



Serviceability and Zeroization Features for IoT

- [Serviceability and Zeroization Features for IoT, on page 1](#)

Serviceability and Zeroization Features for IoT

The following features are included in the Cisco IOS-XE release 16.11.1 for the Internet of Things (IoT) products.

Serviceability Features

Cisco IOS-XE **show tech-support** functionality is extensively used by technical support for various platforms that run IOS-XE and comprises of a library of shell scripts that spawn various show commands to obtain the state of the device for debugging purposes. The tech-support output is very critical in debugging various problems in the system and has been a key component in debug infrastructure.

The **show tech-support** series of commands has been a part of the Cisco IOS and IOS-XE release since release 4.0(0)N1(1a). The IoT products follow the core IOS-XE software functionality.

The output from the **show tech-support** command is very long. To better manage this output, you can redirect the output to a file (for example, **show tech-support > filename**) in the local writable storage file system or the remote file system.

You can use one of the following redirection methods:

> *filename* — Redirects the output to a file.

>> *filename* — Redirects the output to a file in append mode.

Examples

This example shows how to display technical support information:

```
device# show tech-support
```

This example shows how to redirect the technical support information to a file:

```
device# show tech-support > bootflash:TechSupport.txt
```

This example shows how to display the brief technical support information for the device:

```
device# show tech-support brief
```

This example shows how to display the technical support information for a specific feature:

```
device# show tech-support aaa
```

This example shows how to display the commands used to generate the technical support information:

```
device# show tech-support commands
```

For Cisco IOS-XE release 16.11.1, improvements were made to improve the monitoring capabilities of forwarding plane (QFP) using CLI and SNMP. **show platform resources** would display QFP details and an SNMP MIB walk would include all QFP objects including memory related MIB objects. **show inventory** and **show inventory oid** will display the related Forwarding processor and its OID information.

show tech-support is enhanced to include the following CLIs:

```
show platform hardware qfp active infrastructure punt config cause
show platform hardware qfp active infrastructure punt internal-interface
show platform hardware qfp active interface if-name internal0/0/rp:0
show platform hardware qfp active interface if-name internal0/0/recycle:0
show platform hardware qfp active interface if-name internal0/0/crypto:0
show platform hardware qfp active infrastructure uidb internal0/0/rp:0 input config
show platform hardware qfp active infrastructure uidb internal0/0/recycle:0 input config
show platform hardware qfp active infrastructure uidb internal0/0/crypto:0 input config
show platform hardware qfp active infrastructure uidb internal0/0/rp:0 output config
show platform hardware qfp active infrastructure uidb internal0/0/recycle:0 output config
show platform hardware qfp active infrastructure uidb internal0/0/crypto:0 output config
show platform hardware qfp active infrastructure punt statistics interface 1
show platform hardware qfp active infrastructure punt statistics interface 2
show platform hardware qfp active interface if-name internal0/0/rp:0 statistics
show platform hardware qfp active interface if-name internal0/0/recycle:0 statistics
show platform hardware qfp active interface if-name internal0/0/crypto:0 statistics
show platform hardware qfp active infrastructure punt statistics type per-cause
show platform hardware qfp active infrastructure punt statistics type global-drop
show platform hardware qfp active infrastructure punt statistics type punt-drop
show platform hardware qfp active infrastructure punt statistics type inject-drop
show platform hardware qfp active statistics drop
show platform hardware qfp active system state
show platform hardware qfp active system transactions
show platform hardware qfp active datapath infrastructure time basic
show platform hardware qfp active infrastructure exmem statistics
show platform hardware qfp active infrastructure exmem statistics user
show platform hardware qfp active infrastructure exmem resource
show platform hardware qfp active infrastructure exmem region
show platform hardware qfp active infrastructure exmem table
show platform hardware qfp active infrastructure bqs status
show platform hardware qfp active feature acl control
show platform hardware qfp active feature acl tree
show platform hardware qfp active feature tunnel state
show platform hardware qfp active feature erspan state
show platform hardware qfp active feature ess state
show platform hardware qfp active feature ipfrag global
show bootlog FP active
show bootlog RP active
show platform software diagnostic chassis-manager R0 cpld
show platform software diagnostic chassis-manager R0 status
show platform software ipc queue-based chassis-manager R0 connection
show platform software ipc stream-based ios RP active connection
show platform software ipc stream-based ios RP active manager
show platform software process environment ios rp active
```

```
show power
show license tech support
show license summary
```

New CLIs that have been added are:

```
show tech-support l2
show tech-support acl
show tech-support dhcp
show tech-support port-channel
show tech-support private-vlan
show tech-support vlan
show tech-support confidential
```

Detailed information on all of these commands can be found in the Catalyst 9500 Switches Command Reference:

https://www.cisco.com/.../switches/catalyst9500/switches/16-10-command-reference/1610_9500_cmds_and_hw_cmds_commands.html#wp384814936

Device Zeroization or Declassification

Zeroization consists of erasing any and all potentially sensitive information in the device. This function is also referred to as Declassification. This includes erasure of Main memory, cache memories, and other memories containing packet data, NVRAM, and Flash memory. The process of zeroization is launched upon the initiation of a user command and a subsequent trigger.

On the device, the Reset button is used exclusively for triggering the Zeroization/Declassification process which zeroize and erase device configuration files or entire flash file system depending on the option provided under "service declassify".

The zeroization process starts as soon as the reset button is pressed down. The CLI command, "service declassify", is used to set the desired action in response to reset button press. To prevent accidental erasure of the system configuration/image, the default setting is set to "no service declassify".

Command Line Interface

There are two levels of zeroization actions, erase-nvram and erase-all. The following CLI shows the options:

```
device(config)#service declassify ?
erase-nvram  Enable erasure of device configuration as declassification action. Default
is no erasure.
erase-all   Enable erasure of both flash and nvram file systems as part of
declassification. Default is no erasure
```

The "erase-nvram" level of declassification process searches for the following files, and erases the ones found.

- flash:/nvram_config
- flash:/vlan.dat

This also erases the complete NVRAM filesystem, therefore, all configurations, including startup and running configurations will get deleted.

The perma-locked bootable image(s) in the flash file system will still be available and can be used for booting the device.

The "erase-all" level of zeroization process erases the entire flash file system. This also wipes out all files and perma-locked bootable image(s). All interfaces are shut down before this process. Here, erasure of individual files in the flash file system is not possible and the only option is to erase the entire flash file system. This also erases packet data, ASIC data and processors related caches along with scrubbing Main memory.

With any level of zeroization, the device always fall back to the ROMMON prompt on the console after the erasure of configuration files or flash file system.

Zeroization Trigger

The user needs to press the button after configuring the level of erasure required by the above CLI commands. To make sure that the button press has been identified by underlying software, the user needs to press and hold it for ONE second, or at least till the zero LED starts blinking.

Zeroization Support in bootloader

The zeroization process may take several minutes, depending on several system parameters such as the size of DDR memory, EMMC disk size, etc.

It is possible that the zeroization may get interrupted by a power cycle before it completes. Since the primary OS image on EMMC itself gets purged during zeroization, it becomes impossible to continue zeroization after a power cycle. To solve this, zeroization support has been in the bootloader and will run it to completion even if it gets interrupted by power cycles.

The IOS-XE sets a flag in the PMU persistent register before relinquishing control to the bootloader through a reboot. The bootloader then sets an internal variable in QSPI flash so that it is persistent even across power cycles.