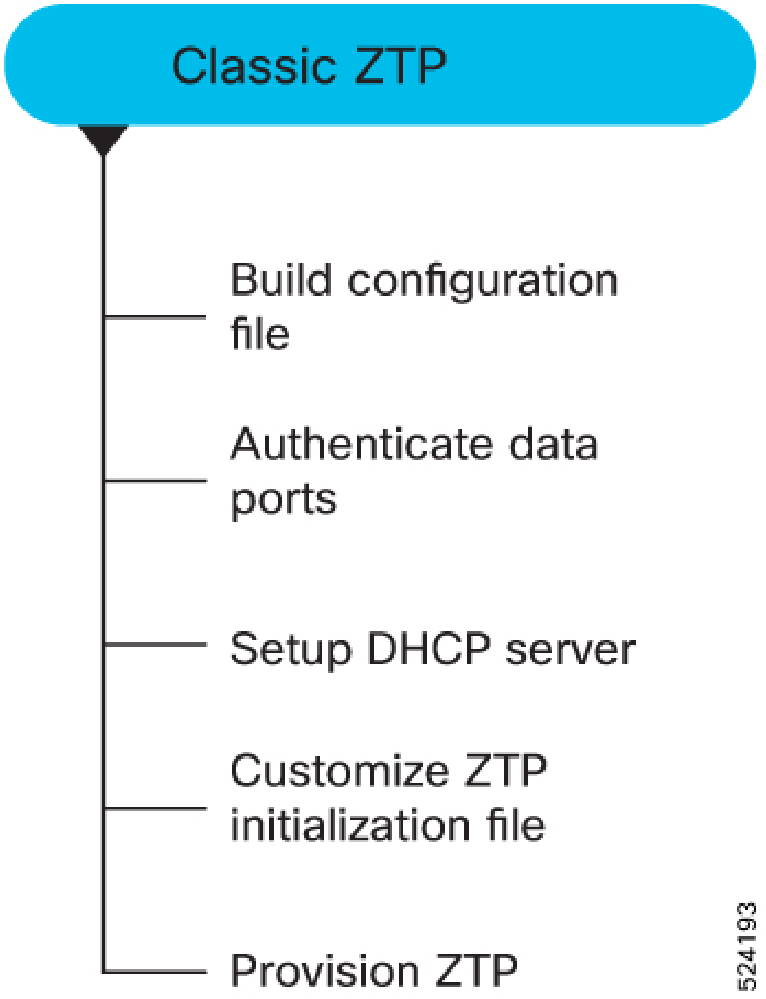
Deploy NCS 1010 Using Classic ZTP

Figure 1: Classic ZTP Work Flow



- Fresh Boot Using DHCP, on page 2
- Build your Configuration File, on page 3
- Configure ZTP BootScript, on page 4

Fresh Boot Using DHCP

The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration.

This image depicts the high-level work flow of the ZTP process:

1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option.

   - DHCP(v4/v6) client-id=Serial Number
   - DHCPv4 option 124: Vendor, Platform, Serial-Number
   - DHCPv6 option 16: Vendor, Platform, Serial-Number

   The following is the default sequential flow of the ZTP process:

   - ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
   - ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

     The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options:

   DHCP server should be configured to respond with the DHCP options.

   - DHCPv4 using BOOTP filename to supply script/config location.
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

3. The network device downloads the file from the web server using the URL location that is provided in the DHCP response.

4. The device receives a configuration file or script file from the HTTP server.

Note

- If the downloaded file content starts with !! IOS XR it is considered as a configuration file.
- If the downloaded file content starts with #! /bin/bash, #! /bin/sh or #!/usr/bin/python it is considered as a script file.

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

Build your Configuration File

Based on the business need, you can use a configuration or script file to initiate the ZTP process.

The configuration file content starts with !! IOS XR.

