see the network diagrams in the "Network Requirements" section of the *Cisco Crosswork Infrastructure 4.0 and Applications Installation Guide*. Enabling this kind of access may require you to add static routes and change firewall configurations.

Assemble ZTP Assets

Both Classic and Secure ZTP require that you collect the following ZTP assets:

- **Software images**: The installable operating system software, such as Cisco IOS-XR, that enables the network device to function. Cisco distributes images as TAR, ISO, or RPM files. Each image file represents a single release of the given network OS for a given device platform or family. Upload image files to Cisco Crosswork one at a time, and enter each the MD5 checksum for each software image file. Cisco Crosswork uses the MD5 checksum to validate the integrity of the file. Be sure to record the checksum when you download device images from Cisco or any third-party manufacturer. You can also generate your own MD5 checksum for an image you want to upload.
- **Software Maintenance Updates (SMUs)**: Cisco software packages that provide point fixes for one or more critical issues in a given software release. Cisco distributes SMUs in nonbootable format with a readme.txt file explaining the associated issues. Cisco rolls SMU contents into the next maintenance release of a software image. Apply SMUs using configuration files, not during software image download. Upload SMUs to Cisco Crosswork one at a time.

  Cisco customers with current devices and valid support contracts can find and download Cisco software images and SMUs using the Cisco Support & Downloads page.
- **Configurations**: ZTP uses configuration files to configure the features of the installed software image on a given device, including upgrading the software using SMUs. Configuration files can be Linux shell scripts (SH), Python scripts (PY), or device operating system CLI commands stored in an ASCII text file (TXT). Your organization or consultants create configurations. Upload configuration files to Cisco Crosswork one at a time. Your custom configuration code can use replaceable parameters and must use

Cisco Crosswork API calls to complete many tasks. In particular, the code must use API calls to notify the Cisco Crosswork server when the device transitions from one ZTP state to another. For examples of how to use these parameters and API calls, see the sample ZTP configuration file. You can download the sample ZTP configuration file from Cisco Crosswork by selecting **Device Management** > **ZTP Configuration Files**, then clicking **Download Sample Script**. For more details, see the following sections, "Default Replaceable Parameters" and "Create Custom Replaceable Parameters". Secure ZTP allows you to load pre-, post-, and day-zero configuration files.

- **Credentials**: The user names and passwords that Cisco Crosswork uses to access a device and control it. You load them as credential profiles, and Cisco Crosswork stores them in encrypted form. You can create credential profiles one at a time using the GUI, or load them in bulk by downloading and modifying a credential profile CSV file.
- **Serial numbers**: The serial numbers of the devices you plan to onboard using ZTP. Enter serial numbers for each device you're planning to onboard using either Classic or Secure ZTP. Load serial numbers in bulk, by importing a CSV file, before creating device entries. If you're planning to use Secure ZTP, submit the serial numbers to Cisco when requesting ownership vouchers.

If you plan to use Secure ZTP, assemble the following extra ZTP assets:

- **Owner certificates**: Load both the owner certificates and the owner key to Cisco Crosswork, so it can generate leaf certificates for each of your devices.
- **Pinned Domain Certificate (PDC)**: Load the PDC to Cisco Crosswork along with your owner certificates. You also submit the PDC to Cisco when requesting ownership vouchers.
- **Ownership vouchers (OVs)**: Load your OVs with the other certificates. Submit your PDC and device serial numbers when you request OVs from Cisco or third party manufacturers. Cisco returns the OVs to you when they are ready, as one or more VCJ files in a TARball. This exchange takes place using a secure method agreed upon by you and your Cisco account team. If you're using vouchers for third-party devices, the VCJ files the manufacturer supplies must follow the naming convention `serial.vcj`, where `serial` is the serial number of the corresponding device. Cisco Crosswork requires this file naming convention in order to map the ownership voucher to the device.
- **SUDI Root CA certificates**: Load SUDI Root CA certificates at the same time as other certificates and OVs. Cisco SUDI root certificates are available for customer download at the Cisco PKI: Policies, Certificates, and Documents page (https://www.cisco.com/security/pki/policies/index.html).

Some organizations maintain libraries of approved assets. If your organization has a library like this, ensure that these assets are easily accessible from your client machine. Doing so makes it easier for you to complete ZTP setup.

Default Replaceable Parameters

The following table lists the default replaceable parameters you can use in your custom configuration files. At runtime, for each of these placeholders, Cisco Crosswork substitutes the appropriate values for each device. For an example of the use of these placeholders, download the sample configuration script from Cisco Crosswork as explained in the preceding section of this topic.

Table 1: Default Parameters in ZTP Configuration Files

Cisco Crosswork substitutes this placeholder...	...using the value from the...
{$HOSTNAME}	Host name of the device as specified in the ZTP device entry.
{$IP_ADDRESS}	IP address of the device, as assigned by DHCP.
{$SSH_USERNAME}	The value of the **User Name** field in the credential profile (when the **Connectivity Type** is **SSH**).
{$SSH_PASSWORD}	The value of the **Password** field in the credential profile (when the **Connectivity Type** is **SSH**).
{$SSH_ENPASSWORD}	The value of the **Enable Password** field in the credential profile (when the **Connectivity Type** is **SSH**)
{$SNMP_READ_COM}	The value of the **Read Community** field in the credential profile (when the **Connectivity Type** is **SNMPv2**).
{$SNMP_WRITE_COM}	The value of the **Write Community** field in the credential profile (when the **Connectivity Type** is **SNMPv2**).
{$SNMP_SEC_LEVEL}	The value of the **Security Level** field in the credential profile (when the **Connectivity Type** is **SNMPv3**).
{$SNMP_USERNAME}	The value of the **User Name** field in the credential profile (when the **Connectivity Type** is either **SNMPv2** or **SNMPv3**).
{$SNMP_AUTH_TYPE}	The value of the **User Name** field in the credential profile (when the **Connectivity Type** is **SNMPv3** and **Security Level** is **AUTH_NO_PRIV**) or **AUTH_PRIV**).
{$SNMP_AUTH_PASS}	The value of the **User Name** field in the credential profile (when the **Connectivity Type** is **SNMPv3** and **Security Level** is **AUTH_NO_PRIV** or **AUTH_PRIV**).
{$SNMP_PRIV_TYPE}	The value of the **User Name** field in the credential profile (when the **Connectivity Type** is **SNMPv3** and **Security Level** is **AUTH_PRIV**).
{$SNMP_PRIV_PASS}	The value of the **Priv Password** field in the credential profile (when the **Connectivity Type** is **SNMPv3** and **Security Level** is **AUTH_PRIV**).

