Chapter 2

Initial Configuration and Recovery

This chapter describes initial configuration and recovery tasks. Subsequent chapters provide details about features introduced in this chapter.

This chapter contains these sections:

2.1 Initial Switch Access

Arista Network switches provide two initial configuration methods:

- Zero Touch Provisioning configures the switch without user interaction (Section 2.1.1).
- Manual provisioning configures the switch through commands entered by a user through the CLI (Section 2.1.2).

2.1.1 Zero Touch Provisioning

Zero Touch Provisioning (ZTP) configures a switch without user intervention by downloading a startup configuration file (***startup-config***) or a boot script from a location specified by a DHCP server. Section 6.4.4 describes network tasks required to set up ZTP.

The switch enters ZTP mode when it boots if flash memory does not contain ***startup-config***. It remains in ZTP mode until a user cancels ZTP mode, or until the switch retrieves a ***startup-config*** or a boot script. After downloading a file through ZTP, the switch reboots again, using the retrieved file.

Security Considerations

The ZTP process cannot distinguish an approved DHCP server from a rogue DHCP server. For secure provisioning, you must ensure that only approved DHCP servers are able to communicate with the switch until after the ZTP process is complete. Arista also recommends validating the EOS image on your ZTP server by confirming that its MD5 checksum matches the MD5 checksum that can be found on the EOS download page of the Arista website. On a UNIX server, the **md5sum** command calculates this checksum:

```
% md5sum EOS.swi
3bac45b96bc820eb1d10c9ee33108a25  EOS.swi
```

To provision the switch through Zero Touch Provisioning:

Step 1 Mount the switch in its permanent location.

Step 2 Connect at least one management or Ethernet port to a network that can access the DHCP server and configuration file.

Step 3 Provide power to the switch.

ZTP provisioning progress can be monitored through the console port. Section 2.1.2.1 provides information for setting up the console port. Section 2.1.2.2 provides information for monitoring ZTP progress and cancelling ZTP mode.

2.1.2 Manual Provisioning

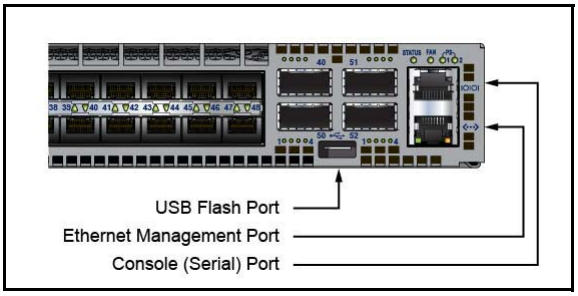
Initial manual switch provisioning requires the cancellation of ZTP mode, the assignment of an IP address to a network port, and the establishment of an IP route to a gateway. Initial provision is performed through the serial console and Ethernet management ports.

- The console port is used for serial access to the switch. These conditions may require serial access:
  - management ports are not assigned IP addresses
  - the network is inoperable
  - the enable password is not available
- The Ethernet management ports are used for out of band network management tasks. Before using a management port for the first time, an IP address must be assigned to that port.

2.1.2.1 Console Port

The console port is a serial port located on the front of the switch. Figure 2-1 shows the console port on the DCS-7050T-64 switch. Use a serial or RS-232 cable to connect to the console port. The accessory kit also includes an RJ-45 to DB-9 adapter cable for connecting the switch.

Figure 2-1 Switch Ports



Port Settings

Use these settings when connecting to the console port:

- 9600 baud
- no flow control
- 1 stop bit
- no parity bits
- 8 data bits

Admin Username

The initial configuration provides one username, **admin,** that is not assigned a password. When using the admin username without a password, you can only log into the switch through the console port. After a password is assigned to the **admin** username, it can log into the switch through any port.

The **username** command assigns a password to the specified username.

Example

- This command assigns the password **pxq123** to the **admin** username:

```
switch(config)#username admin secret pxq123
switch(config)#
```

New and altered passwords that are not saved to the startup configuration file, as described in Section 3.6.4: Saving the Running Configuration Settings, are lost when the switch is rebooted.

2.1.2.2 Cancelling Zero Touch Provisioning

Zero Touch Provisioning (ZTP) installs a ***startup-config*** file from a network location if flash memory does not contain a ***startup-config*** when the switch reboots. Cancelling ZTP is required if the switch cannot download a ***startup-config*** or boot script file.

When the switch boots without a ***startup-config*** file, it displays the following message through the console port:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

localhost login:
```

To cancel ZTP mode, log into the switch with the **admin** password, then enter the **zerotouch cancel** command. The switch immediately boots without installing a ***startup-config*** file.

```
localhost login: admin
admin
localhost>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request
on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9, Management1,
Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a valid DHCP
response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch Provisioning
from the beginning (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on  [
Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21, Ethernet22,
Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9, Management1, Management2 ]


localhost>zerotouch cancel
zerotouch cancel
localhost>Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-CANCEL: Cancelling Zero Touch
Provisioning
Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system
ø

Aboot 1.9.0-52504.EOS2.0
Press Control-C now to enter Aboot shell
```

Section 6.4.1 lists the remaining messages that the switch displays before providing a logon prompt. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file as described by step 8 of Section 2.1.2.3.

2.1.2.3 Ethernet Management Port

Arista switches provide one or more Ethernet management ports for configuring the switch and managing the network out of band. Figure 2-1 shows the location of the Ethernet management ports on a DCS-7050T-64 switch. Only one port is required to manage the switch.

You can access the Ethernet management port(s) remotely over a common network or locally through a directly connected PC. Before you can access the switch through a remote connection, an IP address and a static route to the default gateway are required. On a modular switch with dual supervisors, a virtual IP address can also be configured to access the management port on whichever supervisor is active.

Assigning a Virtual IP Address to Access the Active Ethernet Management Port

On modular switches with dual supervisors, this procedure assigns a virtual IP address which will connect to the Ethernet management port of the active supervisor. (To assign a physical IP address to an individual Ethernet management port, see Assigning an IP Address to a Specific Ethernet Management Port below.)

Step 1 Connect a PC or terminal server to the console port.

Use the settings listed in Section 2.1.2.1 under Port Settings.

Step 2 Type **admin** at the login prompt to log into the switch. Initial login through the console port does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console

switch>
```