The following is the sample configuration file. You can automate all the configurations. For more information on creating ZTP configuration file, refer ZTP Configuration Files Creation.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 192.0.2.254I
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 192.0.2.255 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
Cisco NCS 1010 System Setup and Software Installation Guide, IOS XR Release 7.7.x
19
Bring-up Cisco NCS 1010
Build your Configuration File
!
telnet vrf default ipv4 server max-servers 100a
ssh server v2
ssh server netconf vrf default
```

```
netconf-yang agent
ssh
!
netconf agent tty
grpc
ncs1010 static
address-family ipv4 unicast
0.0.0.0/0 192.0.2.255
end
```

Configure ZTP BootScript

ZTP downloads and executes the script files. These script files include a programmatic approach to complete a task. For example, scripts created using IOS XR commands to perform patch upgrades. The first line of the file must contain #! /bin/bash or #! /bin/sh for ZTP to process the file as script. You can either use the ZTP bash script or the ZTP configuration file.

You can either use the ZTP bash script or the ZTP configuration file.

If you want to hardcode a script to be executed every boot, configure the following.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

The above configuration waits for the first data-plane interface to be configured and then wait an extra minute for the management interface to be configured with an IP address, to ensure that we have connectivity in the third-party namespace for applications to use. If the delay is not desired, use:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript preip /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

Note When the above command is first configured, you will be prompted if you wish to invoke it now. The prompt helps with testing.

This is the example content of **/disk0:/myscript**:

```
host ncs1010_P1B_DT_08_ETH0 {
#hardware ethernet 68:9e:0b:b8:6f:5c ;
option dhcp-client-identifier "FCB2437B05N" ;
if exists user-class and option user-class = "iPXE" {
filename "http://192.0.2.1/P1B_DT_08/ncs1010-x64.iso";
} else {
filename "http://192.0.2.1/P1B_DT_08/startup.cfg";
}
fixed-address 203.0.113.254;
}
```

The following is the sample content of the ZTP bash script.

```
#! /bin/bash
#
# NCS1010 Demo Sample
# ZTP installation of config and day-0 SMU's
#
```

```
source ztp_helper

wget http://downloads.sourceforge.net/project/yourcode/application.tgz
#install the downloaded application.tgz

#Run XR CLI’s from the script
`xrcmd “show version”`
```

The following is the sample content of the ZTP configuration file.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 203.0.113.254
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 192.0.2.255 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/1
description noshut-interface-ztp
no shut
end
```

Invoke ZTP Manually through CLI

Manual ZTP can be invoked through CLI commands. This manual way helps you to provision the NCS 1010 in stages. Ideal for testing out ZTP configuration without a reboot. If you want to invoke a ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first. You can execute the ztp initiate command, even if the interface is down, ZTP script brings it up and invoke dhclient. So ZTP could run over all interfaces no matter it is up or down.

Use the ztp initiate, ztp terminate, and ztp clean commands to force ZTP to run over more interfaces.

- ztp initiate—Invokes a new ZTP DHCP session. Logs can be found in /disk0:/ztp/ztp.log.
- ztp terminate—Terminates any ZTP session in progress.
- ztp clean—Removes only the ZTP state files.

The log file ztp.log is saved in /var/log/ztp.log folder, and a copy of log file is available at /disk0:/ztp/ztp.log location using a soft link. However, executing ztp clean clears files saved on disk and not on /var/logztp.log folder where current ZTP logs are saved. In order to have a log from current ZTP run, you must manually clear the ZTP log file from /var/log/ztp.log folder.

Procedure

Step 1 (optional) **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 2 **ztp initiate**

Example:

```
RP/0/RP0/CPU0:ios#ztp initiate
Fri Jun 17 11:44:08.791 UTC
Initiating ZTP may change your configuration.
Interfaces might be brought up if they are in shutdown state
Would you like to proceed? [no]: yes
ZTP will now run in the background.
Please use "show logging" or look at /var/log/ztp.log to check progress.
RP/0/RP0/CPU0:ios#
```

Use the **show logging** command or see the /var/log/ztp.log to check progress.

Reboots the Cisco NCS 1010 system.

Step 3 (Optional) **ztp terminate**

Example:

```
RP/0/RP0/CPU0:ios#ztp terminate
Fri Apr 29 06:38:59.238 UTC
This would terminate active ZTP session if any (this may leave your system in a partially configured
 state)
Would you like to proceed? [no]: yes
Terminating ZTP
No ZTP process running
```

Terminates the ZTP process.