Custom Replaceable Parameters

You can create your own replaceable parameters in configuration files, as shown in the following example.

```
!
hostname {$name}
username {$ssh_name}
 group root-lr
 group cisco-support
 secret {$ssh_pwd}
!
```

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

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

Load ZTP Assets

Before creating credential profiles, upload the ZTP assets you assembled.

Both Classic and Secure ZTP require you to load:

- Software images
- SMUs
- Configuration files
- Device serial numbers

Secure ZTP requires you to load:

- Pinned domain certificate
- Ownership certificates
- Ownership Vouchers

You may use a mapped network drive to upload software images, SMUs, and configuration files.

Cisco Crosswork checks for duplicate serial numbers and merges them into single entries automatically. Cisco Crosswork also associates all uploaded ownership vouchers with existing serial numbers automatically.

You can upload images, configuration files, and serial numbers in any order. Load the certificates and ownership vouchers only after loading serial numbers.

Step 1 Upload images and SMUs:

a) From the main menu, select **Device Management** > **Software Images** and then click ⊞.

b) Enter the required image or SMU file information and then click **Add**.

You must enter the MD5 checksum for the file.

You can also click **Browse** to select the ISO, TAR, or RPM file.

c) Click ⊞ and repeat step 1b until you have loaded all the image and SMU files.

Step 2 Upload configuration files and scripts:

a) From the main menu, select **Device Management** > **Configuration Files** and then click the +.
b) Enter the required configuration file information and then click **Add**. You can click **Browse** to select the PY, SH, or TXT configuration file.
c) Click + and repeat step 2b until you have loaded all the configuration files. If you're implementing Secure ZTP, include your pre-, post-, and main or day-zero configuration files.

Step 3 Upload device serial numbers:

a) From the main menu, select **Device Management** > **Serial Number and Voucher**, then click **Add Serial Number**.
b) Click **Upload CSV**, then click the **serialnumber.csv** link to download the sampleSerialnumber.csv file.
c) Using your choice of CSV file editor, enter into the template the serial numbers for all the devices you plan to onboard using ZTP. Save the updated CSV file template under a new name.
d) Select **Add Serial Number** again. Click **Browse** to select the updated CSV file, then click **Add Serial Number** to import the serial numbers.

Step 4 Continue with the following steps if you plan to implement Secure ZTP.

Step 5 Upload your pinned domain certificate, owner certificates, and SUDI Root CA certificates:

a) From the main menu, select **Administration** > **Certificate Management**, then click +.
b) In **Certificate Name**, enter a name for this certification grouping.
c) In **Certificate Role**, select **Secure ZTP Provisioning**.
d) Click **Browse** to select the **Pinned Domain CA Certificate**, **Owner Certificate**, and **Owner Key**files.
e) Click **Save**.

Step 6 Upload ownership vouchers:

a) From the main menu, select **Device Management** > **Serial Number and Voucher**, then click **Add Voucher**.
b) Click **Browse** to select the Cisco-supplied VCJ file (or, if there's more than one voucher, the TARball containing the ownership vouchers) . Then click **Upload**.

If you're uploading vouchers for third-party devices, the uploaded VCJ file or files in the TARball must follow the naming convention `serial.vcj`. In this convention, `serial` is the serial number of the corresponding device. Cisco Crosswork requires naming in order to map the ownership voucher to the device.

Create Credential Profiles for ZTP

Cisco Crosswork ZTP requires credential profiles in order to access and configure your devices. The following steps show how to add them in bulk using a CSV file. To add credential profiles one by one, select **Device Management** > **Credential Profiles**, then click the +.

It's good practice to create SNMP credential profiles only for the version of SNMP enabled on the device. For example: If only SNMPv2 is enabled in the device configuration, don't include SNMPv3 credentials in the profile.

Step 1 From the main menu, choose **Device Management** > **Credential Profiles**.

Step 2 Click the ⎘.

Step 3 Click the **Download sample 'Credential template (*.csv)' file** link and save the CSV file template locally.

Step 4 Open the CSV template using your preferred editor. Begin adding rows to the file, one row for each credential profile you want to create.

As you do, observe these guidelines:

- If the **Password** column for any credential profile is blank, you can't import the CSV file. If you wish, you can enter the actual passwords in these fields. Cisco Crosswork stores them in encrypted form. If you choose this method, be sure to destroy the CSV file immediately after upload. We recommend using asterisks to fill the **Password** column in the CSV file and then importing it. After successful import, you can use the Cisco Crosswork GUI to edit each profile and enter the actual passwords, as explained in the following steps.
- Use a semicolon to separate multiple entries in the same field.
- When you separate multiple entries with semicolons, remember that the order in which you enter values in each field is important. The first entry in one column will map to the first entry in the next column, and so on. For example: Suppose you enter in **Password Type** this list of password types: **`ROBOT_USERPASS_SSH;ROBOT_USERPASS_TELNET;ROBOT_USERPASS_NETCONF`**. You then enter in the **User Name** column **`Tom;Dick;Harry;`** and in the **Password** column **`root;MyPass;Turtledove;`**. The order of entry in these three columns determines the resulting mapping between the values you entered:
  - ROBOT_USERPASS_SSH: Tom : root
  - ROBOT_USERPASS_NETCONF: Dick : MyPass
  - ROBOT_USERPASS_TELNET: Harry : Turtledove
- Be sure to delete sample data rows before saving the file. You can ignore the column header row.