Step 3 Type **enable** at the command prompt to enter Privileged EXEC mode. See Section 3.5.1: Mode Types for information about Privileged EXEC mode.

```
switch>enable
switch#
```

Step 4 Type **configure terminal** (or **config**) to enter global configuration mode. See Section 3.5.1: Mode Types for information about global configuration mode.

```
switch#configure terminal
switch(config)#
```

Step 5 Type **interface management 0** to enter interface configuration mode for the virtual interface which accesses management port 1 on the currently active supervisor.

```
switch(config)#interface management 0
switch(config-if-Ma0)#
```

Step 6 Type **ip address**, followed by the desired address, to assign a virtual IP address for access to the active management port.

This command assigns IP address 10.0.2.5 to management port 0.

```
switch(config-if-Ma0)#ip address 10.0.2.5/24
```

Step 7 Type **end** at both the interface configuration and global configuration prompts to return to Privileged EXEC mode.

```
switch(config-if-Ma0)#end
switch(config)#end
switch#
```

Step 8 Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file. See Section 3.6.4: Saving the Running Configuration Settings.

```
switch# write
switch#
```

Assigning an IP Address to a Specific Ethernet Management Port

This procedure assigns an IP address to a specific Ethernet management port:

Step 1 Connect a PC or terminal server to the console port.

Use the settings listed in Section 2.1.2.1 under Port Settings.

Step 2 Type **admin** at the login prompt to log into the switch. The initial login does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console

switch>
```

Step 3 Type **enable** at the command prompt to enter Privileged EXEC mode. See Section 3.5.1: Mode Types for information about Privileged EXEC mode.

```
switch>enable
switch#
```

Step 4 Type **configure terminal** (or **config**) to enter global configuration mode. See Section 3.5.1: Mode Types for information about global configuration mode.

```
switch#configure terminal
```

Step 5 Type **interface management 1** to enter interface configuration mode.

Any available management port can be used in place of management port 1.

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

Step 6 Type **ip address**, followed by the desired address, to assign an IP address to the port.

This command assigns the IP address 10.0.2.8 to management 1 port.

```
switch(config-if-Ma1)#ip address 10.0.2.8/24
```

Step 7 Type **end** at both the interface configuration and global configuration prompts to return to Privileged EXEC mode.

```
switch(config-if-Ma1)#end
switch(config)#end
```

Step 8 Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file. See Section 3.6.4: Saving the Running Configuration Settings.

```
switch# write
```

Configuring a Default Route to the Gateway

This procedure configures a default route to a gateway located at 10.0.2.1.

Step 1 Enter global configuration mode.

```
switch>enable
switch#configure terminal
```

Step 2 Create a static route to the gateway with the IP route command.

```
switch(config)#ip route 0.0.0.0/0 10.0.2.1
```

Step 3 Save the new configuration.

```
switch#write
switch#
```

2.2 Connection Management

The switch supports three connection methods:

- console
- SSH
- Telnet

The switch always enables console and SSH. Telnet is disabled by default.

Management commands place the switch in a configuration mode for changing session connection parameters.

Examples

- The **management console** command places the switch in console management mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- The **management ssh** command places the switch in SSH management mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- The **management telnet** command places the switch in Telnet management mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- The **exit** command returns the switch to global configuration mode.

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

The **idle-timeout** commands shown below configure the idle timeout period for the connection type being configured. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

Examples

- This **idle-timeout (SSH Management)** command configures an ssh idle-timeout period of three hours.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
```

- This **idle-timeout (Telnet Management)** command disables automatic connection timeout for telnet connections.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 0
```

The **shutdown (Telnet Management)** command enables and disables Telnet connections.

Examples

- These commands enable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
```

- These commands disable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#shutdown
```

2.3 Configure Session

The command **`configure session`** allows users to issue configuration sessions as CLIs that do not take effect immediately. Each `configure session` is saved with a unique name. A session is entered, modified and exited at any time by entering `configure session <name of session>` (e.g. `configure session routing_changes`) without impacting the currently running system configuration.

A session is defined as a collection of configuration changes that are grouped together.

When a session is committed, the configuration that was modified during the session is copied into the running configuration. A session can be aborted or removed, thereby removing the session completely and freeing up memory used by the session. The user must explicitly request that the changes in a deferred session be applied to the configuration of the router, entering a `commit` command and exiting the mode. Alternately, the user may abandon the changes, entering an `abort` command.

Configuration sessions are used to make sets of changes, after verifying there are no CLI errors. Configuration sessions allow the administrator to pre-provision a group of CLIs in a named session, thereby committing execution of each configuration session at specified times.

This chapter contains the following sections:

2.3.1 Configuration Session

The command **`configure session`** allows users to make a series of configuration changes in a temporary location and commit them to `running-config` at once by issuing the `commit` command.

- `configure session <name of session>` and `running-config` — The user enters a session (versus `configure terminal` in the case where configuration sessions are not used). If a session name is not specified, a system named session is created. A snapshot of the current `running-config` is copied into the session's data structure as the basis of further configuration changes.
- CLI configuration commands — User can run any configuration commands inside the session.
- `rollback clean-config` — User can run `rollback` command to revert the session's configuration to the default configuration (or clean configuration).
- `show session-config` — User can run `show session-config` to show the session's configuration, which will be the future `running-config` once committed.
- `commit` — User issues `commit` to commit the changes, which will replace the current `running-config`.
- `abort` — to abort the session and throw away all changes.
- `exit` — User can `exit` from the session, and later return to the same session by running `configure session <name>` again.
- For named session — More than one CLI instance can enter the same session and make changes to the session configuration. Once the session is committed in any of the CLIs, no other CLI can commit or make any other changes in that session.

2.3.2 Configure Replace

The command `configure replace <URL>` replaces the current `running-config` with the configuration saved in `<URL>`.

