Chapter 42

SNMP

This chapter describes the Arista switch SNMP agent and contains these sections:

- Section 42.1: SNMP Introduction
- Section 42.2: SNMP Conceptual Overview
- Section 42.3: Configuring SNMP
- Section 42.4: SNMP Commands

42.1 SNMP Introduction

Arista Networks switches support many standard SNMP MIBs, making it easier to integrate these platforms into existing network management infrastructures. With only a few configurations, many public domain and commercially available network management tools can quickly manage Arista switches out of the box. Support of SNMP V2 groups and views and V3 security allow network managers to tune switch monitoring to match the administration policy of the IT organization.

42.2 SNMP Conceptual Overview

Simple Network Management Protocol (SNMP) is a protocol that provides a standardized framework and a common language to monitor and manage network devices.

42.2.1 SNMP Structure

The SNMP framework has three parts:

- **SNMP manager**: The SNMP manager controls and monitors network host activities and is typically part of a Network Management System (NMS).
- **SNMP agent**: The SNMP agent is the managed device component that manages and reports device information to the manager.
- **Management Information Base (MIB)**: The MIB stores network management information.

The agent and MIB reside on the switch. Enabling the SNMP agent requires the definition of the manager-agent relationship. The agent contains MIB variables whose values the manager can request or change. The agent gathers data from the MIB and responds to requests for information. For a list of supported MIBs, please refer to the release notes for a specific EOS version.

This chapter discusses enabling the SNMP agent on an Arista switch and controlling notification transmissions from the agent. Information on using SNMP management systems is available in the appropriate documentation for the corresponding NMS application.
42.2.2 SNMP Notifications

SNMP notifications are messages, sent by the agent, informing of an event or a network condition. A *trap* is an unsolicited notification. An *inform* (or inform request) is a trap that includes a request for a confirmation that the message is received. Events that a notification can indicate include improper user authentication, restart, and connection losses.

For a list of supported traps, please refer to the release notes for a specific EOS version.

42.2.3 SNMP Versions

Arista switches support the following SNMP versions:

- **SNMPv1**: The Simple Network Management Protocol, defined in RFC 1157. Security is based on community strings.
- **SNMPv2c**: Community-string based Administrative Framework for SNMPv2, defined in RFC 1901, RFC 1905, and RFC 1906. Security is based on SNMPv1.
- **SNMPv3**: Version 3, as defined in RFCs 2273 to 2275.

42.3 Configuring SNMP

This section describes the steps that configure the switch SNMP agent to communicate with an SNMP manager, including the following:

- **Enabling and Disabling SNMP**
- **Configuring Community Access Control**
- **Configuring SNMP Parameters**
- **Configuring the Agent to Send Notifications**
- **Extending the SNMP Agent Through Run Time Scripts**

42.3.1 Enabling and Disabling SNMP

SNMP is enabled globally by issuing any `snmp-server community` or `snmp-server user` command. The `no snmp-server` command disables SNMP agent operation by removing all non-default `snmp-server` commands from *running-config*.

42.3.2 Enabling SNMP in a VRF

By default, SNMP is enabled only in the *default* VRF. The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled.

To enable or disable SNMP in a VRF, use the `snmp-server vrf` command.

42.3.3 Configuring Community Access Control

SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. A Network Management System (NMS) can access the switch only if its community string matches at least one of the switch’s community strings.

The `snmp-server community` command configures the community string.
Example

- This command adds the community string `ab_1` to provide read-only access to the switch agent.

```
switch(config)#snmp-server community ab_1 ro
```

Community statements can reference views to limit MIB objects that are available to a manager. A view is a community string object that specifies a subset of MIB objects. The `snmp-server view` command configures the community string.

Example

- These commands create a view that includes all objects in the `system` group except for those in `system.2`.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
```

- This command adds the community string `lab_1` to provide read-only access to the switch agent for the previously defined view.

```
switch(config)#snmp-server community lab_1 view sys-view
```

42.3.4 Configuring SNMP Parameters

This section describes these SNMP parameter configuration tasks:

- Configuring the Engine ID
- Configuring the Group
- Configuring the User
- Configuring the Host
- Enabling Link Trap Generation
- Configuring the Chassis-id String
- Configuring the Contact String
- Configuring the Location String

Configuring the Engine ID

The `snmp-server engineID remote` command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. Use the `snmp-server engineID local` command for the local engine.

A remote agent's engine ID must be configured before remote users for that agent are configured. User authentication and privacy digests are derived from the engine ID and user passwords. The configuration command fails if the remote engine ID is not configured first.

**Important!** When the remote engine ID is changed, all user passwords associated with the engine must be reconfigured.

Example

- This command configures DC945798CAB4 as the name of the remote SNMP engine located at 12.23.104.25, UDP port 162

```
switch(config)#snmp-server engineID remote 10.23.104.25 udp-port DC945798CA
```

```
Configuring the Group

An SNMP group grants specific levels of SNMP access to group users. The `snmp-server group` command configures a new SNMP group.

Example

- This command configures `normal_one` as an SNMPv3 group (authentication and encryption) that provides access to the `all-items` read view.

  ```
  switch(config)#snmp-server group normal_one v3 priv read all-items
  switch(config)#
  ```

Configuring the User

Members of SNMP groups are called “users.” The `snmp-server user` command allows a new user to be added an SNMP group and configures that user’s parameters. Remote users are configured by specifying the IP address or port number that accesses the user’s SNMP agent.