Step 5 When you're finished, save the CSV file to a new name.

Step 6 If necessary, choose **Device Management** > **Credential Profiles** again, then click the .

Step 7 Click **Browse** to navigate to the CSV file and select it.

Step 8 With the CSV file selected, click **Import**.

Step 9 When the import is complete:

a) From the left-hand side of the **Credential Profiles** window, select the profile you want to update, and click .
b) Enter the passwords and community strings for the credential profile and then click **Save**.
c) Repeat these steps as needed until you have entered all passwords and community strings.

Create ZTP Profiles

Cisco Crosswork uses ZTP profiles to automate imaging and configuration processes. While ZTP profiles are optional, we strongly recommend creating them, as they can help simplify the ZTP imaging and configuration process. Use ZTP profiles to help organize defined sets of image and configuration files you can apply to devices in a particular class or device family.

If you're implementing Classic ZTP, each ZTP profile can have only one image file and one configuration file associated with it. Secure ZTP allows you to specify pre-, post-, and main or day-zero configuration files.

ZTP profiles don't require that you specify an image file.

You can create as many ZTP profiles as you like. We recommend that you create only one ZTP profile for each device family, use case, or network role.

Step 1 From the main menu, choose **Device Management** > **Zero Touch Profiles**.

Step 2 Click **+ New Profile**.

Step 3 Enter the required values for the new ZTP profile. You don't need to specify a software image for the profile.

Step 4 If you're implementing Secure ZTP: Adjust the **Enable Secure ZTP** slider and enter the names of the pre- and post-configuration files.

Step 5 Click **Save** to create the new ZTP profile.

Prepare ZTP Device Entry Files

Cisco Crosswork uses ZTP device entries to let you specify in advance the IP addresses, protocols, and other information for the devices you want to provision. Cisco Crosswork populates these imported entries with more information once ZTP processing completes successfully.

You can create ZTP device entries in bulk by importing a device-entry CSV file.

The following topics explain how to download a template for a device entry CSV file. They also explain how to create properly formatted ZTP device entries.

We recommend that you experiment with the device entry CSV file format until you get used to it. Add only one or two device entries in a copy of the template, then import it. You can then see if you get the results you want.

You can also create ZTP device entries one by one, using the Cisco Crosswork UI, as explained in the next topic.

Download and Edit the ZTP Device Entry Template

1. From the main menu, choose **Device Management** > **Devices**.
2. Click the **Zero Touch Devices** tab.
3. Click the  .
4. Click the **Download 'devices import' template (.csv)** link and then **Save** it to a local storage resource. Click **Cancel** to clear the dialog box.
5. Open the CSV template with the application of your choice and save it to a new name. In each row, create an entry for each of the devices you plan to onboard using ZTP. Refer to the next topic section for help on the values to enter in each column.

ZTP Device Entry CSV Template Reference

The following table explains how to use the columns in the template. We mark columns that require entries with an asterisk (*) next to the column name.

The four "Connectivity" columns allow multiple entries, so you can specify multiple connectivity protocols for a single device. If you use this option, use semicolons between entries, and enter the values in the next three columns in the same order. For example: Suppose you enter `SSH;NETCONF;` in the **Connectivity Protocol** column. If you enter `23;830;` in the **Connectivity Port** column, the entries in the two columns map like this:

- SSH: 22
- NETCONF: 830

Table 2: ZTP Device Entry Template Column Reference

Column	Usage
UUID	Cisco Crosswork assigns a random UUID unless you elect to generate and enter it yourself. Enter the 128-bit universally unique identifier assigned to the device.
Host Name *	Enter the host name you want to assign to the device.
Serial Number *	Enter the device serial number. You can enter up to three serial numbers for the same device. These must be the same serial number for each device that you loaded into Cisco Crosswork previously. ZTP requires a serial number entry for all normal deployments. If you're using DHCP option 82 to implement a relay agent, you can leave this field blank, but you must specify a Remote Id and Circuit ID to identify the device.
MAC Address	Enter the device MAC address.
IP Address	Enter the device IP address (IPv4 or IPv6), along with its subnet mask in slash notation.
Credential Profile *	Enter the name of the credential profile you want Cisco Crosswork to use to access and configure the device. Required only if you want to use a credential profile.
OS Platform *	Enter the OS platform for the device. For example: IOS-XR.
Version *	Enter the OS platform version for the device software image. The platform version should be the same version as the ones specified for the image and configuration files you use to provision it. Currently, ZTP supports IOS-XR versions 6.6.3, 7.0.1, 7.0.2 and 7.0.12. Required only if you don't specify a ZTP profile in the Profile Name column.
Device Family *	Enter the device family for the device. The device family must match the device family in the image and configuration files ZTP uses to provision it. Required only if you don't specify a ZTP profile in the Profile Name column.
Image ID	Enter the Cisco Crosswork-assigned ID for the software image file you want to install on the device.
Config ID *	Enter the Cisco Crosswork-assigned ID for the configuration file you want to use when configuring the device.