Invoke ZTP Through Reload

The ZTP process can be automatically invoked by using the reload command.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:P2B_DT_02#configure
```

Enters the configuration mode.

Step 2 **commit replace**

Warning
This operation erases the complete database of the NCS1010 and impacts the traffic.

Example:

```
Fri Apr 29 06:48:46.236 UTC
RP/0/RP0/CPU0:ios(config)#commit replace
Fri Apr 29 06:48:53.199 UTC

This commit will replace or remove the entire running configuration. This
operation can be service affecting.
Do you wish to proceed? [no]: yes
RP/0/RP0/CPU0:ios(config)#end
```

Removes the entire running configuration.

Step 3 **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 4 **reload**

Example:

```
RP/0/RP0/CPU0:ios#reload
Fri Apr 29 06:50:12.312 UTC
Proceed with reload? [confirm]

RP/0/RP0/CPU0:ios#
Preparing system for backup. This may take a few minutes especially for large configurations.
 Status report: node0_RP0_CPU0: BACKUP INPROGRESS
 Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]
```

After the node comes up, you can check that the ZTP is initiated and the configuration has been restored successfully.

```
RP/0/RP0/CPU0:Apr 29 06:55:33.242 UTC: pyztp2[377]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has initiated
 config load and commit operations
RP/0/RP0/CPU0:Apr 29 06:55:39.263 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Down
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Up
RP/0/RP0/CPU0:Apr 29 06:55:39.716 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.728 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:47.904 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Ots0/0/0/1:

User Access Verification

Username: cisco
Password:
ios con0/RP0/CPU0 is now available
```

Reboots the Cisco NCS 1010 system.

Authenticate Data Ports

On fresh boot, ZTP process is initiated from management ports and may switch to data ports. To validate the connection with DHCP server, authentication is performed on data ports through DHCP option 43 for IPv4 and option 17 for IPv6. These DHCP options are defined in option space and are included within **dhcpd.conf** and **dhcpd6.conf** configuration files. You must provide following parameters for authentication while defining option space:

- Authentication code—The authentication code is either 0 or 1; where 0 indicates that authentication is not required, and 1 indicates that MD5 checksum is required.

Note If the option 43 for IPv4, and option 17 for IPv6 is disabled, the authentication fails.

- Client identifier—The client identifier must be 'exr-config'.
- MD5 checksum—This is chassis serial number. It can be obtained using **echo -n $SERIALNUMBER | md5sum | awk '{print $1}'** .

Here is the sample **dhcpd.conf** configuration. In the example below, the option space called **VendorInfo** is defined with three parameters for authentication:

```
class "vendor-classes" {
   match option vendor-class-identifier;
}

option space VendorInfo;
option VendorInfo.clientId code 1 = string;
option VendorInfo.authCode code 2 = unsigned integer 8;
option VendorInfo.md5sum code 3 = string
option vendor-specific code 43 = encapsulate VendorInfo;
subnet 10.65.2.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option ncs 1010 10.65.2.1;
  range 10.65.2.1 10.65.2.200;
}
host cisco-mgmt {
   hardware ethernet 00:50:60:45:67:01;
   fixed-address 10.65.2.39;
   vendor-option-space VendorInfo;
   option VendorInfo.clientId "exr-config";
   option VendorInfo.authCode 1;
   option VendorInfo.md5sum "aedf5c457c36390c664f5942ac1ae3829";
   option bootfile-name "http://10.65.2.1:8800/admin-cmd.sh";
}
```

Here is the sample **dhcpd6.conf** configuration file. In the example below, the option space called **VendorInfo** is defined that has code width 2 and length width 2 (as per dhcp standard for IPv6) with three parameters for authentication:

```
log-facility local7;
option dhcp6.name-servers 2001:1451:c632:1::1;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/lib/dhcpd/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = string;
option space CISCO-EXR-CONFIG code width 2 length width 2;
option CISCO-EXR-CONFIG.client-identifier code 1 = string;
option CISCO-EXR-CONFIG.authCode code 2 = integer 8;
option CISCO-EXR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-EXR-CONFIG code 9 = encapsulate CISCO-EXR-CONFIG;
subnet6 2001:1451:c632:1::/64{
 range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
 option CISCO-EXR-CONFIG.client-identifier "exr-config";
 option CISCO-EXR-CONFIG.authCode 1;
 #valid md5
 option CISCO-EXR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f0";
 if option dhcp6.user-class = 00:04:69:50:58:45 {
  option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/image.iso";
 }
 else {
   #option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";
   option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
  }
}
```

Setup DHCP Server

For ZTP to operate a valid IPv4 or IPv6 address is required and the DHCP server must send a pointer to the configuration script.

The DHCP request from the NCS 1010 has the following DHCP options to identify itself:

- **Option 60**: “vendor-class-identifier” : Used to Identify the following four elements:
    - The type of client: For example, PXEClient

  - The architecture of The system (Arch): For example: 00009 Identify an EFI system using a x86-64 CPU
  - The Universal Network Driver Interface (UNDI):

    For example 003010 (first 3 octets identify the major version and last 3 octets identify the minor version)
  - The Product Identifier (PID):

- **Option 61**: “dhcp-client-identifier” : Used to identify the Serial Number of the device.
- **Option 66** : Used to request the TFTP server name.
- **Option 67**: Used request the TFTP filename.
- **Option 97**: “uuid” : Used to identify the Universally Unique Identifier a 128-bit value (not usable at this time)

Example

The following DHCP request sample provides a fixed IP address and a configuration file with the mac address of the management interface.

