Configuration Commands

SNMP System Commands

engineID

Syntax [**no**] **engineID** *engine-id*

Context config>system>snmp

Description This command sets the SNMP engineID to uniquely identify the SNMPv3 node. By default, the engineID is generated using information from the system backplane.

If SNMP engine ID is changed in the **config>system>snmp> engineID** *engine-id* context, the current configuration must be saved and a reboot must be executed. If not, the previously configured SNMP communities and logger trap-target notify communities will not be valid for the new engine ID.

Note: In conformance with IETF standard RFC 2274, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*, hashing algorithms which generate SNMPv3 MD5 or SHA security digest keys use the engineID. Changing the SNMP engineID invalidates all SNMPv3 MD5 and SHA security digest keys and may render the node unmanageable.

When a chassis is replaced, use the engine ID of the first system and configure it in the new system to preserve SNMPv3 security keys. This allows management stations to use their existing authentication keys for the new system.

Ensure that the engine IDs are not used on multiple systems. A management domain can only have one instance of each engineID.

The **no** form of the command reverts to the default setting.

Default The engine ID is system generated.

Parameters *engine-id* — An identifier from 10 to 64 hexadecimal digits (5 to 32 octet number), uniquely identifying this SNMPv3 node. This string is used to access this node from a remote host with SNMPv3.

general-port

Syntax **general-port** *port-number*
no general-port

Context config>system>snmp

Description This command configures the port number used by this node to receive SNMP request messages and to send replies. Note that SNMP notifications generated by the agent are sent from the port specified in the **config>log>snmp-trap-group>trap-target** CLI command.

The **no** form of the command reverts to the default value.

Default **161**

Parameters *port-number* — The port number used to send SNMP traffic other than traps.

Values 1 — 65535 (decimal)

packet-size

Syntax **packet-size** *bytes*
no packet-size

Context config>system>snmp

Description This command configures the maximum SNMP packet size generated by this node. If the packet size exceeds the MTU size of the egress interface the packet will be fragmented.

The **no** form of this command to revert to default.

Default **1500** bytes

Parameters *bytes* — The SNMP packet size in bytes.

Values 484 — 9216

snmp

Syntax **snmp**

Context config>system

Description This command creates the context to configure SNMP parameters.

streaming

Syntax **snmp**

Context config>system>snmp>streaming

Description This command enables the proprietary SNMP request/response bundling and TCP-based transport mechanism for optimizing network management of the router nodes. In higher latency networks, synchronizing router MIBs from network management via streaming takes less time than synchronizing via classic SNMP UDP requests. Streaming operates on TCP port 1491 and runs over IPv4 or IPv6.

shutdown

Syntax [**no**] **shutdown**

Context config>system>snmp>streaming

Description This command administratively disables proprietary SNMP request/response bundling and TCP-based transport mechanism for optimizing network management of the router nodes..

The **no** form of the command administratively re-enables SNMP request/response bundling and TCP-based transport mechanism.

Default **shutdown**

shutdown

Syntax [**no**] **shutdown**

Context config>system>snmp

Description This command administratively disables SNMP agent operations. System management can then only be performed using the command line interface (CLI). Shutting down SNMP does not remove or change configuration parameters other than the administrative state. This command does not prevent the agent from sending SNMP notifications to any configured SNMP trap destinations. SNMP trap destinations are configured under the **config>log>snmp-trap-group** context.

This command is automatically invoked in the event of a reboot when the processing of the configuration file fails to complete or when an SNMP persistent index file fails while the **bof persist on** command is enabled.

The **no** form of the command administratively enables SNMP which is the default state.

Default **no shutdown**

SNMP Security Commands

access group

Syntax **[no] access group** *group-name* **security-model** *security-model* **security-level** *security-level* [**context** *context-name* [**prefix-match**]] [**read** *view-name-1*] [**write** *view-name-2*] [**notify** *view-name-3*]

Context config>system>security>snmp

Description This command creates an association between a user group, a security model, and the views that the user group can access. Access parameters must be configured unless security is limited to the preconfigured access groups and views for SNMPv1 and SNMPv2. An access group is defined by a unique combination of the group name, security model and security level.

Access groups are used by the usm-community command.

Access must be configured unless security is limited to SNMPv1/SNMPv2c with community strings (see the **community** on page 308).

Default access group configurations cannot be modified or deleted.