Column	Usage
Profile Name	Enter the name of the ZTP profile you want to use to provision this device.
Configuration Attributes	Enter the values you want Cisco Crosswork to use for the replaceable parameters in the configuration file for the device. If you're using Secure ZTP, you can include pre-, post-, and day-zero configuration file parameters.
Connectivity Protocol	The connectivity protocols needed to monitor the device or to support Cisco Crosswork applications and features. Choices are: **`SSH`**, **`SNMPv2`**, **`NETCONF`**, **`TELNET`**, **`HTTP`**, **`HTTPS`**, **`GRPC`**, and **`SNMPv3`**.
Connectivity IP Address *	Enter the IP address (IPv4 or IPv6) and subnet mask for the connectivity protocol. Required only if you chose to set up a connectivity protocol.
Connectivity Port *	Enter the port used for this connectivity protocol. Each protocol maps to a port. Be sure to enter the port number that maps to the protocol you chose. Specify at least one port and protocol for every device, except if you want to: • Set the status of the onboarded device set as unmanaged or down. • Disable Cisco Crosswork reachability checks for the onboarded device. You may need to specify more than one protocol and port per device. The number of protocols and ports you specify depends on how you have configured Cisco Crosswork and the Crosswork applications you're using. See the table in the following section, "Crosswork Connectivity Protocol Requirements".
Connectivity Timeout	Enter the elapsed time (in seconds) before an attempt to communicate using this protocol times out. The default value is 30 seconds; the recommended timeout value is 60 seconds.
Provider Name	Enter the name of the provider to which you want to onboard the new ZTP devices. The name you enter must match exactly the name of the provider managing the device.
Provider Type	The type of provider. For example: NSO.
Provider Node ID	The IP address or URL of the main node of the provider.
Inventory ID	Enter the inventory ID you want to assign to the device.
Secure ZTP Enabled	Enter TRUE if you want to provision the device using Secure ZTP, or FALSE if not.

Column	Usage
PreConfig ID	Enter the Cisco Crosswork ID of the configuration script you want to run before running the associated configuration file.
PostConfig ID	Enter the Cisco Crosswork ID of the configuration script you want to run immediately after running the associated configuration file.
Location Enabled	Enter TRUE if you plan to identify the device using a location ID. Enter FALSE if you plan to identify it by serial number. If you enter TRUE, enter a Remote ID and a Circuit ID in the corresponding columns. If you enter FALSE, enter a Serial Number in the corresponding column.
Remote ID *	If implementing Secure ZTP and using option 82: Identify the name of the remote host acting as the bootstrap server. If you're using DHCP option 82 to implement a relay agent, this entry is required. You must enter a combination of the device RemoteID and CircuitID. If you're not using option 82, you can leave this field blank but you must specify the device serial number.
Circuit ID *	If implementing Secure ZTP and using option 82: Identify the interface or VLAN on which the bootstrap server receives requests. If you're using DHCP option 82 to implement a relay agent, this entry is required. You must enter a combination of the device RemoteID and CircuitID. If you're not using option 82, you can leave this field blank but you must specify the device serial number.
routingInfo.globalospfrouterid	If implementing OSPF on the device: Enter the OSPF Router ID for the device.
routingInfo.globalisissystemid	If implementing IS-IS on the device: Enter the IS-IS System ID for the device.
routingInfo.teRouterid	If implementing Traffic Engineering on the device: Enter the TE router ID for the device.

Crosswork Connectivity Protocol Requirements

Cisco Crosswork features and applications require you to enable a range of connectivity protocols for each device. The following table identifies these requirements for each supported connectivity protocol.

Table 3: Connectivity Protocol Requirements for Applications and Features

Protocol	Port	Application	Feature
GRPC	9090	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine (COE)	• Cisco Crosswork API communication
HTTP	80	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine (COE)	• Onboarding of NSO provider with all three applications
HTTPS	443	• Cisco Crosswork Network Controller (CNC)	• Onboarding of NSO provider
NETCONF	830	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine	• Onboarding of NSO provider with all 3 applications
SNMPv2	161	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine	• SNMPv2 data collection

Protocol	Port	Application	Feature
SNMPv3	161	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine	• SNMPv3 data collection
SSH	22	• Cisco Crosswork Network Controller (CNC) • Cisco Crosswork Change Automation and Health Insights (CAHI) • Cisco Crosswork Optimization Engine	• CLI data collection, SSH access to devices

Prepare Single ZTP Device Entries

If you have only a few devices to onboard using ZTP, you may find it easier to create the device entries one by one. Use the ZTP user interface and the following instructions to create single ZTP device entries.

Step 1 From the main menu, choose **Device Management** > **Devices**.

Step 2 Click the **Zero Touch Devices** tab.

Step 3 Click  .

Step 4 Enter values for the new ZTP device entry.

After ZTP onboards your devices, Cisco Crosswork may display more attributes.

Step 5 Click **Save**.

ZTP Provisioning Workflow

Once you complete ZTP setup, you can provision your devices and maintain them, as follows:

1. Set up DHCP so that Cisco Crosswork can download image and configuration software securely after you trigger ZTP processing.
2. Upload to Cisco Crosswork the ZTP device entry CSV file you created. Importing the file creates the device entries that ZTP populates during onboarding. If you're onboarding only a few ZTP devices, create device entries using the ZTP user interface instead.
3. Trigger ZTP processing by power-cycling or performing a CLI reboot for each device.

4. Complete the information for the onboarded devices. Edit them and supply (for example) geographical location information that ZTP couldn't discover during provisioning.

After completing this core workflow, you can perform ongoing maintenance of your ZTP devices using the advice and methods in the following topics:

- Update ZTP devices with additional information.
- Reconfigure your ZTP devices after onboarding, using other applications or by deleting and reonboarding the devices.
- Retire or replace ZTP devices without consuming more device licenses.
- Perform housekeeping on the ZTP assets you used to onboard your devices.
- Troubleshoot issues with ZTP processing and devices.

The remaining topics in this section discuss how to perform each of these tasks.

Upload ZTP Device Entries

The following steps explain how to create multiple ZTP device entries by importing your previously prepared ZTP device entry CSV file.

Imported ZTP device entries always appear in the **Zero Touch Devices** tab with their **Status** set to **`Unprovisioned`**. They remain **`Unprovisioned`** until you trigger ZTP processing.

Step 1 From the main menu, choose **Device Management** > **Devices**.