Example

- This command configures the local SNMPv3 user `tech-1` as a member of the SNMP group `tech-sup`.

  ```
  switch(config)#snmp-server user tech-1 tech-sup v3
  switch(config)#
  ```

- This command configures the remote SNMPv3 user `tech-2` as a member of the SNMP group `tech-sup`. The remote user is on the agent located at 13.1.1.4.

  ```
  switch(config)#snmp-server user tech-2 tech-sup remote 13.1.1.4 v3
  switch(config)#
  ```

Configuring the Host

The `snmp-server host` command configures an SNMP host (to which SNMP traps will be sent). The `snmp-server host` command sets the community string if it was not previously configured.

Example

- This command adds a v2c inform notification recipient at 12.15.2.3 using the community string `comm-1`.

  ```
  switch(config)#snmp-server host 12.15.2.3 informs version 2c comm-1
  switch(config)#
  ```

Enabling Link Trap Generation

The `snmp trap link-change` command enables SNMP link trap generation on the configuration mode interface. SNMP link trap generation is enabled by default. If SNMP link trap generation was previously disabled, this command removes the corresponding `no snmp link-status` statement from the configuration. The `show snmp notification` command displays the SNMP link trap generation information.

Example

- This command disables SNMP link trap generation on the Ethernet 5 interface.

  ```
  switch(config-if-Et5)#no snmp trap link-change
  switch(config-if-Et5)#
  ```
Specifying the Source Interface

The `snmp-server local-interface` command specifies the interface from where an SNMP trap originates. The `show snmp local-interface` command displays the interface of the IP address for SNMP traps.

Example

- This command configures the Ethernet 1 interface as the source of SNMP traps and informs.

  ```
  switch(config)#snmp-server local-interface ethernet 1
  switch(config)#
  ```

Configuring the Chassis-id String

The chassis ID string is typically set to the serial number of the switch. The SNMP manager uses this string to associate all data retrieved from the switch with a unique identifying label. Under normal operating conditions, editing the chassis ID string contents is unnecessary.

The `snmp-server chassis-id` command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The `show snmp` command displays the chassis ID.

Example

- This command configures `xyz-1234` as the chassis-ID string, then displays the result.

  ```
  switch(config)#snmp-server chassis-id xyz-1234
  switch(config)#show snmp
  Chassis: xyz-1234  <--chassis ID
  ```

Configuring the Contact String

The SNMP contact string is information text that typically displays the name of a person or organization associated with the SNMP agent.

The `snmp-server contact` command configures the system contact string. The contact string is displayed by the `show snmp` and `show snmp v2-mib contact` commands.
Example
- These commands configure **Bonnie H at 3-1470** as the contact string.

```
switch(config)#snmp-server contact Bonnie H at 3-1470
switch(config)#
```

Configuring the Location String

The location string typically provides information about the physical location of the SNMP agent. The `snmp-server location` command configures the system location string. By default, the system location string is not set.

Example
- These commands configure **lab-25** as the location string.

```
switch(config)#snmp-server location lab_25
switch(config)#show snmp v2-mib location
Location: lab_25
switch(config)#
```

42.3.5 Configuring the Agent to Send Notifications

The following steps are mandatory when setting up the SNMP agent to send notifications:

**Step 1** Configure the remote engine ID.

**Step 2** Configure the group.

**Step 3** Configure the user.

**Step 4** Configure the host.

**Step 5** Enable link trap generation on the interfaces.

Section 42.3.4 describes each of these tasks.

42.3.6 Extending the SNMP Agent Through Run Time Scripts

The switch supports the execution of user supplied scripts to service portions of the OID space.

Scripts run under one of two operational modes:
- Normal: scripts run over an indefinite period to process subsequent objects after the initial request. Maintaining an executing script avoids startup and connection delay each time an object requires processing.
- One-shot mode: scripts process a single object, then terminates execution.

Normal extension scripts are conceptually multithreaded: one thread collects data and the other thread is ready to communicate with snmpd. One-shot scripts process a single object, running once and exiting. Startup and data collection overhead is required for each request. In both modes, the SNMP server is blocked from serving other requests when waiting for script responses.

The `snmp-server extension` command configures the execution of user supplied scripts to service portions of the OID space.

Example
- This command specifies the file **example.sh**, located in flash as the script file that services the specified OID space in normal mode.

```
switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
switch(config)#
```
42.3.6.1 Normal Script Behavior

The first time the SNMP server requires a script result, it launches it with no arguments. The server communicates with the script through stdin/stdout. Before each request, the script is the string `PING` on stdin. The expected response is printing `PONG` to stdout.

GET and GETNEXT Requests

For GET and GETNEXT requests, the script is passed two lines on stdin, the command (get or getnext) and the requested OID. The expected response from the script is the printing of three lines to stdout: , the TYPE, the OID for the result varbind, and the VALUE itself.

Table 42-1 lists legal TYPE values and resulting VALUE encodings. If the command does not return an appropriate varbind, it should print "NONE" to stdout and continue running; this results in an SNMP noSuchName error or a noSuchInstance exception.

