

# Reset and Device Zeroization

This section contains the following:


- Device Zeroization, on page 1
- Zeroization Trigger, on page 2
- Push Button, on page 3
- Intelligent Platform Management Interface, on page 4
- Microcontroller Unit (MCU), on page 5


## Device Zeroization

Zeroization consists of erasing any and all potentially sensitive information in the device. 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.

**Important** **service declassify erase-nvram** is NOT guaranteed to securely and completely erase the data from the underlying file system. The data may be recoverable by forensic analysis techniques. Consider using **service declassify erase-all** to securely delete all data on the device

**Important** IOS cannot securely erase an SD Card, so integrators that want secure erasure must not include the SD Card.

By default, the device will have the zeroization feature disabled. SPI: Flash, I2C, and ACT2 are not impacted by this feature.

**Note** Ensure that you are familiar with the Emergency Recovery Installation procedure BEFORE attempting to test the Zeroize feature.

When zeroization is functionally active, the SYS LED indicates blinking yellow until the device reloads.

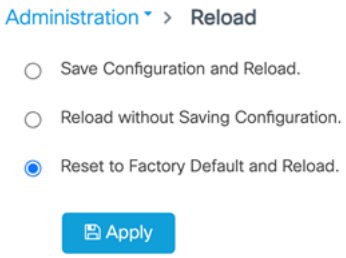
### Factory reset

When you want to remove an undesirable configuration that is present in the switch securely. If you have configured a switch and the configuration has become very complex, or if you want to move a switch from one network to other, you can reset the switch to factory defaults and configure it as a new switch. For more information on performing factory reset, refer factory reset.

### WARNING!

The CLI **service declassify erase-all** is a **software self-destruct mechanism** intended for defense and intelligence environments that attempts to wipe clean, all of the writable non-volatile storage on the device to clear the device configuration, other stored configurations and all security credentials including any additional license keys.

Please do not use this feature in lieu of doing a **write erase** from the CLI or from the Administration page, Reload option of the WebUI. Invoke the reload with the **Reset to Factory Default and Reload** option and click **Apply**. See the following figure.




If **service declassify erase-all** is invoked, after restoring the IOS-XE image and device configuration, you must re-license the device using the standard Cisco Smart Licensing procedures which ultimately require a Cisco Smart Account and access to the internet or a satellite license server.

# Zeroization Trigger

Zeroization can be triggered by software followed by the push button.

To trigger the zeroization from the push button, press and hold the button for 4+ seconds. When the system auto reloads, it will come up in ROMMON mode: "$$" with bootflash: wiped clean.

Zeroization clears:

- main memory,
- cache memory,
- NVRAM, and
- flash memory.

For both software and hardware zeroization, these sequence of commands must be entered.

```
Switch# config terminal
Switch(config)# service declassify {erase-nvram | erase-all}
/*erase-all: Enables theerase of both flash and nvram file systems.*/
/*erase-nvram: Enables the erase of nvram file system.*/
```

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 “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 router always fall back to the ROMMON prompt on the console after the erasure of configuration files or flash file system.

**Caution** The device does not support USB hot plug when it is in ROMMON mode.

To verify if service declassify is enabled or not use the **show declassify** command.

```
Switch# show declassify
 Declassify facility: Enabled=Yes  In Progress=No
                      Erase flash=Yes  Erase nvram=Yes
  Declassify Console and Aux Ports
  Shutdown Interfaces
  Reload system
```

To disable the declassification, use the **no service declassify** command:

```
Switch# config terminal
Switch(config)# no service declassify
```

# Push Button

The term Reset Button does not have the same meaning as with other devices. There is no actual button on the device, and the system integrator must configure their platform with a push button. Reset on a device does not cause the device to reboot, but initiates the configured level of Zeroization.

Zeroization can be triggered by the push button, or software-triggered by a privilege 15 user with console access. There is no remote access for security reasons. On triggering zeroization, the eMMC, NVRAM will be erased completely.

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

**Note** While Cisco IOS and Cisco IOS-XE use the command line text of "declassify" in the command line interface (CLI) to enable the zeroize feature, in no way does this represent any specific endorsement or acknowledgment of a Government approved flash erasure methodology. Device Zeroization Declassification procedures are unique to each Government organization. Cisco solely provides the technical detail of the erasure operation here, not the policy distinction or any specific recommendation per classification. Please refer to your respective Government Agency policies, procedures, and recommendations for the handling of sensitive data to see if this procedure meets with those requirements.