Step 2 Click the **Zero Touch Devices** tab.

Step 3 Click **Import Devices**.

Step 4 Click **Browse** to navigate to the ZTP device entry CSV file you created and then select it.

Step 5 With the CSV file selected, click **Import**.

Set Up DHCP for Crosswork ZTP

Before triggering ZTP processing, update your DHCP configuration file with information that identifies your ZTP devices and the software applied to them. This information permits Cisco Crosswork and DHCP to identify the ZTP devices and respond to requests for network connection and file downloads.

The following topics provide examples showing how to update DHCP server configurations to meet this requirement. The examples in these topics assume the DHCP context settings shown in the following figure. The figure shows example settings for the Internet Systems Consortium DHCP server. The line enabling the `sztp-redirect` option is required for Secure ZTP only. Leave it out if you're using Classic ZTP.

Figure 5: Secure ZTP DHCP Context

```
#
authoritative;

default-lease-time 7200;
max-lease-time 7200;
# Next line is needed for Secure ZTP only;
```

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

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

DHCP Setup for Classic ZTP

We strongly recommend that you use Classic ZTP to provision devices over secure network domains only.

Cisco devices supported by Classic ZTP allow iPXE software image downloads via HTTP only. These same devices support download of configuration files via either HTTP or HTTPS. These options require entry of DHCP bootfile URLs in the DHCP server configuration for your organization.

If you want to use HTTP for both image and configuration file downloads, these URLs must specify the HTTP protocol and port 30604. For help, see the examples in figures 1 and 2.

If you want to use HTTPS for configuration file downloads only, the URL must specify the HTTPS protocol and port 30603. Specify the -k option before the HTTPS protocol in the URL. For help, see the examples in figures 3 and 4.

ZTP permits use of DHCP option 82 for configuration downloads. Option 82, also known as the DHCP Relay Agent Information Option, helps protect your devices from attacks using IP and MAC spoofing or DHCP address starvation. Option 82 allows you to specify an intermediary, or relay, router located between the device you're onboarding and the DHCP server resolving device requests. To use this option, specify a location ID. The location ID consists of a circuit ID (interface or VLAN ID) and remote ID (host name). Specify these values as parameters of the configuration download URL, as shown in the examples in figures 2 and 4. For more information about option 82, see RFC 3046 (http://tools.ietf.org/html/rfc3046).

When following these examples:

- Be sure to replace <*CW_HOST_IP*> with the IP address of your Cisco Crosswork server.
- Replace <*IMAGE_UUID*> with the UUID of the software image file in the ZTP repository. For help with using bootfile names and UUIDs, see the later section of this topic, "Copy Bootfile Names and UUIDs for DHCP Setup".
- Configuration files do not require UUIDs.

Figure 6: Classic ZTP DHCP Setup, Using HTTP

```
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";
  }
}
```

Figure 7: Classic ZTP DHCP Setup, Using HTTP and Option 82

```
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";
  }
}
```

Figure 8: Classic ZTP DHCP Setup, Using HTTPS

```
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";
  }
}
```

Figure 9: Classic ZTP DHCP Setup, Using HTTPS and Option 82

```
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";
  }
}
```

DHCP Setup for Secure ZTP

Secure ZTP allows you to provision devices over both secure and insecure network domains. Use HTTPS for the configuration file download, and specify `option sztp-redirect` for configuration artifacts. Add a remote ID and circuit ID if you want to use option 82. The remote ID identifies the remote host acting as the bootstrap server, and the circuit ID identifies the interface or VLAN on the remote host. See the examples in figures 5 and 6. For help with using bootfile names and UUIDs, see the following section, "Copy Bootfile Names and UUIDs for DHCP Setup".

Figure 10: Secure ZTP DHCP Setup, Using HTTPS

```
host sztp1 {
 hardware ethernet 00:a7:42:86:54:f4;
  if exists user-class and option user-class = "iPXE" {
    filename = "http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";

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

  }
}
```

Figure 11: Secure ZTP DHCP Setup, Using HTTPS and Option 82

```
host sztp2 {
 hardware ethernet 00:a7:42:86:54:f5;
  if exists user-class and option user-class = "iPXE" {
    filename = "http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/<IMAGE_UUID>";

  } else if exists user-class and option user-class ="exr-config" {
```

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

  }
}
```

Copy Bootfile Names and UUIDs for DHCP Setup

When modifying your DHCP server configuration file, specify the bootfile name and UUID for each software image. You can quickly copy both to your clipboard directly from the list of software images that you have already uploaded to Cisco Crosswork. No UUID is required for configuration files.

To copy software image bootfile names and UUIDs:

1. From the main menu, choose **Device Management** > **Software Images**.
2. If you want to copy:
   - The bootfile name and UUID of the software image: Click the ⧉ in the **Image/SMU Name** column.
   - Only the UUID of the software image: Click the ⧉ in the **Image UUID** column.

   Cisco Crosswork copies the bootfile name and/or UUID to your clipboard. You can now paste it into your DHCP host entry.

   When using the copied file path to create DHCP host entries, replace the *IP* variable with the IP address and port of your Cisco Crosswork server.

Generic Internet Systems Consortium (ISC) DHCP Setup Examples

The following figures show examples of the type of host entries you would make for a Classic ZTP and for a Secure ZTP device in the `/etc/dhcp/dhcp.conf` configuration file of an Internet Systems Consortium (ISC) DHCP server.

Other third-party DHCP servers differ in overall implementation, but many use options and formats similar to these ISC examples.

Be sure to reload or restart the ISC DHCP server once you have finished creating these new entries.

Figure 12: Classic ZTP ISC IPv4 DHCP Configuration Example

```
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";
    }
}
```

Figure 13: Classic ZTP ISC IPv6 DHCP Configuration Example