```
configure replace <URL> [ignore-errors]
```

By default, `configure replace <URL>` will replace `running-config` only if the configuration in `<URL>` loads without errors. The `ignore-errors` flag optionally forces the operation in spite of errors.

Note The command `copy <URL> running-config` was typically used to apply a saved configuration file to the system, and append that configuration to the current `running-config` (in lieu of replacing it). However, it is recommended the user uses the CLI command `configure replace <URL>` to streamline the process of deterministically restoring the system back to a known good configuration.

The normal workflow internally uses a configuration session to perform the replace.

2.3.3 Configuration CLI

In the CLI, execute the following configuration steps to create a configuration session.

Step 1 `configure session [<name of session>]`

Create or enter a session. If a name is not specified, it is automatically generated. The user is put in the session configuration mode and the prompt will change to show the first six characters of the session name. Designating the name of a session is optional. When `<name of session>` is not specified, a unique name is assigned.

`no configure session <name of session>`

Delete the specified configuration session. Designating the name of a session is required.

Step 2 `commit`

Commit the changes made in the session. This command must be issued from within the session configuration mode.

`abort`

Abort the session, which is the same as deleting it. This command must be issued from within the session configuration mode.

Step 3 `rollback clean-config`

Revert configuration in the session to the clean, factory-default configuration. This command must be issued from within the session configuration mode.

Step 4 `service configuration session max completed <num>`

Set a limit on the maximum number of committed sessions that are saved.

Step 5 `service configuration session max pending <num>`

Set a limit on the maximum number of uncommitted sessions that can be outstanding.

2.3.4 Show Commands

2.3.4.1 `show configuration sessions [detail]`

This command displays the following information about the sessions that exist in the system:

- The name of each session and its state (completed, pending, aborted, etc.) are displayed.
- If a user has currently entered the session, the user name and the associated terminal are also shown.
- With the detail flag, the process ID of the CLI process that is using the session is also displayed.

Note An asterisk (*) indicates that the user running the show command is currently in the marked session.

Example

```
Arista(config-s-s2)#show configuration sessions detail
Maximum number of completed sessions: 1
Maximum number of pending sessions: 5
Name State         User         Terminal       PID  Description
---- ------------ ------------ -------------- ---- ---------------------------
  s1 completed
* s2 pending       user123      vty870         7729
```

2.3.4.2 show session-config [diff]

This command must be issued from within a session. It shows the following:

- The session configuration, including the changes made in the session.
- The diff flag shows the differences with the `running-config`, which helps highlight the changes made in the session.

Example 1

```
Arista(config-s-s2)#show session-config
! Command: show session-configuration named s2
ip dhcp smart-relay global
!
transceiver qsfp default-mode 4x10G
!
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
ip host one 1.1.1.1
!
no aaa root
!
spanning-tree mode mstp
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
!
interface Ethernet4
!
interface Ethernet5
!
interface Ethernet6
!
no ip routing
!
!
end
```

Example 2

```
Arista(config-s-s2)#show session-config diff
--- system:/running-config
+++ session:/s2
@@ -5,6 +5,7 @@
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
+ip host one 1.1.1.1
!
no aaa root
!
```

2.3.4.3 `show session-config name <name of session>`

Show the session configuration of the named session.

Example

```
Arista#show session-config named s1
! Command: show session-configuration named s1
ip dhcp smart-relay global
!
transceiver qsfp default-mode 4x10G
!
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
!
no aaa root
!
spanning-tree mode mstp
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
!
interface Ethernet4
!
interface Ethernet5
!
interface Ethernet6
!
no ip routing
!
!
end
```

2.4 Recovery Procedures

These sections describe switch recovery procedures:

- Section 2.4.1: Removing the Enable Password from the Startup Configuration
- Section 2.4.2: Reverting the Switch to the Factory Default Startup Configuration
- Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration
- Section 2.4.4: Restoring the Configuration and Image from a USB Flash Drive

The first three procedures require Aboot Shell access through the console port. If the console port is not accessible, use the last procedure in the list to replace the configuration file through the USB Flash Drive.

Chapter 6, starting on page 353 describes the switch booting process and includes descriptions of the Aboot shell, Aboot boot loader, and required configuration files.

2.4.1 Removing the Enable Password from the Startup Configuration

The **enable password** controls access to Privileged EXEC mode. To prevent unauthorized disclosure, the switch stores the **enable password** as an encrypted string that it generates from the clear-text password. When the switch authentication mode is local and an **enable password** is configured, the CLI prompts the user to enter the clear-text password after the user types **enable** at the EXEC prompt.

The ***startup-config*** file stores the encrypted **enable password** to ensure that the switch loads it when rebooting. If the text version of the **enable password** is lost or forgotten, access to enable mode is restored by removing the encrypted **enable password** from the startup configuration file.

This procedure restores access to enable mode without changing any other configuration settings.

Step 1 Access the Aboot shell:

Step a Power cycle the switch by successively removing and restoring access to its power source.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is unknown, refer to Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including the startup configuration and EOS image.

Step 2 Change the active directory to /mnt/flash directory.