Table 42-1 Extension Script Type and Encoding

<table>
<thead>
<tr>
<th>Type string</th>
<th>SNMP type</th>
<th>Encoding for script</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td>Integer32</td>
<td>integer</td>
</tr>
<tr>
<td>unsigned</td>
<td>Unsigned32</td>
<td>integer</td>
</tr>
<tr>
<td>gauge</td>
<td>Gauge32</td>
<td>integer</td>
</tr>
<tr>
<td>counter</td>
<td>Counter32</td>
<td>integer</td>
</tr>
<tr>
<td>counter64</td>
<td>Counter64</td>
<td>integer</td>
</tr>
<tr>
<td>timetick</td>
<td>TimeTicks</td>
<td>integer</td>
</tr>
<tr>
<td>ipaddress</td>
<td>IpAddress</td>
<td>a.b.c.d</td>
</tr>
<tr>
<td>objectid</td>
<td>ObjectID</td>
<td>1.3.6.1.42.99.2468</td>
</tr>
<tr>
<td>octet</td>
<td>OctetString</td>
<td>hexadecimal string</td>
</tr>
<tr>
<td>opaque</td>
<td>Opaque</td>
<td>hexadecimal string</td>
</tr>
<tr>
<td>string</td>
<td>OctetString</td>
<td>ascii string</td>
</tr>
</tbody>
</table>

SET

For SET requests, script is passed three lines on stdin: the command (set), and the requested OID, and the type and value, both on the same line. If the assignment is successful, the expected script response is to print `DONE` to stdout. Indicated errors by writing one of the error strings described in Table 42-2 in each case, the command should continue running.

Table 42-2 Set Request Error Strings

<table>
<thead>
<tr>
<th>error strings</th>
<th>SNMP type</th>
<th>Encoding for script</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorization-error</td>
<td>no-access</td>
<td>too-big</td>
</tr>
<tr>
<td>bad-value</td>
<td>no-creation</td>
<td>undo-failed</td>
</tr>
<tr>
<td>commit-failed</td>
<td>no-such-name</td>
<td>wrong-type</td>
</tr>
<tr>
<td>gen-error</td>
<td>not-writable</td>
<td>wrong-length</td>
</tr>
<tr>
<td>inconsistent-name</td>
<td>read-only</td>
<td>wrong-encoding</td>
</tr>
<tr>
<td>inconsistent-value</td>
<td>resource unavailable</td>
<td>wrong-value</td>
</tr>
</tbody>
</table>

42.3.6.2 One Shot Script Behavior

The command should exit after it finishes processing a single object.
GET and GETNEXT

For each GET or GETNEXT request, the script is invoked once for each OID in the space that it serves. It receives two arguments: -g for GET or -n for GETNEXT, and the requested OID.

The expected script response is the response varbind as three separate lines printed to stdout: the result OID, the type, and the value.

If the command does not return an appropriate varbind, then the script should exit without producing any output. This results in an SNMP noSuchName error, or a noSuchInstance exception.

Possible reasons that a command would not return an appropriate varbind includes:

- The specified OID didn’t correspond to a valid instance for a GET request.
- There were no following instances for a GETNEXT.

SET

A SET request results in the command being called with the arguments: -s, OID, TYPE and VALUE, where TYPE is a listed token Table 42-1, indicating the type of the value passed as the third parameter.

When the assignment is successful, the script exits without producing any output. Errors are indicated by writing just the error name (Table 42-2); the agent generates the appropriate error response.
42.4 **SNMP Commands**

**Global Configuration Commands**
- no snmp-server
- snmp-server chassis-id
- snmp-server community
- snmp-server contact
- snmp-server enable traps
- snmp-server engineID local
- snmp-server engineID remote
- snmp-server extension
- snmp-server group
- snmp-server host
- snmp-server location
- snmp-server local-interface
- snmp-server user
- snmp-server view
- snmp-server vrf

**Interface Configuration Commands**
- snmp trap link-change

**Display Commands**
- show snmp
- show snmp community
- show snmp engineID
- show snmp group
- show snmp local-interface
- show snmp mib
- show snmp notification host
- show snmp notification
- show snmp user
- show snmp v2-mib chassis
- show snmp v2-mib contact
- show snmp v2-mib location
- show snmp view
**no snmp-server**

The `no snmp-server` and `default snmp-server` commands disable Simple Network Management Protocol (SNMP) agent operation by removing all `snmp-server` commands from `running-config`.

SNMP is enabled with any `snmp-server community` or `snmp-server user` command.

**Command Mode**
- Global Configuration

**Command Syntax**

- `no snmp-server`
- `default snmp-server`

**Example**
- This command disables SNMP agent operation on the switch.
  ```
  switch(config)#no snmp-server
  switch(config)#
  ```
show snmp

The show snmp command displays SNMP information including the SNMP counter status and the chassis ID string.

Example

EXEC

Command Syntax

show snmp

Example

- This command displays SNMP counter status, the chassis ID, the previously configured location string, logging status and destination, and the VRFs in which the SNMP agent is operating.

switch>show snmp
Chassis: JFL08320162
Location: 5470ga.dc
2329135 SNMP packets input
   0 Bad SNMP version errors
   0 Unknown community name
   0 Illegal operation for community name supplied
   0 Encoding errors
   38132599 Number of requested variables
   0 Number of altered variables
   563934 Get-request PDUs
   148236 Get-next PDUs
   0 Set-request PDUs
2329437 SNMP packets output
   0 Too big errors
   0 No such name errors
   0 Bad value errors
   0 General errors
   2329135 Response PDUs
   0 Trap PDUs
SNMP logging: enabled
   Logging to 172.22.22.20.162
SNMP agent configured in VRFs: default
SNMP agent enabled in default VRF
switch>
show snmp community

The `show snmp community` command displays the Simple Network Management Protocol (SNMP) community access strings configured by the `snmp-server community` command.

**Example**

EXEC

**Command Syntax**

```
show snmp community
```

**Example**

- This command displays the list of community access strings configured on the switch.

```
switch>show snmp community

Community name: public
switch>
```
show snmp engineID

The `show snmp engineID` command displays the local SNMP engine information configured on the switch.