```
host cisco-rp0 {
   hardware ethernet e4:c7:22:be:10:ba;
   fixed-address 172.30.12.54;
   filename "http://172.30.0.22/configs/cisco-1.config";
}
```

The following DHCP request sample provides a fixed IP address and a configuration file with the mac address of the management interface along with capability to re-image the system using iPXE (exr-config "xr-config" option):

```
host cisco-rp0 {
   hardware ethernet e4:c7:22:be:10:ba;
   fixed-address 172.30.12.54;
   if exists user-class and option user-class = "iPXE" {
      filename = "http://172.30.0.22/boot.ipxe";
   } elsif exists user-class and option user-class = "exr-config"  {
      filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
   }
}
```

DHCP server identifies the device and responds with either an IOS-XR configuration file or a ZTP script as the filename option.

The DHCP server responds with the following DHCP options:

- DHCPv4 using BOOTP filename to supply script/config location.
- DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
- DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

The following sample shows the DHCP response with bootfile-name (option 67):

```
option space cisco-vendor-id-vendor-class code width 1 length width 1;
option vendor-class.cisco-vendor-id-vendor-class code 9 = {string};

######### Network 11.11.11.0/24 ################
```

```
shared-network 11-11-11-0 {

####### Pools ##############
  subnet 11.11.11.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
 option broadcast-address 192.0.2.255;
 option ncs 1010 198.51.100.254;
 option domain-name-servers 198.51.100.254;
 option domain-name "cisco.local";
 # DDNS statements
   ddns-domainname "cisco.local.";
 # use this domain name to update A RR (forward map)
   ddns-rev-domainname "in-addr.arpa.";
   # use this domain name to update PTR RR (reverse map)

    }

######## Matching Classes ##########

  class "cisco" {
       match if (substring(option dhcp-client-identifier,0,11) = "FGE194714QS");
  }


  pool {
      allow members of "cisco";
      range 203.0.113.1 203.0.113.4;
      next-server 198.51.100.254;

      if exists user-class and option user-class = "iPXE" {
         filename="http://198.51.100.254:9090/cisco-mini-x-6.2.25.10I.iso";
      }


      if exists user-class and option user-class = "exr-config"
       {
          if (substring(option vendor-class.cisco-vendor-id-vendor-class,19,99)="cisco")
          {
            option bootfile-name
"http://198.51.100.254:9090/scripts/exhaustive_ztp_script.py";
          }
        }


      ddns-hostname "cisco-local";
      option ncs 1010 198.51.100.254;
  }
}
```

Customize ZTP Initialization File

You can customize the following ZTP configurable options in the *ztp.ini* file:

- `ZTP`: You can enable or disable ZTP at boot using CLI or by editing the *ztp.ini* file.
- `Retry`: Set the ZTP DHCP retry mechanism: The available values are infinite and once.
- `Fetcher Priority`: Fetcher defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file. You can modify the default priority of the fetcher. Allowed range is from 0 to 9.

Note Lower the number higher the priority. The value 0 has the highest priority and 9 has the lowest priority.

By default, the USB port has the higher priority.

In the following example, the Mgmt4 port has the highest priority:

```
[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

- `progress_bar`: Enable progress bar on the console. By default, the progress bar is disabled. To enable the progress bar, add the following entry in the `ztp.ini` file.

```
[Options]
progress_bar: True
```

- `config_check`: Saves ZTP configuration hashes in the **`/disk0:/ztp/`** location on the NCS 1010. By default, the config check is disabled. To enable the config check, add the following entry in the`ztp.ini` file.

```
[Startup]
start:         True
retry_forever: True
config_check: True
```