To remove the user group with associated, security model(s), and security level(s), use:
no access group *group-name*

To remove a security model and security level combination from a group, use:
no access group *group-name* **security-model** {**snmpv1** | **snmpv2c** | **usm**} **security-level** {**no-auth-no-privacy** | **auth-no-privacy** | **privacy**}

Default **none**

Parameters *group-name* — Specify a unique group name up to 32 characters.

security-model {**snmpv1** | **snmpv2c** | **usm**} **—** Specifies the security model required to access the views configured in this node. A group can have multiple security models. For example, one view may only require SNMPv1/ SNMPv2c access while another view may require USM (SNMPv3) access rights.

security-level {**no-auth-no-priv** | **auth-no-priv** | **privacy**} **—** Specifies the required authentication and privacy levels to access the views configured in this node.

security-level no-auth-no-privacy — Specifies that no authentication and no privacy (encryption) is required. When configuring the user's authentication, select the **none** option.

security-level auth-no-privacy — Specifies that authentication is required but privacy (encryption) is not required. When this option is configured, both the **group** and the **user** must be configured for authentication.

security-level privacy — Specifies that both authentication and privacy (encryption) is required. When this option is configured, both the **group** and the user must be configured for **authentication**. The user must also be configured for **privacy**.

context *context-name* **—** Specifies a set of SNMP objects that are associated with the context-name.

The *context-name* is treated as either a full context-name string or a context name prefix depending on the keyword specified (**exact** or **prefix**).

prefix-match — Specifies the context name **prefix-match** keywords, **exact** or **prefix**.

The VPRN context names begin with a **vprn** prefix. The numerical value is associated with the service ID that the VPRN was created with and identifies the service in the service domain. For example, when a new VPRN service is created such as **config>service>vprn 2345 customer 1**, a VPRN with context name **vprn2345** is created.

The **exact** keywork specifies that an exact match between the context name and the prefix value is required. For example, when **context vprn2345 exact** is entered, matches for only **vprn2345** are considered.

The **prefix** keyword specifies that only a match between the prefix and the starting portion of context name is required. If only the **prefix** keyword is specified, simple wildcard processing is used. For example, when **context vprn prefix** is entered, all **vprn** contexts are matched.

Default **exact**

read *view-name* **—** Specifies the keyword and variable of the view to read the MIB objects.
This command must be configured for each view to which the group has read access.

Default **none**

write *view-name* **—** Specifies the keyword and variable of the view to configure the contents of the agent.
This command must be configured for each view to which the group has write access.

Values Up to 32 characters

notify *view-name* **—** specifies keyword and variable of the view to send a trap about MIB objects.
This command must be configured for each view to which the group has notify access.

Values none

attempts

Syntax **attempts** [*count*] [**time** *minutes1*] [**lockout** *minutes2*]
no attempts

Context config>system>security>snmp

Description This command configures a threshold value of unsuccessful SNMP connection attempts allowed in a specified time frame. The command parameters are used to counter denial of service (DOS) attacks through SNMP.

If the threshold is exceeded, the host is locked out for the lockout time period.

If multiple **attempts** commands are entered, each command overwrites the previously entered command.

The **no** form of the command resets the parameters to the default values.

Default **attempts 20 time 5 lockout 10** — 20 failed SNMP attempts allowed in a 5 minute period with a 10 minute lockout for the host if exceeded.

Parameters *count* — The number unsuccessful SNMP attempts allowed for the specified **time**.

Default **20**

Values 1 — 64

time *minutes1* — The period of time, in minutes, that a specified number of unsuccessful attempts can be made before the host is locked out.

Default **5**

Values 0 — 60

lockout *minutes2* — The lockout period in minutes where the host is not allowed to login. When the host exceeds the attempted count times in the specified time, then that host is locked out from any further login attempts for the configured time period.

Default **10**

Values 0 — 1440

community

Syntax **community** *community-string* [**hash** | **hash2**] *access-permissions* [**version** *SNMP-version*] [**src-access-list** *list-name*]
no community *community-string* [**hash** | **hash2**]

Context config>system>security>snmp

Description This command creates SNMP community strings for SNMPv1 and SNMPv2c access. This command is used in combination with the predefined access groups and views. To create custom access groups and views and associate them with SNMPv1 or SNMPv2c access use the **usm-community** command.

When configured, community implies a security model for SNMPv1 and SNMPv2c only. For SNMPv3 security, the **access group** command on page 306 must be configured.

The **no** form of the command removes a community string.