**Example**

```
EXEC
```

**Command Syntax**
```
show snmp engineID
```

**Example**
- This command displays the ID of the local SNMP engine.

```
switch>show snmp engineid
Local SNMP EngineID: f5717f001c730436d700
switch>
```
**show snmp group**

The **show snmp group** command shows the names of configured SNMP groups along with the security model, and view status of each group.

**Example**

EXEC

**Command Syntax**

```
show snmp group [GROUP_LIST]
```

**Parameters**

- **GROUP_LIST** the name of the group.
- <no parameter> displays information about all groups.
- **group_name** the name of the group.

**Field Descriptions**

- **groupname** name of the SNMP group.
- **security model** security model used by the group: v1, v2c, or v3.
- **readview** string identifying the group’s read view. Refer to **show snmp view**.
- **writeview** string identifying the group’s write view.
- **notifyview** string identifying the group’s notify view.

**Example**

- This command displays the groups configured on the switch.

  ```
  switch>show snmp group
  groupname : normal                          security model:v3 priv
  readview : all                              writeview: <no writeview specified>
  notifyview: <no notifyview specified>
  switch>
  ```
show snmp local-interface

The `show snmp local-interface` command displays the interface whose IP address is the source address for SNMP traps.

**Example**

```
EXEC
```

**Command Syntax**

```
show snmp local-interface
```

**Example**

- This command displays the source interface for the SNMP notifications.

```
switch>show snmp local-interface
SNMP source interface: Ethernet1
switch>
```
**show snmp mib**

The `show snmp mib` command displays values associated with specified MIB object identifiers (OIDs) that are registered on the switch.

**Example**

EXEC

**Command Syntax**

```
show snmp mib OBJECTS
```

**Parameters**

- **OBJECTS** object identifiers for which the command returns data. Options include:
  - `get oid_1 [oid_2 ... oid_x]` values associated with each listed OID.
  - `get-next oid_1 [oid_2 ... oid_x]` values associated with subsequent OIDs relative to listed OIDs.
  - `table oid` table associated with specified OID.
  - `translate oid` object name associated with specified OID.
  - `walk oid` objects below the specified subtree.

**Example**

- This command uses the get option to retrieve information about the sysORID.1 OID.
  
  `switch#show snmp mib get sysORID.1`
  
  `SNMPv2-MIB::sysORID[1] = OID: TCP-MIB::tcpMIB`

- This command uses the get-next option to retrieve information about the OID that is after sysORID.8.
  
  `switch#show snmp mib get-next sysORID.8`
  
  `SNMPv2-MIB::sysORDescr[1] = STRING: The MIB module for managing TCP implementations`
show snmp notification host

The show snmp notification host command displays information for Simple Network Management Protocol notification. Details include IP address and port number of the Network Management System, notification type, and SNMP version.

Example
EXEC

Command Syntax
show snmp notification host

Field Descriptions
• Notification host  IP address of the host.
• udp-port  port number.
• type  notification type.
• user  access type of the user.
• security model  SNMP version used.
• traps  details of the notification.

Example
• This command displays the hosts configured on the switch.

switch>show snmp notification host
Notification host: 172.22.22.20  udp-port: 162  type: trap
user: public  security model: v2c

switch>
### show snmp notification

The `show snmp notification` command displays the SNMP trap generation information.

**Example**

**EXEC**

**Command Syntax**

```
show snmp notification
```

**Example**

- This command displays the SNMP traps configured on the switch.

```
switch> show snmp notification

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>entity</td>
<td>entConfigChange</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>entity</td>
<td>entStateOperDisabled</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>entity</td>
<td>entStateOperEnabled</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>lldp</td>
<td>lldpRemTablesChange</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>msdpBackwardTransition</td>
<td>msdpBackwardTransition</td>
<td>Yes</td>
</tr>
<tr>
<td>msdpEstablished</td>
<td>msdpEstablished</td>
<td>Yes</td>
</tr>
<tr>
<td>snmp</td>
<td>linkDown</td>
<td>Yes</td>
</tr>
<tr>
<td>snmp</td>
<td>linkUp</td>
<td>Yes</td>
</tr>
<tr>
<td>snmpConfigManEvent</td>
<td>aristaConfigManEvent</td>
<td>Yes (default)</td>
</tr>
</tbody>
</table>
| switchover         | aristaRedundancySwitchOverNotif | Yes |}

```
switch>
```
show snmp user

The show snmp user command shows information about Simple Network Management Protocol (SNMP) users. Information that the command displays about each user includes their SNMP version, the engine ID of the host where they reside, and security information.

Example

EXEC

Command Syntax

show snmp user [USER_LIST]

Parameters

- **USER_LIST**  the name of the group.
- <no parameter> displays information about all users.
- **user_name** specifies name of displayed user.

Example

- This command displays information about the users configured on the switch.

  switch>show snmp user
  
  User name: test
  
  Security model: v3
  Engine ID: f5717f001c73010e0900
  Authentication protocol: SHA
  Privacy protocol: AES-128
  Group name: normal
  
  switch>
show snmp v2-mib chassis

The `show snmp v2-mib chassis` command displays the Simple Network Management Protocol (SNMP) server serial number or the chassis ID string configured by the `snmp-server chassis-id` command.

**Example**

```
EXEC
```

**Command Syntax**

```
show snmp v2-mib chassis
```

**Example**

- This command displays the chassis ID string.

```
switch>show snmp v2-mib chassis
Chassis: JFL08320162
switch>
```
show snmp v2-mib contact

The `show snmp v2-mib contact` command displays the Simple Network Management Protocol (SNMP) system contact string configured by the `snmp-server contact` command. The command has no effect if a contact string was not previously configured.

**Example**

```
EXEC
```

**Command Syntax**

```
show snmp v2-mib contact
```

**Example**

- This command displays the contact string contents.