You can view the ZTP hashes by using the **show ztp log** command as seen below:

```
RP/0/RP0/CPU0:ios# show ztp log

============================== /var/log/ztp.log ==============================

2023-03-14 12:51:29,251 53612 [Configuration] INF: Provisioning via config replace
2023-03-14 12:51:43,131 53612 [Configuration] INF: Configuration has been applied
2023-03-14 12:51:43,131 53612 [Env ] DEB: cfg::createRefOnConfigCommit: called
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg:: Generating hash for File name:
/disk0:/ztp/customer/config.inithash_tmp
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH :
c7980cfc23a401bbbf296e3d49c76bf9, type : 1
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg:: Generating hash for File name:
/disk0:/ztp/customer/config.successhash_tmp
..........
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH :
c7980cfc23a401bbbf296e3d49c76bf9, type : 2
2023-03-14 12:52:04,901 53612 [Env ] DEB: cfg::getRefOnSuccess :: called
..........
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:active, exit
code:success
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:final, exit
code:success: state changed to final
```

By default, the `ztp.ini` file is located in the `/pkg/etc/` location. To modify the ZTP configurable options, make a copy of the file in the `/disk0:/ztp/` directory and then edit the `ztp.ini` file.

To reset to the default options, delete the ztp.ini file in the `/disk0:/ztp/` directory.

Note Do not edit or delete the `ztp.ini` file in the `/pkg/etc/` location to avoid issues during installation.

The following example shows the sample of the `ztp.ini` file:

```
[Startup]
start: True
retry_forever: True

[Fetcher Priority]
USB: 0

Mgmt4:  1
Mgmt6:  2
DPort4: 3
DPort6: 4
```

Enable ZTP Using CLI

If you want to enable ZTP using CLI, use the **ztp enable** command.

Configuration example

```
RP/0/RP0/CPU0:ios#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

Disable ZTP Using CLI

If you want to disable ZTP using CLI, use the **ztp disable** command.

Configuration example

```
RP/0/RP0/CPU0:ios#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

Provision ZTP

When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration. During the process, the NCS 1010 receives the details of the configuration file from the DHCP server. The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration. Here is the high-level work flow of the ZTP process for the Fresh boot:

1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option

   - DHCP(v4/v6) client-id=Serial Number

   - DHCPv4 option 124: Vendor, Platform, Serial-Number

   - DHCPv6 option 16: Vendor, Platform, Serial-Number

The following is the default sequential flow of the ZTP process:

- ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
- ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

  The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options: DHCP server should be configured to respond with the DHCP options.
   - DHCPv4 using BOOTP filename to supply script/config location
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location
3. The network device downloads the file from the web server using the URI location that is provided in the DHCP response.
4. The device receives a configuration file or script file from the HTTP server.

Note

- If the downloaded file content starts with !! IOS XR it is considered as a configuration file.
- If the downloaded file content starts with #! /bin/bash, #! /bin/sh or #!/usr/bin/python it is considered as a script file.

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

ZTP Logging

ZTP logs its operation on the flash file system in the directory /disk0:/ztp/. ZTP logs all the transaction with the DHCP server and all the state transition.

