

## Chapter 7

# Configuring SNMPv3

To configure SNMPv3, include the following statements at the [edit snmp v3] and [edit snmp] hierarchy levels:

```
[edit snmp]
engine-id {
    (local engine-id | use-fxp0-mac-address | use-default-ip-address);
}
view view-name; {
    oid object-identifier (include | exclude);
}

[edit snmp v3]
notify name {
    tag tag-name;
    type trap;
}
notify-filter profile-name {
    oid object-identifier (include | exclude);
}
snmp-community community-index {
    community-name community-name;
    security-name security-name;
    tag tag-name;
}
target-address target-address-name {
    address address;
    address-mask address-mask;
    port <port-number>;
    tag-list [ tag-list ];
    target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
        message-processing-model (v1 | v2c | v3);
        security-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

```

usm {
  local-engine {
    user username {
      authentication-md5 {
        authentication-password authentication-password;
      }
      authentication-none;
      authentication-sha {
        authentication-password authentication-password;
      }
      privacy-3des {
        privacy-password privacy-password;
      }
      privacy-aes128 {
        privacy-password privacy-password;
      }
      privacy-des {
        privacy-password privacy-password;
      }
      privacy-none;
    }
  }
}
vacm {
  access {
    group group-name {
      default-context-prefix {
        security-model (any | usm | v1 | v2c) {
          security-level (authentication | none | privacy) {
            notify-view view-name;
            read-view view-name;
            write-view view-name;
          }
        }
      }
    }
  }
}
security-to-group {
  security-model (usm | v1 | v2c) {
    security-name security-name {
      group group-name;
    }
  }
}
}

```

This section includes the following topics for configuring SNMPv3:

Minimum SNMPV3 Configuration on page 47

Configuring the Local Engine ID on page 48

Creating SNMPv3 Users on page 49

Configuring MIB Views on page 54

Defining Access Privileges for an SNMP Group on page 55

Configuring SNMP Traps on page 62

Configuring the SNMP Community on page 72

Example: SNMPv3 Configuration on page 75

## Minimum SNMPV3 Configuration

---

To configure the minimum requirements for SNMPv3, include the following statements at the [edit snmp v3] and [edit snmp] hierarchy levels:

```
[edit snmp]
view view-name {
    oid object-identifier (include | exclude);
}

[edit snmp v3]
notify name {
    tag tag-name;
}
notify-filter profile-name {
    oid object-identifier (include | exclude);
}
snmp-community community-index {
    security-name security-name;
}
target-address target-address-name {
    address address;
    target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
        message-processing-model (v1 | v2c | v3);
        security-model (usm | v1 | v2c);
        security-level (authentication | none | privacy);
        security-name security-name;
    }
}
```

```

}
usm {
local-engine {
    user username {
    }
}
}
vacm {
access {
    group group-name {
    default-context-prefix {
    security-model (any | usm | v1 | v2c) {
    security-level (authentication | none | privacy) {
    }
}
}
}
}
}
security-to-group {
    security-model (usm | v1 | v2c) {
    security-name security-name {
    group group-name;
    }
}
}
}
}

```



**NOTE:** You must configure at least one view (notify, read, or write) at the [edit snmp view-name] hierarchy level.

## Configuring the Local Engine ID

By default, the local engine ID uses the default IP address of the router. The local engine ID is the administratively unique identifier for the SNMPv3 engine. This statement is optional. To configure the local engine ID, include the `engine-id` statement at the [edit snmp] hierarchy level:

```

[edit snmp]
engine-id {
    (local engine-id-suffix | use-default-ip-address | use-mac-address);
}

```

`local engine-id-suffix`—The engine ID suffix is explicitly configured.

`use-default-ip-address`—The engine ID suffix is generated from the default IP address.

`use-mac-address`—The SNMP engine identifier is generated from the MAC address of the management interface on the routing platform.

The local engine ID is defined as the administratively unique identifier of an SNMPv3 engine, and is used for identification, not for addressing. There are two parts of an engine ID: prefix and suffix. The prefix is formatted according to the specifications defined in RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*. You can configure the suffix here.



**NOTE:** SNMPv3 authentication and encryption keys are generated based on the associated passwords and the engine ID. If you configure or change the engine ID, you must commit the new engine ID before you configure SNMPv3 users. Otherwise the keys generated from the configured passwords will be based on the previous engine ID.

For the engine ID, we recommend using the MAC address of fxp0.

## Creating SNMPv3 Users

For each SNMPv3 user, you can specify the username, authentication type, authentication password, privacy type, and privacy password. After the password is entered, a key based on the engine ID and password is generated and is written to the configuration file. After key generation, the password is deleted from this file.



**NOTE:** You can only configure one encryption type for each SNMPv3 user.

To create users, include the user statement at the [edit snmp v3 usm local-engine] hierarchy level:

```
[edit snmp v3 usm local-engine]
user username;
```

*username* is the name that identifies the SNMPv3 user.

To configure user authentication and encryption, include the following statements at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```
[edit snmp v3 usm local-engine user username]
authentication-md5 {
    authentication-password authentication-password;
}
authentication-sha {
    authentication-password authentication-password;
}
authentication-none;
privacy-aes128 {
    privacy-password privacy-password;
}
privacy-des {
    privacy-password privacy-password;
}
```

```

privacy-3des {
  privacy-password privacy-password;
}
privacy-none;

```

This section discusses the following topics:

Configuring the Authentication Type on page 50

Configuring the Encryption Type on page 51

Example: Creating SNMPv3 Users Configuration on page 53

### **Configuring the Authentication Type**

By default, the authentication type is set to none.

This section includes the following topics:

Configuring the MD5 Authentication on page 50

Configuring the SHA Authentication on page 51

Configuring No Authentication on page 51

#### **Configuring the MD5 Authentication**

To configure the message digest algorithm (MD5) as the authentication type for an SNMPv3 user, include the authentication-md5 statement at the [edit snmp v3 usm local-engine user *username*] hierarchy level:

```

[edit snmp v3 usm local-engine user username]
authentication-md5 {
  authentication-password authentication-password;
}

```

*authentication-password* is the password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

The password must be at least 8 characters long.

You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring the SHA Authentication

To configure the secure hash algorithm (SHA) as the authentication type for an SNMPv3 user, include the `authentication-sha` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[edit snmp v3 usm local-engine user username]  
authentication-sha {  
    authentication-password authentication-password;  
}
```

*authentication-password* is the password used to generate the key used for authentication.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

The password must be at least 8 characters long.

You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring No Authentication

To configure no authentication for an SNMPv3 user, include the `authentication-none` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[edit snmp v3 usm local-engine user username]  
authentication-none;
```

## Configuring the Encryption Type

By default, encryption is set to none.



**NOTE:** Before you configure encryption, you must configure the MD5 or SHA authentication.

Before you configure the `privacy-3des` and `privacy-aes128` statements, you must install the `jcrypto` package.

---

This section includes the following topics:

Configuring the Advanced Encryption Standard Algorithm on page 52

Configuring the Data Encryption Algorithm on page 52

Configuring Triple DES on page 52

Configuring No Encryption on page 53

### Configuring the Advanced Encryption Standard Algorithm

To configure the Advanced Encryption Standard (AES) algorithm for an SNMPv3 user, include the `privacy-aes128` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[snmp v3 usm local-engine user username]  
privacy-aes128 {  
    privacy-password privacy-password;  
}
```

*privacy-password* is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

The password must be at least 8 characters long.

You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring the Data Encryption Algorithm

To configure the data encryption algorithm (DES) for an SNMPv3 user, include the `privacy-des` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[edit snmp v3 usm local-engine user username]  
privacy-des {  
    privacy-password privacy-password;  
}
```

*privacy-password* is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

The password must be at least 8 characters long.

You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring Triple DES

To configure triple DES for an SNMPv3 user, include the `privacy-3des` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[snmp v3 usm local-engine user username]  
privacy-3des {  
    privacy-password privacy-password;  
}
```

*privacy-password* is the password used to generate the key used for encryption.

SNMPv3 has special requirements when you create plain-text passwords on a routing platform:

The password must be at least 8 characters long.

You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.

### Configuring No Encryption

To configure no encryption for an SNMPv3 user, include the `privacy-none` statement at the `[edit snmp v3 usm local-engine user username]` hierarchy level:

```
[edit snmp v3 usm local-engine user username]  
privacy-none;
```

### Example: Creating SNMPv3 Users Configuration

Define SNMPv3 users:

```
[edit]  
snmp {  
  v3 {  
    usm {  
      local-engine {  
        user user1 {  
          authentication-md5 {  
            authentication-password authentication-password;  
          }  
          privacy-des {  
            privacy-password password;  
          }  
        }  
        user user2 {  
          authentication-sha {  
            authentication-password authentication-password;  
          }  
          privacy-none;  
        }  
        user user3 {  
          authentication-none;  
          privacy-none;  
        }  
        user user4 {  
          authentication-md5 {  
            authentication-password authentication-password;  
          }  
          privacy-none {  
            privacy-password privacy-password;  
          }  
        }  
      }  
    }  
  }  
}
```



**Example: Ping Proxy MIB**

Restrict the ping mib community to read and write access of the ping MIB and jnxpingMIB only. Read or write access to any other MIB using this community is not allowed.

```
[edit snmp]
view ping-mib-view {
  oid 1.3.6.1.2.1.80 include;      #pingMIB
  oid jnxPingMIB include;        #jnxPingMIB
}
community ping-mib {
  authorization read-write;
  view ping-mib-view;
}
```

For more information on the ping MIB, see RFC 2925 and “Juniper Networks Enterprise-Specific MIBs” on page 101.

**Defining Access Privileges for an SNMP Group**

SNMPv3 uses the view-based access control model (VACM), which allows you to configure the access privileges granted to a group. Access is controlled by filtering the MIB objects available for a specific operation through a predefined view. You assign views to determine the objects that are visible for read, write, and notify operations for a particular group, using a particular context (only the default context is supported), a particular security model (v1,v2c, or usm), and particular security level (authenticated, privacy, or none). For information about how to configure views, see “Configuring MIB Views” on page 54.

You define user access to management information at the [edit snmp v3 vacm] hierarchy level. All access control within VACM operates on groups, which are collections of users as defined by USM, or community strings as defined in the SNMPv1 and SNMPv2c security models. The term *security-name* refers to these generic end users. The group to which a specific security name belongs is configured at the [edit snmp v3 vacm security-to-group] hierarchy level. That security name can be associated with a group defined at the [edit snmp v3 vacm security-to-group] hierarchy level. A group identifies a collection of SNMP users that share the same access policy. You then define the access privileges associated with a group at the [edit snmp v3 vacm access] hierarchy level. Access privileges are defined using views. For each group, you can apply different views depending on the SNMP operation; for example, reads (get, getNext, or getbulk) writes (set), notifications, the security level used (authentication, privacy, or none), and the security model (v1, v2c, or usm) used within an SNMP request.

You configure members of a group with the security-name statement. For v3 packets using USM, the security name is the same as the username. For SNMPv1 or SNMPv2c packets, the security name is determined based on the community string. Security names are specific to a security model. If you are also configuring VACM access policies for SNMPv1 or SNMPv2c packets, you must assign security names to groups for each security model (SNMPv1 or SNMPv2c) at the [edit snmp v3 vacm security-to-group] hierarchy level. You must also associate a security name with an SNMP community at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

To configure the access privileges for an SNMP group, include statements at the [edit snmp v3 vacm] hierarchy level:

```
[edit snmp v3 vacm]
access {
  group group-name {
    default-context-prefix {
      security-model (any | usm | v1 | v2c) {
        security-level (authentication | none | privacy) {
          notify-view view-name;
          read-view view-name;
          write-view view-name;
        }
      }
    }
  }
}
security-to-group {
  security-model (usm | v1 | v2c) {
    security-name security-name {
      group group-name;
    }
  }
}
```

This section describes the following topics related to defining privileges for an SNMP group:

Configuring the Access Privileges Granted to a Group on page 56

Assigning Security Names to Groups on page 60

### ***Configuring the Access Privileges Granted to a Group***

This section includes the following topics:

Configuring the Group on page 57

Configuring the Security Model on page 57

Configuring the Security Level on page 57

Associating MIB Views with an SNMP User Group on page 58

Example: Access Privilege Configuration on page 59

### Configuring the Group

To configure the access privileges granted to a group, include the group statement at the [edit snmp v3 vacm access] hierarchy level:

```
[edit snmp v3 vacm access]
group group-name;
```

*group-name* is a collection of SNMP users that belong to a common SNMP list that defines an access policy. Users belonging to a particular SNMP group inherit all access privileges granted to that group.

### Configuring the Security Model

To configure the security model, include the security-model statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix]
security-model (any | usm | v1 | v2c);
```

any—Any security model

usm—SNMPv3 security model

v1—SNMPv1 security model

v2c—SNMPv2c security model

### Configuring the Security Level

To configure the access privileges granted to packets with a particular security level, include the security-level statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c)] hierarchy level:

```
[edit snmp v3 access group group-name default-context-prefix security-model
(any | usm | v1 | v2c)]
security-level (authentication | none | privacy);
```

none—Provides no authentication and no encryption.

authentication—Provides authentication but no encryption.

privacy—Provides authentication and encryption.



**NOTE:** Access privileges are granted to all packets with a security level equal to or greater than that configured.

If you are configuring the SNMPv1 or SNMPv2c security model, use none as your security level. If you are configuring the SNMPv3 security model (USM), use the authentication, none, or privacy security level.

---

## Associating MIB Views with an SNMP User Group

MIB views define access privileges for members of a group. Separate views can be applied for each SNMP operation (read, write, and notify) within each security model (usm, v1, and v2c) and each security level (authentication, none, and privacy) supported by SNMP.

To associate MIB views with an SNMP user group, include the following statements at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-mode (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix
  security model (any | usm | v1 | v2c) security-level (authentication | none |
  privacy)]
  notify-view view-name;
  read-view view-name;
  write-view view-name;
```



**NOTE:** You must associate at least one view (notify, read, or write) at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level.

You must configure the MIB view at the [edit snmp view *view-name*] hierarchy level. For information about how to configure MIB views, see “Configuring MIB Views” on page 54.

This section describes the following topics related to this configuration:

Configuring the Notify View on page 58

Configuring the Read View on page 59

Configuring the Write View on page 59

### Configuring the Notify View

To associate notify access with an SNMP user group, include the notify-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix security-model
  (any | usm | v1 | v2c) security-level (authentication | none | privacy)]
  notify-view view-name;
```

*view-name* specifies the notify access, which is a list of notifications that can be sent to each user in an SNMP group. A view name cannot exceed 32 characters.

### Configuring the Read View

To associate a read view with an SNMP group, include the read-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix security-model
  (any | usm | v1 | v2c) security-level (authentication | none | privacy)]
read-view view-name;
```

*view-name* specifies read access for an SNMP user group. A view name cannot exceed 32 characters.

### Configuring the Write View

To associate a write view with an SNMP user group, include the write-view statement at the [edit snmp v3 vacm access group *group-name* default-context-prefix security-model (any | usm | v1 | v2c) security-level (authentication | none | privacy)] hierarchy level:

```
[edit snmp v3 vacm access group group-name default-context-prefix security-model
  (any | usm | v1 | v2c) security-level (authentication | none | privacy)]
write-view view-name;
```

*view-name* specifies write access for an SNMP user group. A view name cannot exceed 32 characters.

### Example: Access Privilege Configuration

Define access privileges:

```
[edit snmp v3]
access {
  group group1 {
    default-context-prefix {
      security-model usm {      #Define an SNMPv3 security model
        security-level privacy {
          notify-view nv1;
          read-view rv1;
          write-view wv1;
        }
      }
    }
  }
  group group2 {
    default-context-prefix {
      security-model usm {      #Define an SNMPv3 security model
        security-level authentication {
          read-view rv2;
          write-view wv2;
        }
      }
    }
  }
}
```

```

group group3 {
  default-context-prefix {
    security-model v1 { #Define an SNMPv1 security model
      security-level none {
        read-view rv3;
        write-view ww3;
      }
    }
  }
}

```

### Assigning Security Names to Groups

To assign security names to groups, include the following statements at the [edit snmp v3 vacm security-to-group] hierarchy level:

```

[edit snmp v3 vacm security-to-group]
security-model (usm | v1 | v2c) {
  security-name security-name {
    group group-name;
  }
}

```

This section includes the following topics:

Configuring the Security Model on page 60

Configuring the Security Name on page 61

Configuring the Group on page 61

Example: Security Group Configuration on page 62

### Configuring the Security Model

To configure the security model, include the security-model statement at the [edit snmp v3 vacm security-to-group] hierarchy level:

```

[edit snmp v3 vacm security-to-group]
security-model (usm | v1 | v2c);

```

usm—SNMPv3 security model

v1—SNMPv1 security model

v2c—SNMPv2 security model

## Configuring the Security Name

To associate a security name with a user or community string, include the security-name statement at the [edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c)] hierarchy level:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c)]
security-name security-name;
```

*security-name* is the username configured at the [edit snmp v3 usm local-engine user *username*] hierarchy level. For SNMPv1 and SNMPv2c, the security name is the community string configured at the [edit snmp v3 snmp-community *community-index*] hierarchy level. For information about configuring usernames, see “Creating SNMPv3 Users” on page 49. For information about configuring a community string, see “Configuring the SNMP Community” on page 72.



**NOTE:** The USM security name is separate from the SNMPv1 and SNMPv2c security name. If you are supporting SNMPv1 and SNMPv2c, you must configure separate security names within the security-to-group configuration at the [edit snmp v3 vacm access] hierarchy level.

---

## Configuring the Group

After you have created users, v1, or v2 security names, you associate them with a group. A group is a set of security names belonging to a particular security model. A group defines the access rights for all users belonging to it. Access rights define what SNMP objects can be read, written to, or created. A group also defines what notifications a user is allowed to receive.

If you already have a group that is configured with all of the view and access permissions that you want to give a user, you can add the user to that group. If you want to give a user view and access permissions that no other groups have, or if you do not have any groups configured, create a group and add the user to it.

To configure the access privileges granted to a group, include the group statement at the [edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c) security-name *security-name*] hierarchy level:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c) security-name
security-name]
group group-name;
```

*group-name* identifies a collection of SNMP security names that share the same access policy. For more information about groups, see “Defining Access Privileges for an SNMP Group” on page 55.

### Example: Security Group Configuration

Assign security names to groups:

```

vacm {
  security-to-group {
    security-model usm {
      security-name user1 {
        group group1;
      }
      security-name user2 {
        group group2;
      }
      security-name user3 {
        group group3;
      }
    }
  }
}

```

## Configuring SNMP Traps

In SNMPv3, traps are created by configuring the notify, target address, and target parameters. The target address defines a management application's address and parameters to be used in sending notifications. Target parameters define the message processing and security parameters that are used in sending notifications to a particular management target. SNMPv3 also lets you define SNMPv1 and SNMPv2c traps.



**NOTE:** When you configure SNMP traps, make sure your configured access privileges allow the traps to be sent. Access privileges are configured at the [edit snmp v3 vacm access] and [edit snmp v3 vacm security-to-group] hierarchy levels.

To configure SNMP traps, include the following statements at the [edit snmp v3] hierarchy level:

```

[edit snmp v3]
notify name {
  tag tag-name;
  type trap;
}
notify-filter name {
  oid object-identifier (include | exclude);
}
target-address target-address-name {
  address address;
  address-mask address-mask;
  port <port-number>;
  tag-list [ tag-list ];
  target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
  notify-filter profile-name;
  parameters {
    message-processing-model (v1 | v2c | v3);
    security-model (usm | v1 | v2c);
  }
}

```

```

        security-level (authentication | none | privacy);
        security-name security-name;
    }
}

```

This section includes the following topics:

Configuring the Trap Notification on page 63

Configuring the Trap Notification Filter on page 64

Configuring the Trap Target Address on page 65

Define the Trap Target Parameters on page 68

### Configuring the Trap Notification

The `notify` statement specifies the type of notification (trap) and contains a single tag. The tag defines a set of target addresses to receive a trap. The tag list contains one or more tags and is configured at the `[edit snmp v3 target-address target-address-name]` hierarchy level. If the tag list contains this tag, the JUNOS software sends a notification to all the target addresses associated with this tag.

To configure the trap notifications, include the `notify` statement at the `[edit snmp v3]` hierarchy level:

```

[edit snmp v3]
notify name {
    tag tag-name;
    type trap;
}

```

*name* is the name assigned to the notification.

*tag-name* defines the target addresses that are sent this notification. All the target-addresses that have this tag in their tag list are sent this notification. The *tag-name* is not included in the notification.

trap is the type of notification.



**NOTE:** Each notify entry name must be unique.

The JUNOS software supports only one type of notification: trap.

---

For information about how to configure the tag list, see “Configuring the Tag List” on page 66.

**Example: Trap Notification Configuration**

Specify three sets of destinations to send traps:

```
[edit snmp v3]
notify n1 {
  tag router1;
  type trap;
}
notify n2 {
  tag router2;
  type trap;
}
notify n3 {
  tag router3;
  type trap;
}
```

**Configuring the Trap Notification Filter**

SNMPv3 uses the notify filter to define which traps (or which objects from which traps) will be sent to the network management station (NMS). The trap notification filter limits the type of traps that are sent to the NMS.

Each object identifier represents a subtree of the MIB object hierarchy. The subtree can be represented either by a sequence of dotted integers (such as 1.3.6.1.2.1.2) or by its subtree name (such as interfaces).

To configure the trap notifications filter, include the notify-filter statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
notify-filter profile-name;
```

*profile-name* is the name assigned to the notify filter.

By default, the OID is set to include. To define access to traps (or objects from traps), include the oid statement at the [edit snmp v3 notify-filter *profile-name*] hierarchy level:

```
[edit snmp v3 notify-filter profile-name]
oid oid (include | exclude);
```

*oid* is the object identifier. All MIB objects represented by this statement have the specified OID as a prefix. It can be specified either by a sequence of dotted integers or by a subtree name.

include—Include the subtree of MIB objects represented by the specified OID.

exclude—Exclude the subtree of MIB objects represented by the specified OID.

## Configuring the Trap Target Address

The target address defines a management application's address and parameters that are used in sending notifications. It can also identify management stations that are allowed to use specific community strings. When you receive a packet with a recognized community string and a tag is associated with it, the JUNOS software looks up all the target addresses with this tag and verifies that the source address of this packet matches one of the configured target addresses.



**NOTE:** You must configure the address mask when you configure the SNMP community.

---

To specify where you want the traps to be sent and define what SNMPv1 and SNMP2vc packets are allowed, include the target-address statement at the [edit snmp v3] hierarchy level:

```
[edit snmp v3]
target-address target-address-name;
```

*target-address-name* is the string that identifies the target address.

To configure the target address properties, include the following statements at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
address address;
address-mask address-mask;
port <port-number>;
tag-list [ tag-list ];
target-parameters target-parameters-name;
```

This section includes the following topics:

Configuring the Address on page 65

Configuring the Address Mask on page 66

Configuring the Port on page 66

Configuring the Tag List on page 66

Applying Target Parameters on page 68

### Configuring the Address

To configure the address, include the address statement at the [edit snmp v3 target-address *target-address-name*] hierarchy level:

```
[edit snmp v3 target-address target-address-name]
address address;
```

*address* is the SNMP target address.

### Configuring the Address Mask

The address mask specifies a set of addresses that are allowed to use a community string and verifies the source addresses for a group of target addresses.

To configure the address mask, include the `address-mask` statement at the `[edit snmp v3 target-address target-address-name]` hierarchy level:

```
[edit snmp v3 target-address target-address-name]  
  address-mask address-mask;
```

*address-mask* combined with the `address` define a range of addresses. For information about how to configure the community string, see “Configuring the SNMP Community” on page 72.

### Configuring the Port

By default, the UDP port is set to 162. To configure the port, include the `port` statement at the `[edit snmp v3 target-address target-address-name]` hierarchy level:

```
[edit snmp v3 target-address target-address-name]  
  port <port-number>;
```

*port* is the SNMP target port number.

### Configuring the Tag List

Each `target-address` statement can have one or more tags configured in its tag list. Each tag can appear in more than one tag list. When a significant event occurs on the network device, the tag list identifies the targets to which a notification is sent.

To configure the tag list, include the `tag-list` statement at the `[edit snmp v3 target-address target-address-name]` hierarchy level:

```
[edit snmp v3 target-address target-address-name]  
  tag-list [ tag-list ];
```

*tag-list* specifies one or more tags.

For information about how to specify a tag at the `[edit snmp v3 notify notify-name]` hierarchy level, see “Configuring the Trap Notification” on page 63.

**Example: Configuring the Tag List**

In the following example, two tag entries (router1 and router2) are defined at the [edit snmp v3 notify *notify-name*] hierarchy level. When an event triggers a notification, the JUNOS software sends a trap to all target addresses that have router1 or router2 configured in their target-address tag list. This results in the first two targets getting one trap each, and the third target getting two traps.