```
switch>show snmp v2-mib contact
Contact: John Smith
switch>
```
show snmp v2-mib location

The `show snmp v2-mib location` command displays the Simple Network Management Protocol (SNMP) system location string. The `snmp-server location` command configures system location details. The command has no effect if a location string was not previously configured.

Example

EXEC

Command Syntax

`show snmp v2-mib location`

Example

- This command displays the location string contents.

```
switch>show snmp v2-mib location
Location: santa clara
switch>
```
**show snmp view**

The `show snmp view` command displays the information of a Simple Network Management Protocol configuration and the associated MIB. SNMP views are configured with the `snmp-server view` command.

**Example**

EXEC

**Command Syntax**

```
show snmp view [VIEW_LIST]
```

**Parameters**

- **VIEW_LIST** the name of the view.
  - `<no parameter>` displays information about all views.
  - `view_name` the name of the view.

**Field Descriptions**

- **First column** view name.
- **Second column** name of the MIB object or family.
- **Third column** inclusion level of the specified family within the view.

**Example**

- These commands configure an SNMP view, then displays that view.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
switch(config)#show snmp view
sys-view system - included
sys-view system.2 - excluded
```
**snmp-server chassis-id**

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

The **no snmp-server chassis-id** and **default snmp-server chassis-id** commands restore the default chassis ID string by removing the **snmp-server chassis-id** command from the configuration.

**Command Mode**
- Global Configuration

**Command Syntax**

```
  snmp-server chassis-id  id_text
  no snmp-server chassis-id
  default snmp-server chassis-id
```

**Parameters**
- `id_text`    chassis ID string

**Example**
- These commands configure **xyz-1234** as the chassis-id string, then display the result.

```plaintext
switch(config)#snmp-server chassis-id xyz-1234
switch(config)#show snmp
Chassis: xyz-1234
<---chassis ID
8 SNMP packets input
  0 Bad SNMP version errors
  0 Unknown community name
  0 Illegal operation for community name supplied
  0 Encoding errors
  8 Number of requested variables
  0 Number of altered variables
  4 Get-request PDUs
  4 Get-next PDUs
  0 Set-request PDUs
21 SNMP packets output
  0 Too big errors
  0 No such name errors
  0 Bad value errors
  0 General errors
  8 Response PDUs
  0 Trap PDUs
SNMP logging: enabled
  Logging to taccon.162
SNMP agent enabled
switch(config)#
```
**snmp-server community**

The *snmp-server community* command configures the community string. SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. The Network Management System (NMS) must define a community string that matches at least one of the switch community strings to access the switch.

The *no snmp-server community* and *default snmp-server community* commands remove the community access string from the configuration.

**Command Mode**

Global Configuration

**Command Syntax**

```
snmp-server community string_text [MIB_VIEW] [ACCESS] [ACL_NAMES]
no snmp-server community string_text
default snmp-server community string_text
```

**Parameters**

-  *string_text*  community access string.
-  *MIB_VIEW*  community access availability. Options include:
  -  <no parameter>  community string allows access to all objects.
  -  *view view_name*  community string allows access only to objects in the *view_name* view.
-  *ACCESS*  community access availability. Options include:
  -  <no parameter>  read-only access (default setting).
  -  *ro*  read-only access.
  -  *rw*  read-write access.
-  *ACL_NAMES*  community access availability. Options include:
  -  <no parameter>  community string allows access to all objects.
  -  *list_v4*  IPv4 ACL list.
  -  *ipv6 list_v6*  IPv6 ACL list.
  -  *ipv6 list_v6 list_v4*  IPv4 and IPv6 ACL list.

**Example**

- This command adds the community string *lab_1* to provide read-only access to the switch agent.
  
  ```
  switch(config)#snmp-server community lab_1 ro
  switch(config)#
  ```
snmp-server contact

The `snmp-server contact` command configures the system contact string. The contact is displayed by the `show snmp` and `show snmp v2-mib contact` commands.

The `no snmp-server contact` and `default snmp-server contact` commands remove the `snmp-server contact` command from the configuration.

**Command Mode**
Global Configuration

**Command Syntax**

```
  snmp-server contact  contact_string
  no snmp-server contact
  default snmp-server contact
```

**Parameters**
- `contact_string` system contact string.

**Example**
- These commands configure *Bonnie H* as the contact string, then display the result.

```
  switch(config)#snmp-server contact Bonnie H
  switch(config)#show snmp
  Chassis: xyz-1234
  Contact: Bonnie H.
  8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
  24 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
  SNMP logging: enabled
    Logging to taccon.162
  SNMP agent enabled
  switch(config)#
```
### snmp-server enable traps

The **snmp-server enable traps** command enables both Simple Network Management Protocol (SNMP) traps and SNMP inform requests; use the **snmp-server host** command to specify which will receive SNMP notifications. Sending notifications requires at least one **snmp-server host** command.

The **snmp-server enable traps** and **no snmp-server enable traps** commands, without a MIB parameter, specify the default notification trap generation setting for all MIBs. These commands, when specifying a MIB, control notification generation for the specified MIB. The **default snmp-server enable traps** command resets notification generation to the default setting for the specified MIB.