```
Aboot#cd /mnt/flash
```

Step 3 Open the startup-config file in vi.

```
Aboot#vi startup-config
```

Step 4 Remove the enable password line.

This is an example of an enable password line:

```
enable secret 5 $1$dBXo2KpF$Pd4XYLpI0ap1ZaU7glG1w/
```

Step 5 Save the changes and exit vi.

Step 6 Exit Aboot. This boots the switch.

```
Aboot#exit
```

Refer to Section 4.2.1.4: Enable Command Authorization for information on the **enable password**.

2.4.2 Reverting the Switch to the Factory Default Startup Configuration

The ***startup-config*** file contains configuration parameters that the switch uses during a boot. Parameters that do not appear in ***startup-config*** are set to their factory defaults when the switch reloads. The process requires the Aboot password if Aboot is password protected.

This procedure reverts EOS configuration settings to the default state through bypassing the ***startup-config*** file during a switch boot.

Step 1 Access the Aboot shell through the console port:

Step a Type **reload** at the Privileged EXEC prompt.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is unknown, refer to Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including ***startup-config*** and EOS image.

Step 2 Change the active directory to /**mnt/flash** directory.

```
Aboot#cd /mnt/flash
```

Step 3 Rename the startup configuration file.

```
Aboot#mv startup-config startup-config.old
```

Step 4 Exit Aboot. This boots the switch

```
Aboot#exit
```

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

Step 6 Configure the **admin** and **enable** passwords.

Refer to Section 4.2.1: Local Security File for information about creating usernames and passwords.

```
switch>enable
switch#configure terminal
switch(config)#enable secret xyz1
switch(config)#username admin secret abc41
```

Step 7 Save the new ***running-config*** to the startup configuration file.

```
switch#write
```

Step 8 (Optional) Delete the old startup configuration file.

```
switch#delete startup-config.old
```

After ZTP is cancelled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.4.3 Restoring the Factory Default EOS Image and Startup Configuration

A **fullrecover** command removes all internal flash contents (including configuration files, EOS image files, and user files), then restores the factory default EOS image and ***startup-config***. A subsequent installation of the current EOS image may be required if the default image is outdated. This process requires Aboot shell access through the console port.

This procedure restores the factory default EOS image and startup configuration.

Step 1 Access the Aboot shell through the console port:

Step a Type **reload** at the Privileged EXEC prompt.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is not known, enter an empty password three times, after which the CLI displays:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot:
```

Type **fullrecover** and go to step 4.

Step 2 Type **fullrecover** at the Aboot prompt.

```
Aboot#fullrecover
```

Aboot displays this warning:

```
All data on /mnt/flash will be erased; type "yes" and press Enter to proceed,
or just press Enter to cancel:
```

Step 3 Type **yes** and press **Enter**.

The switch performs these actions:

- erases the contents of /mnt/flash
- writes new boot-config, startup-config, and EOS.swi files to /mnt/flash
- returns to the Aboot prompt

Step 4 Exit Aboot. This boots the switch.

```
Aboot#exit
```

The serial console settings are restored to their default values (9600/N/8/1/N).

Step 5 Reconfigure the console port if non-default settings are required.

Step 6 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is cancelled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.4.4 Restoring the Configuration and Image from a USB Flash Drive

The USB flash drive port can be used to restore an original configuration when you cannot establish a connection to the console port. This process removes the contents of the internal flash drive, restores the factory default configuration, and installs a new EOS image from the USB flash drive.

This procedure restores the factory default configuration and installs an EOS image stored on a USB flash drive.

Step 1 Prepare the USB flash drive:

Step a Verify the drive is formatted with MS-DOS or FAT file system.

Most USB drives are pre-formatted with a compatible file system.

Step b Create a text file named **fullrecover** on the USB flash drive.

The filename does not have an extension. The file may be empty.

Step c Create a text file named **boot-config**.

The last modified timestamp of the **boot-config** file on the USB flash must differ from the timestamp of the **boot-config** file on the switch.

Step d Enter this line in the new **boot-config** file on the USB flash:

```
SWI=flash:EOS.swi
```

Step e Copy an EOS image file to the flash drive. Rename it **EOS.swi** if it has a different file name.

For best results, the flash drive should contain only these three files, because the procedure copies all files and directories on the USB flash drive to the switch.

- fullrecover
- boot-config
- EOS.swi

Step 2 Insert the USB flash drive into the USB flash port on the switch, as shown in Figure 2-1.

Step 3 Connect a terminal to the console port and configure it with the default terminal settings (9600/N/8/1) to monitor progress messages on the console.

Step 4 Power up or **reload** the switch.

The switch erases internal flash contents and copies the files from the USB flash drive to internal flash. The switch then boots automatically.

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is cancelled, the switch reboots using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.5 Session Management Commands

Global Configuration Commands

Management Configuration Commands

Display Commands

domain (XMPP Management)

The domain command configures the switch's XMPP domain name. Only messages using a domain matching the locally configured one are accepted by the XMPP client. The switch's domain name is used if none is specified.

Management over XMPP is disabled by default. To enable it, you must provide the location of the server along with the domain, username and password for the switch.

Arista recommends configuring the XMPP domain before the username, because it will provide shortcuts for the **switch-group** and **username** so they can be configured without the domain attached to it (eg. USERNAME instead of USERNAME@DOMAIN).

The **no domain** and **default domain** commands delete the domain name by removing the **domain** command from ***running-config***.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
domain string
no domain
default domain
```

Parameters

- *string* domain name (text string)

Example

- This command configures *test.aristanetworks.com* as the switch's domain name.