```
host 5501
{
    host-identifier option dhcp6.client-id
```

```
00:02:00:00:00:09:46:4f:43:32:33:30:38:52:30:53:33:00;
    fixed-address6 fc00:15:2::36;
    if exists dhcp6.user-class and substring(option dhcp6.user-class, 2, 4) = "iPXE" {
     option dhcp6.bootfile-url "http://<CW_HOST_IP>:30604/crosswork/imagesvc/v1/device/files/

        <IMAGE_UUID>";
    } else {if exists dhcp6.user-class and substring(option dhcp6.user-class, 0, 10) =
"exr-config" {
      option dhcp6.bootfile-url
"http://<CW_HOST_IP>:30604/crosswork/crosswork/configsvc/v1/file";
    }
}
```

Figure 14: Secure ZTP ISC IPv4 DHCP Configuration Example

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

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

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

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

  }

}
```

Figure 15: Secure ZTP ISC IPv6 DHCP Configuration Example

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

option dhcp6.info-refresh-time 21600;
subnet6 fc00::/64 {
 range6 fc00::10:10:101 fc00::10:10:105;
}
 host CW14-NCS {

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

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

 }

 }
```

The following table describes each line in the IPv4 ISC DHCP device entry examples given, and the source of the values used. The descriptions apply to both Classic ZTP and Secure ZTP for IPv4. Descriptions for the entries in the IPv6 example are identical, but adapted for the IPv6 addressing scheme.

Table 4: ISC IPv4 DHCP Configuration Host Entries and Values

IPv4 Entry	Description
`host NCS5k-1`	The device entry host name. The host name can be the same as the actual assigned host name, but need not be.
`option dhcp-client-identifier`	The unique ID of the device entry. The value `"FOC2302R09H"` shown in the Classic ZTP and IPv4 example is the serial number of the device. You can find the serial number on the device chassis. If you don't have physical access to the device, the IOS-XR `show inventory` command provides the serial number.
`hardware ethernet 00:cc:fc:bb:be:6a`	The MAC address of the Ethernet NIC port on the device. This address is the address on which you trigger the ZTP process. The address can be a management or data port, as long as it's reachable from Cisco Crosswork.
`fixed-address 105.1.1.16`	The IP address to be assigned to the device during configuration. The example is for a static IP, but you can also use standard DHCP IP pool assignment commands.
`option user-class = "iPXE"` and `filename =`	This line checks that the incoming ZTP request contains the `"iPXE"` option. Classic ZTP uses this option to image the device. If the request includes this option, then the device downloads the image file matching the UUID and path specified in the `filename =` parameter.
For Classic ZTP: `option user-class = "exr-config"` and `ffl filename =` For Secure ZTP: `option sztp-redirect code 143=text`	This line checks that the incoming ZTP request contains the `"exr-config"` option. ZTP uses this option to configure the device. If the request includes this option, then the device downloads the configuration file matching the path specified in the `filename =` parameter.

Classic ZTP DHCP Setup Scripts for Cisco Prime Network Registrar (CPNR)

Following are two sets of scripts that allow you to add ZTP device, image and configuration file entries to the CPNR DHCP server configuration file. There's one set of three scripts for IPv4, and a separate set of five scripts for IPv6. To use these scripts:

1. Copy and paste the contents of the scripts into local text files with the names given here.
2. Modify the device, image, and configuration entries in the `ztp-v4-setup-vi-nrcmd.txt` script, or the `ztp-v6-setup-vi-nrcmd.txt` script, to fit your needs, as explained in the script comments.

3. Copy the script files you want to use to the root folder of your local CPNR server.
4. Execute the scripts on the CPNR server using the following command:

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

   Where:

   - *username* is the name of a user ID with administrator privileges on the CPNR server.
   - *password* is the password for the corresponding CPNR admin user ID.
   - *IPVersion* is either `v4` for the IPv4 version of the scripts, or `v6` for the IPv6 version of the scripts.

Note The following scripts are for use with Classic ZTP only. You can't use them with Secure ZTP.

Figure 16: IPv4 Script 1 of 3: ztp-v4-setup-vi-nrcmd.txt

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

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

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

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

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

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

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


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

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

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

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


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

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

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

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

Figure 17: IPv4 Script 2 of 3: ztp-v4-setup-vi-nrcmd.txt

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

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

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

# Set the lease time for clients on this scope
scope-policy ztp-ncs-5501-mgmt setoption dhcp-lease-time 216000
#
```

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

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

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


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

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

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

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


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

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

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

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

Figure 18: IPv4 Script 3 of 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"
)
```

Figure 19: IPv6 Script 1 of 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
```

Figure 20: IPv6 Script 2 of 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"
)
```

Figure 21: IPv6 Script 3 of 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")

  )
)
```

Figure 22: IPv6 Script 4 of 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")
  )
)
```

Figure 23: IPv6 Script 5 of 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 = , )
                }
              ] )
            }
          ] )
        }
      ] )
    }
```

Trigger ZTP Device Bootstrap

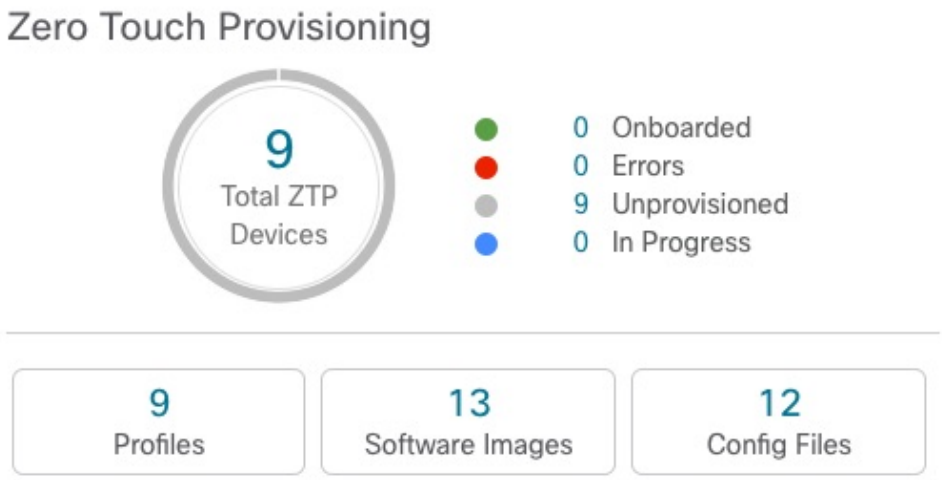
With device entries imported to Cisco Crosswork and DHCP configured, you can initiate ZTP processing by rebooting each of the devices.