**Command Mode**

Global Configuration

**Command Syntax**

```
snmp-server enable traps [trap_type]
no snmp-server enable traps [trap_type]
default snmp-server enable traps [trap_type]
```

**Parameters**

- **trap_type** controls the generation of informs or traps for the specified MIB:
  - `<no parameter>` controls notifications for MIBs not covered by specific commands.
  - **entity** controls entity-MIB modification notifications.
  - **lldp** controls LLDP notifications.
  - **msdpBackwardTransition** controls msdpBackwardTransition notifications.
  - **msdpEstablished** controls msdpEstablished notifications.
  - **snmp** controls SNMP-v2 notifications.
  - **switchover** controls switchover notifications.
  - **snmpConfigManEvent** controls snmpConfigManEvent notifications.
  - **test** controls test traps.

**Examples**

- These commands enables notification generation for all MIBs except spanning tree.
  ```
  switch(config)#snmp-server enable traps
  switch(config)#no snmp-server enable traps spanning-tree
  switch(config)#
  ```

- This command enables spanning-tree MIB notification generation, regardless of the default setting.
  ```
  switch(config)#snmp-server enable traps spanning-tree
  switch(config)#
  ```

- This command resets the spanning-tree MIB notification generation to follow the default setting.
  ```
  switch(config)#default snmp-server enable traps spanning-tree
  switch(config)#
  ```

- This command enables switchover MIB notification generation, regardless of the default setting.
  ```
  switch(config)#snmp-server enable traps switchover
  switch(config)#
  ```

- This command resets the switchover MIB notification generation to follow the default setting.
  ```
  switch(config)# default snmp-server enable traps switchover
  switch(config)#
  ```
**snmp-server engineID local**

The `snmp-server engineID local` command configures the name for the local Simple Network Management Protocol (SNMP) engine. The default SNMP engineID is generated by the switch and is used when an engineID is not configured with this command. The `show snmp engineID` command displays the default or configured engine ID.

SNMPv3 authenticates users through security digests (MD5 or SHA) that are based on user passwords and the local engine ID. Passwords entered on the CLI are similarly converted, then compared to the user’s security digest to authenticate the user.

**Important!** Changing the local engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The `no snmp-server engineID` and `default snmp-server engineID` commands restore the default engineID by removing the `snmp-server engineID` command from the configuration.

**Command Mode**
Global Configuration

**Command Syntax**
- `snmp-server engineID local engine_hex`
- `no snmp-server engineID local`
- `default snmp-server engineID`

**Parameters**
- `engine_hex` the switch’s name for the local SNMP engine (hex string).
  The string must consist of at least ten characters with a maximum of 64 characters.

**Example**
- This command configures DC945798CAB4 as the name of the local SNMP engine.

```plaintext
switch(config)#snmp-server engineID local DC945798CAB4
switch(config)#
```
**snmp-server engineID remote**

The `snmp-server engineID remote` command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. The switch generates a default engineID; use the `show snmp engineID` command to view the configured or default engineID.

An SNMPv3 inform requires a remote engine ID to compute the security digest that authenticates and encrypts data transmitted to remote users. SNMPv3 authenticates users with MD5 or SHA through the engine ID and user passwords. CLI passwords are similarly authenticated.

---

**Important!** Changing the engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The `no snmp-server engineID remote` and `default snmp-server engineID remote` commands remove the `snmp-server engineID remote` command from the configuration.

**Command Mode**
Global Configuration

**Command Syntax**

```
snmp-server engineID remote engine_addr [PORT] engine_hex
no snmp-server engineID remote engine_addr [PORT]
default snmp-server engineID remote engine_addr [PORT]
```

**Parameters**

- `engine_addr` location of remote engine (IP address or host name).
- `PORT` udp port location of the remote engine. Options include:
  - `<No parameter>` port number 161 (default).
  - `udp-port port_num` port number. Ranges from 0 to 65535.
- `engine_hex` the switch’s name for the remote SNMP engine (hex string).
  The string must have at least ten characters and can contain a maximum of 64 characters.

**Example**

- This command configures DC945798CA as the engineID of the remote SNMP engine located at 10.23.10.25, UDP port 162.

  ```
  switch(config)#snmp-server engineID remote 10.23.10.25 udp-port 162 DC945798CA
  switch(config)#
  ```
**snmp-server extension**

The **snmp-server extension** command configures the execution of user supplied scripts to service portions of the OID space.

The **no snmp-server extension** and **default snmp-server extension** commands deletes the **snmp-server extension** command from **running-config**.

**Command Mode**

Global Configuration

**Command Syntax**

```
    snmp-server extension OID_space FILE_PATH [DURATION]
```

**Parameters**

- **OID_space**  
  OID branch serviced by the script, in numerical format.

- **FILE_PATH**  
  path and name of the script file. Options include:
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.

- **DURATION**  
  the execution scope of the script.
  - **<no parameter>**  
    script runs after initial request to process subsequent requests.
  - **one-shot**  
    script processes a single object (runs once), then terminates.

**Examples**

- This command specifies the file **example.sh**, located in flash, as the script file that services the listed OID space.

  ```
  switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
  ```
### snmp-server group

The **snmp-server group** command configures a new Simple Network Management Protocol (SNMP) group or modifies an existing group. An SNMP group is a data structure that user statements reference to map SNMP users to SNMP contexts and views, providing a common access policy to the specified users.

An SNMP context is a collection of management information items accessible by an SNMP entity. Each item of may exist in multiple contexts. Each SNMP entity can access multiple contexts. A context is identified by the EngineID of the hosting device and a context name.

The **no snmp-server group** and **default snmp-server group** commands delete the specified group by removing the corresponding **snmp-server group** command from the configuration.