```
switch(config)#management xmpp
test1(config-mgmt-xmpp)#server arista-xmpp
test1(config-mgmt-xmpp)#domain test.aristanetworks.com
test1(config-mgmt-xmpp)#username test1@test.aristanetworks.com password 0 arista
test1(config-mgmt-xmpp)#no shutdown
```

- This command removes the domain name from the XMPP configuration.

```
switch(config-mgmt-xmpp)#no domain
switch(config-mgmt-xmpp)#
```

idle-timeout (Console Management)

The **idle-timeout (Console Management)** command configures the idle timeout period for console connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode Mgmt-console

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - ***<1*** to ***86400>*** Automatic timeout period (minutes).

Example

- These commands configure a console idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 180
switch(config-mgmt-console)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 0
switch(config-mgmt-console)#
```

idle-timeout (SSH Management)

The **idle-timeout (SSH Management)** command configures the idle timeout period for SSH connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode Mgmt-ssh Configuration

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - *<**1** to **86400**>* Automatic timeout period (minutes).

Example

- These commands configure an ssh idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
switch(config-mgmt-ssh)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 0
switch(config-mgmt-ssh)#
```

idle-timeout (Telnet Management)

The **idle-timeout (Telnet Management)** command configures the idle timeout period for Telnet connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode Mgmt-telnet

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - *<**1** to **86400**>* Automatic timeout period (minutes).

Example

- These commands configure a telnet idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 180
switch(config-mgmt-telnet)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 0
switch(config-mgmt-telnet)#
```

management api http-commands

The **management api http-commands** command places the switch in mgmt-api-http-cmds configuration mode.

The **no management api http-commands** and **default management api http-commands** commands delete mgmt-api-http-command configuration mode statements from ***running-config***.

Mgmt-api-http-cmds configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-api-http-cmds configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode Global Configuration

Command Syntax

```
management api http-commands
no management api http-commands
default management api http-commands
```

Commands Available in Mgmt-api-http-commands Configuration Mode

- **protocol http (API Management)**
- **protocol https (API Management)**
- **protocol https certificate (API Management)**
- **shutdown (API Management)**
- **vrf (API Management)**

Example

- This command places the switch in mgmt-api-http-cmds configuration mode.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#
```

- This command returns the switch to global management mode.

```
switch(config-mgmt-api-http-cmds)#exit
switch(config)#
```

management console

The **management console** command places the switch in mgmt-console configuration mode to adjust the idle timeout period for console connection sessions. The idle timeout period determines the inactivity interval that terminates a connection session.

The **no management console** and **default management console** commands delete mgmt-console configuration mode statements from ***running-config***.

Mgmt-console configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-console configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode Global Configuration

Command Syntax

```
management console
no management console
default management console
```

Commands Available in mgmt-console Configuration Mode

- **idle-timeout (Console Management)**

Example

- This command places the switch in mgmt-console configuration mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-console)#exit
switch(config)#
```

management ssh

The **management ssh** command places the switch in mgmt-ssh configuration mode to adjust SSH session connection parameters.

The **no management ssh** and **default management ssh** commands delete the mgmt-ssh configuration mode statements from ***running-config***.

Mgmt-ssh configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-ssh configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode Global Configuration

Command Syntax

```
management ssh
no management ssh
default management ssh
```

Commands Available in Mgmt-ssh Configuration Mode

- authentication mode (Management-SSH)
- cipher (Management-SSH)
- fips restrictions (Management-SSH)
- hostkey (Management-SSH)
- idle-timeout (Management-SSH)
- ip access group (Management-SSH)
- ipv6 access group (Management-SSH)
- key-exchange (Management-SSH)
- login timeout (Management-SSH)
- mac hmac (Management-SSH)
- server-port (Management-SSH)
- shutdown (Management-SSH)
- vrf (Management-SSH)

Example

- This command places the switch in mgmt-ssh configuration mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

management telnet

The **management telnet** command places the switch in mgmt-telnet configuration mode to adjust telnet session connection parameters.

The **no management telnet** and **default management telnet** commands delete the mgmt-telnet configuration mode statements from ***running-config***.

Mgmt-telnet configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-telnet configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode Global Configuration

Command Syntax

```
management telnet
no management telnet
default management telnet
```

Commands Available in mgmt-telnet Configuration Mode

- idle-timeout (Management-Telnet)
- ip access group (Management-Telnet)
- ipv6 access group (Management-Telnet)
- shutdown (Management-Telnet)
- vrf (Management-Telnet)

Example

- This command places the switch in mgmt-telnet configuration mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-telnet)#exit
switch(config)#
```

management xmpp

The **management xmpp** command places the switch in mgmt-xmpp configuration mode. Management over XMPP is disabled by default. To enable XMPP, you must provide the location of the XMPP server along with the username and password for the switch.

The **no management xmpp** and **default management xmpp** commands delete the mgmt-xmpp configuration mode statements from ***running-config***.

Mgmt-xmpp configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-xmpp configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode Global Configuration

Command Syntax

```
management xmpp
no management xmpp
default management xmpp
```

Commands Available in Mgmt-xmpp Configuration Mode

- domain (Management-xmpp)
- server (Management-xmpp)
- session (Management-xmpp)
- shutdown (Management-xmpp)
- switch-group (Management-xmpp)
- username (Management-xmpp)
- vrf (Management-xmpp)

Example

- This command places the switch in mgmt-xmpp configuration mode:

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-xmpp)#exit
switch(config-mgmt-xmpp)#
```

protocol http (API Management)

The **protocol http** command enables the hypertext transfer protocol (HTTP) server.