Default **none**

Parameters *community-string* — Configure the SNMPv1 and/or SNMPv2c community string.

Values community-string — 32 characters maximum
hash-key — 33 characters maximum
hash2-key — 96 characters maximum

hash | **hash2** — Configures the hashing scheme for *community-string*. **Hash** specifies that the key is entered in an encrypted form. If the hash parameter is not used, the key is assumed to be in a non-encrypted, clear text form. For security, all keys are stored in encrypted form. **Hash2** specifies that the key is entered in a more complex encrypted form.

access-permissions — Configures the access permissions for objects in the MIB.

r — Grants only read access to objects in the MIB, except security objects, using the internal "snmp-ro" access group and the "no-security" snmp view.

rw — Grants read and write access to all objects in the MIB, using the internal "snmp-rw" access group and the "no-security" snmp view.

rwa — Grants read and write access to all objects in the MIB, including security, using the internal "snmp-rwa" access group and the "iso" snmp view.

mgmt — Assigns a unique SMMP community string for SNMP access via the "management" routing instance. This community uses the internal "snmp-mgmt" access group and the "mgmt" snmp view.

vpls-mgmt — Assigns a unique SNMP community string for SNMP access via the "vpls-management" routing instance. This community uses the internal "snmp-vpls-mgmt" access group and "mgmt-view" snmp view.

version {**v1** | **v2c** | **both**} — Configures the scope of the community string to be for SNMPv1, SNMPv2c, or both SNMPv1 and SNMPv2c access.

Default **both**

list-name — Configures the **community** to reference a specific **src-access-list**, which will be used to validate the source IP address of all received SNMP requests that use this **community**. Multiple **community**, **usm-community**, or **vprn snmp community** instances can reference the same **src-access-list**.

mask

Syntax **mask** *mask-value* [**type** {**included** | **excluded**}]
no mask

Context config>system>security>snmp>view *view-name*

Description The mask value and the mask type, along with the *oid-value* configured in the **view** command, determines the access of each sub-identifier of an object identifier (MIB subtree) in the view.

Each bit in the mask corresponds to a sub-identifier position. For example, the most significant bit for the first sub-identifier, the next most significant bit for the second sub-identifier, and so on. If the bit position on the sub-identifier is available, it can be included or excluded.

For example, the MIB subtree that represents MIB-II is 1.3.6.1.2.1. The mask that catches all MIB-II would be 0xfc or 0b11111100.

Only a single mask may be configured per view and OID value combination. If more than one entry is configured, each subsequent entry overwrites the previous entry.

Per RFC 2575, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP),* each MIB view is defined by two sets of view subtrees, the included view subtrees, and the excluded view subtrees. Every such view subtree, both the included and the excluded ones, are defined in this table. To determine if a particular object instance is in a particular MIB view, compare the object instance's object identifier (OID) with each of the MIB view's active entries in this table. If none match, then the object instance is not in the MIB view. If one or more match, then the object instance is included in, or excluded from, the MIB view according to the value of vacmViewTreeFamilyType in the entry whose value of vacmViewTreeFamilySubtree has the most sub-identifiers.

The **no** form of this command removes the mask from the configuration.

Default none

Parameters *mask-value* — The mask value associated with the OID value determines whether the sub-identifiers are included or excluded from the view. (Default: all 1^s)

The mask can be entered either:

- In hex. For example, 0xfc.
- In binary. For example, 0b11111100.

Note: If the number of bits in the bit mask is less than the number of sub-identifiers in the MIB subtree, then the mask is extended with ones until the mask length matches the number of sub-identifiers in the MIB subtree.

type {included | excluded} — Specifies whether to include or exclude MIB subtree objects. *included* - All MIB subtree objects that are identified with a 1 in the mask are available in the view. (*Default: included*).

excluded - All MIB subtree objects that are identified with a 1 in the mask are denied access in the view. (*Default: included*).

Default **included**

snmp

Syntax **snmp**

Context config>system>security

Description This command creates the context to configure SNMPv1, SNMPv2, and SNMPv3 parameters.

src-access-list

Syntax **src-access-list** *list-name*
no src-access-list *list-name*

Context config>system>security>snmp

Description This command is used to identify a list of source IP addresses that can be used to validate SNMPv1 and SNMPv2c requests once the list is associated with one or more SNMPv1 and SNMPv2c communities.