Step 1 Initiate ZTP processing using one of the following options:

- Power-cycle the device to restart it.
- Using a pin, press the chassis reset button at the back of the device. Press for 15 seconds, or until the power light on the device starts flashing.
- For a previously imaged device: Connect to it via Telnet, then issue a **ztp initiate** command.

Repeat this step as needed for each of the devices you plan to provision during this session. You need not restart all the devices you have uploaded as device entries in a single session.

Step 2 Monitor ZTP progress using the Zero Touch Provisioning status tile shown in the following figure. To view the tile, click the **Home** icon on the main menu.



The tile provides a summary view of your current ZTP processing status. It gives a count of all the ZTP profiles, images, and configuration files currently in use. The tile also shows the number of devices in each of the possible ZTP processing states.

Complete Onboarded ZTP Device Information

ZTP devices, once onboarded, are automatically part of the shared Cisco Crosswork device inventory. You can edit them like any other device. The following steps explain two ways to add information to devices onboarded using ZTP.

Before editing any device, it's always good practice to export a CSV backup of the devices you want to change. You can do this using the export function described in Step 2.

Before you begin

Some information needed for a complete device inventory record is either not necessary or not available via automation. For example: Geographical data, indicating that a device is located in a building at a given address, or at a set of GPS coordinates. Location data like this is a requirement for most organizations with active networks, and can only be added by a human operator.

Still other kinds of inventory information are useful when you use other applications to manage your network. For example: Cisco Crosswork tags make it easier to apply Cisco Crosswork Health Insights KPIs to particular devices. Similarly, associating an SRE policy with devices makes it easier to use Cisco Crosswork Network Controller or Cisco Crosswork Optimization Engine. Some Cisco Crosswork providers, such as Cisco NSO, base convenient functions on this kind of extended device information. All of it needs update from humans.

You can add this kind of information using functions in the other Cisco Crosswork applications and providers. For more information on this topic, see the user documentation for the application. You can also add much of it using Cisco Crosswork ZTP.

Step 1 To update the inventory record for a ZTP device:

a) From the main menu, choose **Device Management** > **Network Devices**.

b) Click the **ZTP Devices** tab.
c) Select the device you want to change, then click the  .
d) Change the value of the **Status** field to **`Unprovisioned`**.
e) Edit the other values configured for the device, as needed.
f) Click **Save**.

Step 2 To update the inventory records for devices in bulk, including devices onboarded using ZTP:

a) From the main menu, choose **Device Management** > **Devices**.
b) Click  . Save the CSV file.
c) Open the CSV template with the application of your choice and edit the device information you want to add or update. It's a good idea to delete rows for devices you don't want to update.
d) When you're finished, save the edited CSV file.
e) If needed: Choose **Device Management** > **Devices**, then click the **Zero Touch Devices** tab.
f) Click  .
g) Click **Browse** to navigate to the CSV file you created and then select it.
h) With the CSV file selected, click **Import**.

Reconfigure Onboarded ZTP Devices

The purpose of Cisco Crosswork ZTP is to onboard new devices quickly and easily, without requiring you to locate experts on site with the new devices. ZTP performs imaging and configuration as part of that task, and can run scripts as part of device configuration. But it's not designed as an all-purpose device configuration utility, and shouldn't be used in that way.

If you need to reconfigure a device onboarded using ZTP, use:

- A Cisco Crosswork Change Automation Playbook, which allows you to roll out configuration changes to devices on demand.
- The configuration change functions of Cisco Network Services Orchestrator (Cisco NSO), or any of the other Cisco Crosswork providers you're using.
- A direct connection to the device and the device OS command line interface.

If you can't use any of these methods, the best approach is to delete the device. You can onboard the device again, this time with the correct configuration.

To delete a ZTP device, select **Device Management** > **Devices** > **Zero Touch Devices**, select the device in the table, then click  .

Retire or Replace Devices Onboarded With ZTP

Sometimes you must retire a Cisco device that was onboarded using ZTP. Device licenses are associated with the device serial number that you entered at the time of onboarding. ZTP permits association of a single device with up to three different serial numbers. You can use this fact to remove a failed or obsolete device from your network and from Cisco Crosswork inventory. You can replace it later without consuming an extra license.

This rule applies not only to devices with a chassis, but also to line cards and other pluggable device modules. Each of these modules has its own serial number. If you need to RMA a module, associate the old license with the serial number of the new module. But first remove the old line card and its serial number from inventory, as explained in the following steps.

1. Select **Device Management** > **Devices** > **Zero Touch Devices**.
2. Find the old device in the table and make a record of its serial number.
3. Select the device and then click the to delete it.

   After you delete the device, Cisco Crosswork will still count the license associated with this serial number as consumed. Track this license as part of any new or RMA replacement device purchase, so you can return the license for the old device to active use.

   Cisco Crosswork won't allow two active devices with the same license. You must delete the old device before you can onboard a new or replacement device.
4. When it's time to onboard the new device:
   a. When you create a ZTP device entry for the new device, enter both the new and old serial numbers.
   b. If you're using Secure ZTP: Submit both the old and new device serial numbers with the Ownership Voucher request for the new device. Cisco associates the old and new serial numbers with the in-use license in the regenerated Ownership Voucher.
   c. Onboard the new device as you would any other ZTP device. Only the old device license is consumed.