You can only have HTTP or HTTPS enabled at one time. Trying to enable both simultaneously generates this error message:

```
% Cannot enable HTTP and HTTPS simultaneously
```

The **no protocol http** and **default protocol http** commands disable the HTTP server by removing the **protocol http** statement from ***running-config*.**

Command Mode Mgmt-API Configuration

Command Syntax

```
protocol http [TCP_PORT]
no protocol http
default protocol http
```

Parameters

- ***TCP_PORT*** Port number to be used for the HTTP server. Options include:
  - <no parameter> Specifies default port number 80.
  - **port** <***1*** to ***65535***> Specifies HTTP server port number. Value ranges from ***1*** to ***65535***.

Related Commands

- **management api http-commands** places the switch in Management-api configuration mode.

Examples

- These commands enables the management API for the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#
```

protocol https (API Management)

The **protocol https** command enables the HTTP secure server. The HTTP secure server is active by default.

You can only have HTTP or HTTPS enabled at one time, if you try to enable them both you will receive the error message:

```
% Cannot enable HTTP and HTTPS simultaneously
```

The **default protocol https** command restores the default setting by removing the **no protocol https** statement from ***running-config*.** The **no protocol https** command disables the HTTP secure server.

Command Mode Mgmt-API Configuration

Command Syntax

```
protocol https [TCP_PORT]
no protocol https
default protocol https
```

Parameters

- *TCP_PORT* Port number to be used for the HTTPS server. Options include:
  - <no parameter> Specifies default port number 443.
  - **port** <*1* to *65535*> Specifies HTTP server port number. Value ranges from *1* to *65535*.

Related Commands

- **management api http-commands** places the switch in Management-api configuration mode.

Examples

- These commands enables service to the HTTP server. The **no shutdown** command allows access to the service.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https
switch(config-mgmt-api-http-cmds)# no shutdown
```

- These commands specifies the port number that should be used for the HTTPS server. The **no shutdown** command allows access to the service.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https port 52
switch(config-mgmt-api-http-cmds)#no shutdown
```

protocol https certificate (API Management)

The **protocol https certificate** command configures the HTTP secure server to request an X.509 certificate from the client. The client then authenticates the certificate with a public key.

The **no protocol https certificate** and **default protocol https certificate** commands restore default behavior by removing the **protocol https certificate** statement from ***running-config*.**

Command Mode Mgmt-API Configuration

Command Syntax

```
protocol https certificate
no protocol https certificate
default protocol https certificate
```

Related Commands

- **management api http-commands** places the switch in Management-api configuration mode.

Examples

- These commands configure the HTTP secure server to request an X.509 certificate from the client for authentication.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https certificate
switch(config-mgmt-api-http-cmds)#
```

server (XMPP Management)

The **server** command adds a XMPP server to ***running-config***. Multiple XMPP servers can be set up for redundancy. For redundant configurations, the XMPP server location should be a DNS name and not a raw IP address. The DNS server is responsible for returning the list of available XMPP servers, which the client can go through until an accessible server is found.

User authentication is provided by the XMPP server. Command authorization can be provided by EOS local configuration or TACACS+. The XMPP server should use the same authentication source as the switches. RADIUS is not supported as an XMPP authorization mechanism.

The **no server** and **default server** commands remove the specified XMPP server from ***running-config***.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
server SERVER_NAME [SERVER_PORT]
no server
default server
```

Parameters

- *SERVER_NAME* XMPP server location. Options include:
  - *IP address* in dotted decimal notation.
  - a host name for the XMPP server.
- *SERVER_PORT* Server port. Options include:
  - **port <*1* to *65535*>** where *number* ranges from 1 to 65535. If no port is specified, the default port 5222 is used.

Examples

- This command configures the server hostname arista-xmpp to server port 1.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#server arista-xmpp port 1
```

- This command removes the XMPP server.

```
switch(config-mgmt-xmpp)# no server
```

session privilege (XMPP Management)

The **session privilege** command will place the user in EXEC mode. The initial privilege level is meaningless by default. However, with the configuration of roles, users can add meaning to the different privilege levels. By default, XMPP does not limit access to any command.

Level 1-15: Commands accessible from EXEC Mode.

If AAA is not configured and the switch is configured to connect to the XMPP client, any message received is executed with privilege level 1 by default.

The **no session privilege** and **default session privilege** commands revert the list contents to ***none*** for the specified privilege levels.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
session privilege PRIV_LEVEL
no session privilege
default session privilege
```

Parameters

- ***PRIV_LEVEL*** Privilege levels of the commands. Value ranges from **0** and **15.**

Examples

- These commands authorizes configuration commands (privilege level config 5) for XMPP.

```
switch(config)#(config)#management xmpp
switch(config-mgmt-xmpp)#session privilege 5
switch(config-mgmt-xmpp)#
```

- This command removes the privilege levels set for the XMPP session.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#no session privilege
```

show inventory

The **show inventory** command displays the hardware components installed in the switch. Serial numbers and a description is also provided for each component.