There is a zeroization function available on the device when the system integrator has configured their platform with a push button.

- When the system is running in IOS mode, pressing this push button for 4+ seconds will cause files erase in flash, and will reset to factory-default mode on boot up.
- The button must be pressed while the system is turned on at the same time.
- The push button must continue to be held for more than 4 seconds after the power is turned on.
- Config-reg setting is in NVRAM, not changed by the push button.
- Pressing the push button when in rommon mode has no effect.
- Pressing the push button when in IOS mode causes a syslog message to appear and triggers a reload.
- Pressing the push button for more than 4+ seconds after power up displays the following message when reset has been triggered:

```
System Bootstrap, Version 1.4(DEV) [vandvisw-vandvisw 113], DEVELOPMENT SOFTWARE
Copyright (c) 1994-2019 by cisco Systems, Inc.
Compiled at Mon Jun 3 10:56:19 2019 by vandvisw
ESS-9300-CON-K9 platform with 4194304 Kbytes of main memory
MCU Version - Bootloader: 8, App: 10
MCU is in application mode.
Reset button push detected
```

# Intelligent Platform Management Interface

Intelligent Platform Management Interface (IPMI) handles the software reset and power-cycle management for the Cisco ESS-9300-8X16T of the Curtiss-Wright VPX3-623 card using SHUTDOWN_REQUEST GPIO and SHUTDOWN_READY GPIO pins.

These series of actions takes place for a software system reload, or power push button of Cisco ESS-9300-8X16T of the Curtiss-Wright VPX3-623:

- The system asserts the SHUTDOWN_REQUEST pin, causing the device to shut down and enter ROMMON mode.
- ROMMON verifies SHUTDOWN_REQUEST. If asserted, ROMMON asserts SHUTDOWN_READY, signaling IPMI to deassert SHUTDOWN_REQUEST and the system falls back to the `switch:` prompt. This allows power down to proceed; if SHUTDOWN_REQUEST is not asserted, ROMMON polls ALT_BOOT instead.
- During image loading, ROMMON monitors both SHUTDOWN_REQUEST and ALT_BOOT, acting on whichever is asserted.

- Once IOS initializes, it takes over monitoring SHUTDOWN_REQUEST, while ALT_BOOT becomes inactive.



**Note** Shutting down the device using SHUTDOWN_REQUEST GPIO and SHUTDOWN_READY GPIO pins are applicable only for ESS-9300-8X16T of the Curtiss-Wright VPX3-623.

# Microcontroller Unit (MCU)

The MCU is part of the device hardware. It performs the following functions:

- Monitors the Push button status at power up
- Monitors the system hardware watchdog output
- Maintains Reset Reason register
- Controls the SYS LED

The MCU versions are displayed using show version. Details on MCU version and upgrade status are also stored in Flash: as boothelper.log. The MCU is automatically upgraded by the software.

```
Router#show ver | i MCU
MCU bootloader version: 8
MCU application version: 10
Router#cat flash:boothelper.log
Logging at Fri Nov 15 05:00:54 Universal 2019
boot loader upgrade enabled
Bootloader is up-to-date
Current MCU App version is 10
MCU firmware is up-to-date
```

In the event the MCU Application is corrupt, or does not match the Release Notes version, this has to be repaired. Steps to recover from this state: Reload router, hit Ctrl+C to break into rommon mode.

```
Rommon>set MCU_UPGRADE=IGNORE
 –
Ignore MCU firmware upgrade errors
.
Rommon>sync
Rommon>reset
Rommon>boot bootflash:<image>
```

Once the MCU successfully upgrades, you can disable/unset this IGNORE option in rommon. Details on other MCU setting rommon options follow: (there are no available IOS configuration options or linux shell mode troubleshooting measures)

```
set MCU_UPGRADE=SKIP
 –
Prevents MCU firmware upgrade from taking place
.
set MCU_UPGRADE=FORCE
 –
Forces MCU firmware upgrade to take place
.
unset MCU_UPGRADE
 –
```

```
Normal operation. Allows automatic upgrade
.
```