An src-address-list referenced by one or more **community** instances is used to verify the source IP addresses of an SNMP request using the **community** regardless of which VPRN/VRF interface (or 'Base' interface) the request arrived on. For example, if an SNMP request arrives on an interface in vprn 100 but the request is referencing a **community**, then the source IP address in the packet would be validated against the src-address-list configured for the **community**. This occurs regardless of whether the request is destined to a VPRN interface address and the VPRN has SNMP access enabled, or the reques is destined to the base system address via GRT leaking. If the request's source

IP address does not match the *ip-address* of any of the **src-host**s contained in the list, then the request will be discarded and logged as an SNMP authentication failure.

Using src-access-list validation can have an impact on the time it takes for an SR OS node to reply to an SNMP request. It is recommended to keep the lists short, including only the addresses that are needed, and to place SNMP managers that send the highest volume of requests, such as the 5620 SAM, at the top of the list.

You can configure a maximum of 16 **src-access-list**s. Each **src-access-list** can contain a maximum of 16 **src-host**s.

The **no** form of this command removes the named src-access-list. You cannot remove an **src-access-list** that is referenced by one or more **community** instances.

Default none

Parameters *list-name* — Configures the name or key of the **src-access-list**. The *list-name* parameter must begin with a letter (a-z or A-Z).

src-host

Syntax **src-host** *host-name* **address** *ip-address*
no src-host *host-name*

Context config>system>security>snmp>src-access-list

Description This command is used to configure a source IP address entry that can be used to validate SNMPv1 and SNMPv2c requests.

The **no** form of this command removes the specified entry.

Default none

Parameters *host-name* — Configures the name of the **src-host** entry.

ip-address — Configures an allowed source address for SNMP requests. This can be an IPv4 or IPv6 address.

Values

ipv4-address	a.b.c.d
ipv6-address	x:x:x:x:x:x:x:x x:x:x:x:x:x:d.d.d.d x: [0..FFFF]H d: [0..255]D

usm-community

Syntax **usm-community** *community-string* **group** *group-name* [**src-access-lis**t *list-name*]
no usm-community *community-string*

Context config>system>security>snmp

Description This command is used to associate a community string with an SNMPv3 access group and its view. The access granted with a community string is restricted to the scope of the configured group.

Alcatel-Lucent's SR OS implementation of SNMP uses SNMPv3. In order to implement SNMPv1 and SNMPv2c configurations, several access groups are predefined. In order to implement SNMP with security features (Version 3), security models, security levels, and USM communities must be explicitly configured. Optionally, additional views which specify more specific OIDs (MIB objects in the subtree) can be configured.

The **no** form of this command removes a community string.

Default none

Parameters *community-string* — Configures the SNMPv1/SNMPv2c community string to determine the SNMPv3 access permissions to be used.

group — Specify the group that governs the access rights of this community string. This group must be configured first in the **config system security snmp access group** context. (*Default: none*)

list-name — Configures the **usm-community** to reference a specific **src-access-list** that will be used to validate the source IP address of all received SNMP requests that use this **usm-community**. Multiple **community**, **usm-community**, or **vprn snmp community** instances can reference the same **src-access-list**.

view

Syntax **view** *view-name* **subtree** *oid-value*
no view *view-name* [**subtree** *oid-value*]

Context config>system>security>snmp

Description This command configures a view. Views control the accessibility of a MIB object within the configured MIB view and subtree. Object identifiers (OIDs) uniquely identify MIB objects in the subtree. OIDs are organized hierarchically with specific values assigned by different organizations.

Once the subtree (OID) is identified, a mask can be created to select the portions of the subtree to be included or excluded for access using this particular view. See the **mask** command. The view(s) configured with this command can subsequently be used in read, write, and notify commands which are used to assign specific access group permissions to created views and assigned to particular access groups.

Multiple subtrees can be added or removed from a view name to tailor a view to the requirements of the user access group.

The **no view** *view-name* command removes a view and all subtrees.

The **no view** *view-name* **subtree** *oid-value* removes a sub-tree from the view name.

Default No views are defined.

Parameters *view-name* — Enter a 1 to 32 character view name. (Default: *none*)

oid-value — The object identifier (OID) value for the *view-name*. This value, for example, 1.3.6.1.6.3.11.2.1, combined with the mask and include and exclude statements, configures the access available in the view.

It is possible to have a view with different subtrees with their own masks and include and exclude statements. This allows for customizing visibility and write capabilities to specific user requirements.