Command Mode EXEC

Command Syntax

show inventory

Examples

- This command displays the hardware installed in a DCS-7150S-52 switch.

```
switch>show inventory
System information
  Model                     Description
  ------------------------- ----------------------------------------------------
  DCS-7150S-52-CL           52-port SFP+ 10GigE 1RU + Clock

  HW Version  Serial Number  Mfg Date
  ----------- -------------- ----------
  02.00       JPE13120702    2013-03-27

System has 2 power supply slots
  Slot Model            Serial Number
  ---- ---------------- ----------------
  1    PWR-460AC-F      K192KU00241CZ
  2    PWR-460AC-F      K192L200751CZ

System has 4 fan modules
  Module  Number of Fans  Model            Serial Number
  ------- --------------- ---------------- ----------------
  1       1               FAN-7000-F       N/A
  2       1               FAN-7000-F       N/A
  3       1               FAN-7000-F       N/A
  4       1               FAN-7000-F       N/A

System has 53 ports
  Type             Count
  ---------------- ----
  Management       1
  Switched         52

System has 52 transceiver slots
  Port Manufacturer     Model            Serial Number    Rev
  ---- ---------------- ---------------- ---------------- ----
  1    Arista Networks  SFP-10G-SR       XCW1225FD753     0002
  2    Arista Networks  SFP-10G-SR       XCW1225FD753     0002
<-------OUTPUT OMITTED FROM EXAMPLE-------->
  51   Arista Networks  SFP-10G-SR       XCW1225FD753     0002
  52   Arista Networks  SFP-10G-SR       XCW1225FD753     0002

switch>
```

show xmpp neighbors

The **show xmpp neighbors** command displays all neighbors and their connection status. The XMPP server keeps track of all relationships between its users.

Command Mode EXEC

Command Syntax

show xmpp neighbors

Example

- This command displays all the XMPP neighbors and their connection status.

```
switch#show xmpp neighbors
Neighbor                         State           Last Seen Login Time
-------------------------------  --------------- ------------------------
admin@test.aristanetworks.com    present         0:01:40 ago
test1@test.aristanetworks.com    present         20:29:39 ago

Neighbor                         Status Message
-------------------------------  ----------------------------------------
admin@test.aristanetworks.com
test1@test.aristanetworks.com    Arista Networks DCS-7048T-4S
switch#
```

show xmpp status

The **show xmpp status** command displays the current XMPP connection status to the server.

The XMPP server keeps track of all relationships between its users. In order for two users to directly communicate, this relationship must first be established and confirmed by the other party.

Switches automatically confirm requests from outside parties as long as they are a user from the same domain name, for example when you chat with your switch from your own XMPP chat client.

Command Mode EXEC

Command Syntax

show xmpp status

Example

- This command displays the current XMPP connection status to the server.

```
switch# show xmpp status
XMPP Server:  port 5222
Client username: test@test.aristanetworks.com
Default domain: test.aristanetworks.com
Connection status: connected
switch#
```

show xmpp switch-group

The **show xmpp switch-group** command displays the configured and active switch groups for the switch.

Command Mode EXEC

Command Syntax

```
show xmpp switch-group
```

Example

- This command displays the configured and active switch groups.

```
switch#show xmpp switch-group
testroom@conference.test.aristanetworks.com
switch#
```

shutdown (API Management)

The **shutdown** command, in Mgmt-API mode, disables or enables management over API on the switch. API is disabled by default.

The **no shutdown** command, in Mgmt-API mode, re-enables the management API access**.**

The default shutdown command, in Mgmt-API mode, disables the management API access and removes the command from the from ***running-config***.

Command Mode Mgmt-API Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **management api http-commands** places the switch in Management-API configuration mode.

Example

- These commands disables API access to the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)# shutdown
switch(config-mgmt-api-http-cmds)#
```

- These commands enables API access to the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)# no shutdown
switch(config-mgmt-api-http-cmds)#
```

shutdown (Telnet Management)

The **shutdown** command, in management-telnet mode, disables or enables Telnet on the switch. Telnet is disabled by default. The **management telnet** command places the switch in management-telnet mode.

- To enable Telnet, enter **no shutdown** at the management-telnet prompt.
- To disable Telnet, enter **shutdown** at the management-telnet prompt.

Command Mode Management-Telnet Configuration

Command Syntax

```
shutdown
no shutdown
```

Example

- These commands enable Telnet, then return the switch to global configuration mode.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
switch(config-mgmt-telnet)#exit
switch(config)#
```

- This command disables Telnet.

```
switch(config-mgmt-telnet)#shutdown
```

shutdown (XMPP Management)

The **shutdown** command, in mgmt-xmpp mode, disables or enables management over XMPP on the switch. XMPP is disabled by default.

The **no shutdown** and **default shutdown** commands re-enable XMPP by removing the **shutdown** command from ***running-config***.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands enable management over XMPP, then return the switch to global configuration mode.

```
switch(config-mgmt-xmpp)#no shutdown
switch(config-mgmt-xmpp)#exit
switch(config)#
```

- This command disables management over XMPP.

```
switch(config-mgmt-xmpp)#shutdown
switch(config-mgmt-xmpp)#
```

switch-group (XMPP Management)

The **switch-group** command allows you to configure each switch to join specified chat rooms on startup. In order for the switch to participate in a chat group, the switch has to be configured to belong to the specified chatroom.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
switch-group name SECURITY
no switch-group
default switch-group
```

Parameters

- *name* Group name text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   -   _
  =   +   ;   <   >   ,   .   ~   |
  ```

- *SECURITY* password assignment.
  - **password** *pwd_txt* *name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 0** *pwd_txt* *name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 7** *pwd_txt* *name* is protected by specified password. *pwd_txt* is encrypted string.

Guidelines

- A switch group is an arbitrary grouping of switches within the network which belong to one chat group.
- In order to belong to one or more switch groups, the switch has to be manually assigned to it.
- Switch groups are defined dynamically based on the configuration of all of the switches in the network.
- As per the multi-user chat XMPP standard (XEP-0045), switch groups have a full name of GROUPNAME@conference.DOMAIN
- All CLI commands allow either the full group name or the short name, which are appended the @conference.DOMAIN
- If the switch belongs to multiple chat rooms, you must configure each group with a separate command.

Examples

- These commands configures the switch-group to be part of the chatroom.