#### Command Mode
Global Configuration

#### Command Syntax
```
**snmp-server group**  
group_name  
**VERSION**  
[**CNTX**]  
[**READ**]  
[**WRITE**]  
[**NOTIFY**]
```

```
**no snmp-server group**  
group_name  
**VERSION**
```

```
**default snmp-server group**  
group_name  
**VERSION**
```

#### Parameters
- **group_name**  the name of the group.
- **VERSION**  the security model utilized by the group.
  - v1  SNMPv1. Uses a community string match for authentication.
  - v2c  SNMPv2c. Uses a community string match for authentication.
  - v3 no auth  SNMPv3. Uses a username match for authentication.
  - v3 auth  SNMPv3. HMAC-MD5 or HMAC-SHA authentication.
  - v3 priv  SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- **CNTX**  associates the SNMP group to an SNMP context.
  - <no parameter>  command does not associate group with an SNMP context.
  - **context context_name**  associates group with context specified by **context_name**.
- **READ**  specifies read view for SNMP group.
  - <no parameter>  command does not specify read view.
  - **read read_name**  read view specified by **read_name** (string – maximum 64 characters).
- **WRITE**  specifies write view for SNMP group.
  - <no parameter>  command does not specify write view.
  - **write write_name**  write view specified by **write_name** (string – maximum 64 characters).
- **NOTIFY**  specifies notify view for SNMP group.
  - <no parameter>  command does not specify notify view.
  - **notify notify_name**  notify view specified by **notify_name** (string – maximum 64 characters).

#### Example
- This command configures normal_one as SNMP version 3 group (authentication and encryption) that provides access to the **all-items** read view.

  switch(config)#**snmp-server group normal_one v3 priv read all-items**

  switch(config)#
**snmp-server host**

The `snmp-server host` command command configures an SNMP host (to which SNMP traps will be sent) and sets the community string if it was not previously configured. The host is denoted by host location and community string. The command also specifies the type of SNMP notifications that are sent: a *trap* is an unsolicited notification; an *inform* is a trap that includes a request for a confirmation that the message is received.

The configuration can contain multiple statements to the same host location with different community strings. For instance, a configuration can simultaneously contain all of the following:

- `snmp-server host host-1 version 2c comm-1`
- `snmp-server host host-1 informs version 2c comm-2`
- `snmp-server host host-1 version 2c comm-3 udp-port 666`
- `snmp-server host host-1 version 3 auth comm-3`

The `no snmp-server host` and `default snmp-server host` commands remove the specified host by deleting the corresponding `snmp-server host` statement from the configuration. When removing a statement, the host (address and port) and community string must be specified.

**Command Mode**

Global Configuration

**Command Syntax**

```
snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
no snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
default snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
```

**Parameters**

- `host_id`  hostname or IP address of the SNMP host.
- `VRF_INST` specifies the VRF instance being modified.
  - `<no parameter>` changes are made to the default VRF.
  - `vrf vrf_name` changes are made to the specified user-defined VRF.
- `MESSAGE`  message type that is sent to the host.
  - `<no parameter>` sends SNMP traps to host (default).
  - `informs` sends SNMP informs to host.
  - `traps` sends SNMP traps to host.
- `VERSION`  SNMP version. Options include:
  - `<no parameter>` SNMPv2c (default).
  - `version 1` SNMPv1; option not available with informs.
  - `version 2c` SNMPv2c.
  - `version 3 noauth` SNMPv3; enables user-name match authentication.
  - `version 3 auth` SNMPv3; enables MD5 and SHA packet authentication.
  - `version 3 priv` SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- `comm_str`  community string to be sent with the notification as a password.
  - Arista recommends setting this string separately before issuing the `snmp-server host` command.
  - To set the community string separately, use the `snmp-server community` command.
- `PORT`  port number of the host.
•  <no parameter>  socket number set to 162 (default)
•  udp-port  p-name  socket number specified by  p-name

Guidelines
The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the default VRF. Enable or disable SNMP in a VRF with the  snmp-server vrf  command.

Example
•  This command adds a version 2c inform notification recipient.
  switch(config)#snmp-server host 10.15.2.3 informs version 2c comm-1
  switch(config)#
**snmp-server location**

The `snmp-server location` command configures the system location string. By default, no system location string is set.

The `no snmp-server location` and `default snmp-server location` commands delete the location string by removing the `snmp-server location` command from the configuration.

**Command Mode**

Global Configuration

**Command Syntax**

```plaintext
snmp-server location node_locate
no snmp-server location
default snmp-server location
```

**Parameters**

- `node_locate` system location information (string).

**Example**

- These commands configure `lab-east` as the location string.

```
switch(config)#snmp-server location lab-east
```
The `snmp-server local-interface` command specifies the interface where SNMP originates informs and traps.

The `no snmp-server local-interface` and `default snmp-server local-interface` commands remove the inform or trap source assignment by removing the `snmp-server local-interface` command from running-config.

**Command Mode**

Global Configuration

**Command Syntax**

```
snmp-server local-interface INTERFACE
no snmp-server local-interface
default snmp-server local-interface
```

**Parameters**

- **INTERFACE**  
  Interface type and number. Values include:
  - `ethernet e_num`  Ethernet interface specified by `e_num`.
  - `loopback l_num`  Loopback interface specified by `l_num`.
  - `management m_num`  Management interface specified by `m_num`.
  - `port-channel p_num`  Port-Channel Interface specified by `p_num`.
  - `vlan v_num`  VLAN interface specified by `v_num`.
  - `vrf vrf_name`  The VRF in which SNMP is enabled. The keyword `default` specifies the default VRF.

**Example**

- This command configures the Ethernet 1 interface as the source of SNMP traps and informs.
  
  `switch(config)#snmp-server local-interface ethernet 1`
snmp-server user

The `snmp-server user` command adds a user to a Simple Network Management Protocol (SNMP) group or modifies an existing user's parameters.