The following example displays the execution of a simple configuration script downloaded from a management interface or a data interface using the command ztp initiate network interface Ten 0/0/0/0 verbose. This script unshuts all the interfaces of the system and configure a load interval of 30 seconds on all of them.

```
2022-06-17 11:52:34,682  19292 [Xr           ] INF: Downloading the file to /tmp/ztp.script
2022-06-17 11:52:35,329  19292 [Report       ] INF: User script downloaded successfully.
Provisioning in progress.
2022-06-17 11:52:35,330  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: Config device work for ZAdmin. done = False
2022-06-17 11:52:35,330  19292 [ZAdmin       ] DEB: Proceeding to provision the NCS 1010
2022-06-17 11:52:35,331  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,331  19292 [Engine      ] INF: ZAdmin, current state:active: state tag
 changed to provision
RP/0/RP0/CPU0:Jun 17 11:52:35.341 UTC: pyztp2[140]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has
```

```
 initiated config load and commit operations
2022-06-17 11:52:35,339  19292 [Env          ] DEB: No MTU configs detected
2022-06-17 11:52:35,340  19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,354  19292 [Xr           ] DEB: Will apply the following config:
/disk0:/ztp/customer/config.candidate
2022-06-17 11:52:35,354  19292 [Xr           ] INF: Applying user configurations
2022-06-17 11:52:35,355  19292 [Configuration] INF: Provisioning via config replace
2022-06-17 11:52:54,656  19292 [Configuration] INF: Configuration has been applied
2022-06-17 11:52:54,656  19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Sending standby sync message. done = False
2022-06-17 11:52:54,663  19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: [privileged] getting engine status. done = False
2022-06-17 11:52:54,664  19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Execute post-configuration script. done = False
2022-06-17 11:52:55,212  19292 [Env          ] INF: Env::cleanup, success:True, exiting:False
2022-06-17 11:52:55,213  19292 [ZtpHelpers   ] DEB: Executing: source /pkg/bin/ztp_helper.sh
 && echo -ne  | xrcmd "show running-config"
2022-06-17 11:52:55,825  19292 [Env          ] INF: Executing command ip netns exec
vrf-default /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 -r Mg0_RP0_CPU0_0 to
release IP
2022-06-17 11:52:56,968  19292 [Xr           ] INF: Removing linux route with ip 203.0.113.254
2022-06-17 11:52:57,023  19292 [Engine       ] INF: ZAdmin, current state:active, exit
code:success
2022-06-17 11:52:57,023  19292 [Engine       ] INF: ZAdmin, current state:final, exit
code:success: state changed to final
2022-06-17 11:52:59,737  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: Sending standby sync message. done = False
2022-06-17 11:52:59,738  19292 [Engine       ] WAR: ZAdmin, current state:final, exit
code:success: work is ignored: work=<desc='Sending standby sync message' done=False
priv=False>
2022-06-17 11:52:59,738  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] getting engine status. done = False
2022-06-17 11:53:04,744  19292 [main         ] DEB: Moved to final state
2022-06-17 11:53:04,745  19292 [main         ] DEB: ZTP completed successfully
2022-06-17 11:53:04,745  19292 [main         ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:04,746  19292 [main         ] DEB: Exiting. Will not retry now.
2022-06-17 11:53:04,746  19292 [main         ] DEB: Shutting down adaptor. Cleanup False. Exiting
 False
2022-06-17 11:53:04,748  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] prepare engine shutdown. done = False
2022-06-17 11:53:04,849  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] shutting down ZAdmin engine. done = False
2022-06-17 11:53:04,849  19292 [Engine       ] INF: ZAdmin, current state:final, exit
code:shutdown
2022-06-17 11:53:04,849  19292 [Engine       ] INF: ZAdmin, exit code:shutdown: state changed
 to None
2022-06-17 11:53:04,849  19292 [Engine       ] DEB: ZAdmin, exit code:shutdown: breaking
engine loop after shutdown
2022-06-17 11:53:04,850  19292 [Engine       ] DEB: ZAdmin, exit code:shutdown: end of event
 loop
2022-06-17 11:53:04,850  19292 [Adaptor      ] DEB: Adaptor : Cleanup for admin context on
 Terminate
2022-06-17 11:53:06,119  19292 [main         ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:06,119  19292 [main         ] INF: ZTP Exited
RP/0/RP0/CPU0:Jun 17 11:53:06.119 UTC: pyztp2[140]: %INFRA-ZTP-4-EXITED : ZTP exited
```

Generate Tech Support Information for ZTP

When you have a problem in the ztp process that you cannot resolve, the resource of last resort is your Cisco Systems technical support representative. To analyze a problem, your technical support representative needs certain information about the situation and the symptoms that you are experiencing. To speed up the problem isolation and resolution process, collect the necessary data before you contact your representative.

Use the **show tech-support ztp** command to collect all debugging information of ztp process.

Example:

```
RRP/0/RP0/CPU0:ios#show tech-support ztp
Thu Jul 28 08:33:27.531 UTC
++ Show tech start time: 2022-Jul-28.083327.UTC ++
Thu Jul 28 08:33:28 UTC 2022 Waiting for gathering to complete
..
Thu Jul 28 08:33:34 UTC 2022 Compressing show tech output
Show tech output available at 0/RP0/CPU0 :
/harddisk:/showtech/showtech-R1-ZTP-2022-Jul-28.083327.UTC.tgz
++ Show tech end time: 2022-Jul-28.083334.UTC ++
RP/0/RP0/CPU0:R1#
```

In the above example, the tech support information is saved as .tgz file in the specified location. This information can be shared with the Cisco Technical Support representatives for troubleshooting the ztp process.