  ```
  switch(config)#management xmpp
     switch(config-mgmt-xmpp)#switch-group testroom@conference.test.aristanetworks.com
     password 0 arista
  ```

- Use the **show xmpp switch-group** to verify the active switch-group for the switch.

  ```
  switch# show xmpp switch-group
  testroom@conference.test.aristanetworks.com
  ```

username (XMPP Management)

The **username** command configures the switch's username and password on the XMPP server.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode Mgmt-xmpp Configuration

Command Syntax

```
username name SECURITY
no username
default username
```

Parameters

- *name* username text that defines the XMPP username and password.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   (   )   -   _   =
  +   {   }   [   ]   ;   <   >   ,   .   ~   |
  ```

- ***SECURITY*** password assignment.

  — **password** *pwd_txt* *name* specifies and unencrypted shared key. *pwd_txt* is a clear-text string.
  — **password 0** *pwd_txt* *name* specifies and unencrypted key. *pwd_txt* is a clear-text string.
  — **password 7** *pwd_txt* *name* specifies a hidden key. *pwd_txt* is encrypted string.
  —

Guidelines

Encrypted strings entered through this parameter are generated elsewhere. The **password 7** option (***SECURITY***) is typically used to enter a list of username-passwords from a script.

Examples

- These commands create the username and assigns it a password. The password is entered in clear text because the parameter is set to 0.

  ```
  switch(config)#management xmpp
  switch(config-mgmt-xmpp)#server arista-xmpp
  switch(config-mgmt-xmpp)#domain test.aristanetworks.com
  switch(config-mgmt-xmpp)#username test1@test.aristanetworks.com password 0 arista
  switch(config-mgmt-xmpp)#no shutdown
  ```

- This command removes all usernames from the XMPP server.

  ```
  switch(config-mgmt-xmpp)#no username
  switch(config-mgmt-xmpp)#
  ```

vrf (API Management)

The **vrf** command places the switch in VRF configuration mode for the server. If the named VRF does not already exist, this command creates it.

Command Mode Mgmt-API Configuration

Command Syntax

vrf *VRF_INSTANCE*

Parameters

- *VRF_INSTANCE* specifies the VRF instance.
  - **default** Instance is created in the default VRF.
  - *vrf_name* Instance is created in the specified user-defined VRF.

Related Commands

- **management api http-commands** places the switch in Management-api configuration mode.

Example

- This command creates a VRF named *management-vrf* and places the switch in VRF configuration mode for the server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#vrf management-vrf
switch(config-mgmt-api-http-cmds-vrf-management-vrf)#
```

vrf (XMPP Management)

The **vrf** command places the switch in VRF configuration mode for the XMPP server. If the named VRF does not already exist, this command creates it.

The VRF configuration for the client is for the entire XMPP service, rather than per server. All servers resolving on a particular hostname must be reachable in the same VRF.

Command Mode Mgmt-xmpp Configuration

Command Syntax

vrf [***VRF_INSTANCE***]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - **default** Instance is created in the default VRF.
  - *vrf_name* Instance is created in the specified user-defined VRF.

Example

- This command creates a VRF named *management-vrf* and places the switch in VRF configuration mode for the server.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#vrf management-vrf
switch(config-mgmt-xmpp)
```

xmpp send

The **xmpp send** command can be used to connect to the XMPP server and send messages to switches or switch groups within the network.

Before switches can send messages to each other, they must friend each other. An easy way to have them auto friend each other is to have them join the same chat room. The friendship between switches can be verified by using the **show xmpp neighbor** command.

Command Mode Privileged EXEC

Command Syntax

xmpp send to *neighbor* ***XMIT_TYPE*** *content*

Parameters

- *neighbor* Options include switches or switch groups within the network that are connected as friends in a chat room.
- ***XMIT_TYPE*** Transmission type. Valid options include:
  - **command** Sends an XMPP command.
  - **message** Sends an XMPP message.
- *content* The command you want the friends within the chat room to display or execute.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g. into configuration mode)
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospace font, such as Courier, for the incoming messages.

Example

- This command sends the switch in the chat room the request to execute the **show version** command.

```
switch# xmpp send test2 command show version
message from user: test2@test.aristanetworks.com
-----------------------------------------------
Hardware version:    04.40
Serial number:       JFL08432083
System MAC address:  001c.7301.7d69
Software image version: 4.12.3
Architecture:           i386
Internal build version: 4.12.3
Internal build ID:      f5ab5f57-9c26-4fe4-acaa-fb60fa55d01d
Uptime:                 2 hours and 38 minutes
Total memory:           1197548 kB
Free memory:            182452 kB
```

xmpp session

The **xmpp session** command is similar to running SSH from the switch. The user is required to input their username (default is to USER@DEFAULTDOMAIN) and password in order to connect to the XMPP server. This command allows you to interact in the enable mode with a switch or switch group over XMPP using the standard CLI, with access to help and tab completion. All commands are then executed remotely and only the non-empty results are displayed on the screen.

Command Mode Privileged EXEC

Command Syntax

xmpp session *switchgroup*

Parameters

- *switchgroup* The option includes the switch group within the network that is connected as friends in a chat room.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g. into configuration mode)
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospace font, such as Courier, for the incoming messages.

Example

- This command displays the status of Ethernet 3 from *test1*, which is a member of the switch group chat room.

```
switch# xmpp session all@test.aristanetworks.com
xmpp-all# show int Eth3 status

response from: test1@test.aristanetworks.com

------------------------------------------------

Port  Name  Status     Vlan     Duplex  Speed   Type
Et3   bs3   connected  in Po3   a-full  a-1000  10GBASE-SR
switch#
```