To configure a user, the IP address or port number of the device where the user’s remote SNMP agent resides must be specified. A user's authentication come from the engine ID and the user's password. Remote user configuration commands fail if the remote engine ID is not configured first.

The `no snmp-server user` and `default snmp-server user` commands remove the user from an SNMP group by removing the user command from `running-config`.

Command Mode

Global Configuration

Command Syntax

```
snmp-server user user_name group_name [AGENT] VERSION [ENGINE] [SECURITY]
no snmp-server user user_name group_name [AGENT] VERSION
default snmp-server user user_name group_name [AGENT] VERSION
```

Parameters

- `user_name` name of user.
- `group_name` name of group to which user is being added.
- `AGENT` Options include:
  - <no parameter> local SNMP agent.
  - `remote addr [udp-port p_num]` remote SNMP agent location.
    - `addr` denotes the IP address; `p_num` denotes the udp port socket. (default port is 162).
- `VERSION` SNMP version; options include:
  - `v1` SNMPv1.
  - `v2c` SNMPv2c.
  - `v3` SNMPv3.
- `ENGINE` engine ID used to localize passwords. Available only if `VERSION` is `v3`.
  - <no parameter> Passwords localized by SNMP copy specified by `agent`.
  - localized `engineID` octet string of `engineID`.
- `SECURITY` Specifies authentication and encryption levels. Available only if `VERSION` is `v3`. Encryption is available only when authentication is configured.
  - <no parameter> no authentication or encryption.
  - `auth a_meth a_pass [priv e_meth e_pass]` authentication parameters.
    - `a_meth` authentication method: options are `md5` (HMAC-MD5-96) and `sha` (HMAC-SHA-96).
    - `a-pass` authentication string for users receiving packets.
    - `e_meth` encryption method: Options are `aes` (AES-128) and `des` (CBC-DES).
    - `e-pass` encryption string for the users sending packets.

Example

- This command configures the remote SNMP user `tech-1` to the `tech-sup` SNMP group.
  ```
  switch(config)#snmp-server user tech-1 tech-sup remote 10.1.1.2 v3
  ```
snmp-server view

The `snmp-server view` command defines a view.

An SNMP view defines a subset of objects from an MIB. Every SNMP access group specifies views, each associated with read or write access rights, to allow or limit the group’s access to MIB objects.

The `no snmp-server view` command deletes a view entry by removing the corresponding `snmp-server view` command from the `running-config`.

**Command Mode**
Global Configuration

**Command Syntax**
```
snmp-server view view_name family_name INCLUSION
no snmp-server view view_name [family_name]
```

**Parameters**
- `view_name` Label for the view record that the command updates. Other commands reference the view with this label.
- `family_name` Name of the MIB object or family.
  MIB objects and MIB subtrees can be identified by name or by the numbers representing the position of the object or subtree in the MIB hierarchy.
- `INCLUSION` Inclusion level of the specified family within the view. Options include:
  - `include` View includes the specified subtree.
  - `exclude` View excludes the specified subtree.

**Example**
- These commands create a view named `sys-view` that includes all objects in the `system` subtree except for those in `system.2`.

  switch(config)#snmp-server view sys-view system include
  switch(config)#snmp-server view sys-view system.2 exclude
snmp-server vrf

The `snmp-server vrf` command enables SNMP in the specified VRF. By default, SNMP is enabled only in default VRF.

- User-defined VRFs: The `no snmp-server vrf` command disables SNMP in the specified VRF by removing the corresponding `snmp-server vrf` command from the `running-config`.
- Default VRF: The `no snmp-server vrf` command disables SNMP in the VRF by adding a `no snmp-server vrf default` statement to `running-config`.

Command Mode
Global Configuration

Command Syntax
```
  snmp-server vrf vrf_name
  no snmp-server vrf vrf_name
  default snmp-server vrf vrf_name
```

Parameters
- `vrf_name` The VRF in which SNMP is enabled. The keyword `default` specifies the default VRF.

Guidelines
The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the default VRF. Enable or disable SNMP in a VRF with the `snmp-server vrf` command.

Example
- These commands disable SNMP in the default VRF, then enable it in the user-defined VRFs named “magenta” and “columbia.”
```
switch(config)#no snmp-server vrf default
switch(config)#snmp-server vrf magenta
switch(config)#snmp-server vrf columbia
switch(config)#
```
snmp trap link-change

The `snmp trap link-change` command enables Simple Network Management Protocol (SNMP) link-status trap generation on the configuration mode interface. The generation of link-status traps is enabled by default. If SNMP link-trap generation was previously disabled, this command removes the corresponding `no snmp link-status` statement from the configuration to re-enable link-trap generation.

The `no snmp trap link-change` command disables SNMP link trap generation on the configuration mode interface.

The `snmp trap link-change` and `default snmp trap link-change` commands restore the default behavior by removing the `no snmp trap link-change` command from `running-config`.

**Command Mode**
- Interface-Ethernet Configuration
- Interface-Loopback Configuration
- Interface-Management Configuration
- Interface-Port-channel Configuration
- Interface-VLAN Configuration
- Interface-VXLAN Configuration

**Command Syntax**
- `snmp trap link-change`
- `no snmp trap link-change`
- `default snmp trap link-change`

**Guidelines**

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the `default` VRF. Enable or disable SNMP in a VRF with the `snmp-server vrf` command.

**Example**

- This command disables SNMP link trap generation on the Ethernet 5 interface.
  ```
  switch(config-if-Et5)#no snmp trap link-change
  switch(config-if-Et5)#
  ```