ZTP Asset Housekeeping

Once you have completed onboarding your devices with ZTP, you can delete offline copies of some of the ZTP assets you assembled. Retain others, depending on the policies and best practices of your organization. We recommend:

- **ZTP profiles**: Usually, it's safe to delete ZTP profiles after onboarding is complete. To delete a ZTP profile, select **Device Management** > **Zero Touch Profiles** . On the tile representing the ZTP profile you want to delete, click the ••• and then select **Delete** from the dropdown menu.
- **ZTP device entry CSV file**: You may want to retain an offline copy of this file for use as a template. This file can be handy if, say, you have many branch offices sharing the same network architecture and device types. Otherwise, you can simply delete it from the file system. You can download the CSV file template at any time. You may find it more useful to export a backup CSV file containing all the data for your ZTP devices, including data you entered after onboarding. To export a CSV device backup, select **Device Management** > **Devices** > **Zero Touch Devices** . Then click the and save the CSV file.
- **Software images and SMUs**: Save the production versions of these files offline, and delete older ones per the policies of your organization. Don't delete the uploaded image files from Cisco Crosswork if you plan to use them to image more devices of the same family. To delete obsolete images, select **Device Management** > **Software Images**, select the file in the table, then click the .
- **Configuration files**: You need not retain configurations you already uploaded to Cisco Crosswork, but the policy of your organization may differ. Don't delete uploaded configuration files if you plan to configure more devices of the same family using ZTP. When configurations change, you can easily

update the stored version. Prepare the new configuration file or script, select **Device Management** > **Configuration Files**, select the file in the table, and then click the . You can then browse to the new script file you created, and copy/paste the new configuration. If a configuration becomes obsolete, delete it: Select **Device Management** > **Configuration Files**, select the file in the table, then click the .

- **Credential profiles**: You can delete an imported credential profile CSV file immediately. Don't delete the uploaded credential profiles. When user names and passwords change, update the credential profiles: Select **Device Management** > **Credentials**, select the credential profile in the table, then click the .

Troubleshoot ZTP Issues

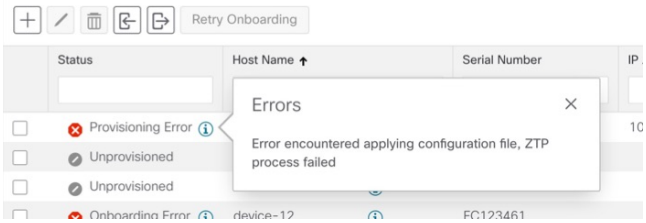
Cisco Crosswork ZTP provisioning and onboarding happen quickly and automatically, but errors and problems do occur. The following topics discuss how to remedy common problems.

Third-party devices that conform 100 percent to the Secure ZTP RFC are the only third-party devices you can onboard using Cisco Crosswork ZTP.

Inspect Status Errors

The **Status** column in the Zero Touch Devices window displays the next to every device entry whose ZTP processing finished with a **`Provisioning Error`**, **`Onboarding Error`** or (for Secure ZTP only) **`ZTP Error`**. Click on the icon to display a popup window with information about the error, as in the following example. When you're finished viewing the popup window, click to close it.

Figure 24: Provisioning Error Popup Window



Errors while uploading image files

Make sure that the MD5 checksum for the file is correct. If the file information is correct, image uploads can still fail due to slow network connections. If you're running into this problem, retry the upload.

Uploaded images and configuration files aren't in the drop-down menu when creating ZTP device entries or ZTP profiles

The drop-down menu selects images and configuration files based on the device family and release number you specify in your device entry or ZTP profile. Make sure that the file information matches the information for the device entry or profile you're using.

Errors during import of devices

If devices in inventory have the same serial numbers as the devices you're importing, check that the devices are in the **`Unprovisioned`** state before import. All the devices imported using CSV files have their status set to **`Unprovisioned`** on import. Before import, make sure the configurations, images, and ZTP profiles

mentioned in the CSV file exist. You can edit device image and configuration files by exporting a device CSV file and reimporting it with changes. If you use this edit method, make sure the CSV file has the correct UUIDs before import.

Image file download fails

Check that there's network connectivity between Cisco Crosswork and the device. Make sure that the device is getting its IP address from the DHCP server. Ensure that the UUID of the software image given in the configuration file of the DHCP server is correct. If you must correct the image UUID specified in the configuration file, make sure you restart the DHCP server before initiating ZTP processing again.

Configuration file download fails

Check that there's network connectivity between Cisco Crosswork and the device. Make sure that the device is getting its IP address from the DHCP server. Ensure that the UUID of the software image given in the DHCP server configuration file is correct. If you must correct the image UUID specified in the DHCP configuration file, make sure you restart the DHCP server before reinitiating ZTP processing. Make sure that the device serial number matches the serial number on the chassis of the device. Ensure that the status of the device is either **`Unprovisioned`** or **`In Progress`** before initiating ZTP processing. Configuration downloads continue to fail as long as the device is in any other state.

Device state is showing Onboarded and not Provisioned

`Provisioned` is an intermediate state in ZTP processing. When the device state changes to **`Provisioned`**, Cisco Crosswork attempts to onboard the device immediately. The status changes to **`Onboarded`** or **`Onboarding Error`** after.

Onboarding Error

The default Cisco Crosswork device life-cycle management (DLM) policy for identifying devices uniquely is the IP address. If you import a new device with an IP address that matches an existing device, the device status changes to **`Provisioned`**, then to **`Onboarding Error`**. If the IP address of the new device is blank, you get the same result. These same issues apply if your installation uses an OSPF ID, ISIS ID, or other DLM policy for determining device IDs. Onboarding can only succeed when you fill all the DLM policy fields with unique, nonblank values. If onboarding fails, inspect the popup error message, update the corresponding fields and retry onboarding.