```
[edit snmp v3]
notify n1 {
  tag router1;          # Identifies a set of target addresses
  type trap;           # Defines the type of notification
}
notify n2 {
  tag router2;
  type trap;
}
target-address ta1 {
  address 10.1.1.1
  address-mask 255.255.255.0;
  port 162;
  tag-list router1;
  target-parameters tp1;
}
target-address ta2 {
  address 10.1.1.2
  address-mask 255.255.255.0;
  port 162;
  tag-list router2;
  target-parameters tp2;
}
target-address ta3 {
  address 10.1.1.3
  address-mask 255.255.255.0;
  port 162;
  tag-list [router1 router2]; #Define multiple tags in the target address tag
  target-parameters tp3;    #list
}
```



**NOTE:** When you configure SNMP traps, make sure your configured access privileges allow the traps to be sent. Configure access privileges at the [edit snmp v3 vacm access] hierarchy level.

---

### Applying Target Parameters

The `target-parameters` statement at the `[edit snmp v3]` hierarchy level applies the target parameters configured at the `[edit snmp v3 target-parameters target-parameters-name]` hierarchy level.

To reference configured target parameters, include the `target-parameters` statement at the `[edit snmp v3 target-address target-address-name]` hierarchy level:

```
[edit snmp v3 target-address target-address-name]
target-parameters target-parameters-name;
```

*target-parameters-name* is the name associated with the message processing and security parameters that are used in sending notifications to a particular management target.

### Define the Trap Target Parameters

Target parameters define the message processing and security parameters that are used in sending notifications to a particular management target.

To define a set of target parameters, include the `target-parameters` statement at the `[edit snmp v3]` hierarchy level:

```
[edit snmp v3]
target-parameters target-parameters-name;
```

*target-parameters-name* is the name assigned to the target parameters.

To configure target parameter properties, include the following statements at the `[edit snmp v3 target-parameters target-parameter-name]` hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name]
notify-filter profile-name;
parameters {
    message-processing-model (v1 | v2c | V3);
    security-level (authentication | none | privacy);
    security-model (usm | v1 | v2c);
    security-name security-name;
}
```

This section includes the following topics:

Applying the Trap Notification Filter on page 69

Configuring the Target Parameters on page 69

### Applying the Trap Notification Filter

To apply the trap notification filter, include the `notify-filter` statement at the `[edit snmp v3 target-parameters target-parameter-name]` hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name]  
  notify-filter profile-name;
```

*profile-name* is the name of a configured notify filter. For information about configuring notify filters, see “Configuring the Trap Notification Filter” on page 64.

### Configuring the Target Parameters

To configure target parameter properties, include following statements at the `[edit snmp v3 target-parameters target-parameter-name parameters]` hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]  
  message-processing-model (v1 | v2c | v3);  
  security-model (usm | v1 | v2c);  
  security-level (authentication | none | privacy);  
  security-name security-name;
```

This section includes the following topics:

Configuring the Message Processing Model on page 69

Configuring the Security Model on page 70

Configuring the Security Level on page 70

Configuring the Security Name on page 70

Example: Trap Configuration on page 71

### Configuring the Message Processing Model

The Message Processing Model defines which version of SNMP to use when generating SNMP notifications. To configure the message processing model, include the `message-processing` statement at the `[edit snmp v3 target-parameters target-parameter-name parameters]` hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]  
  message-processing-model (v1 | v2c | v3);
```

v1—SNMPv1 message processing model

v2c—SNMPv2c message processing model

v3—SNMPV3 message processing model

**Configuring the Security Model**

To define the security model to use when generating SNMP notifications, include the security-model statement at the [edit snmp v3 target-parameters target-parameter-name parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-model (usm | v1 | v2c);
```

usm—SNMPv3 security model

v1—SNMPv1 security model

v2c—SNMPv2c security model

**Configuring the Security Level**

The security-level statement specifies whether the trap is authenticated and encrypted before it is sent.

To configure the security level to use when generating SNMP notifications, include the security-level statement at the [edit snmp v3 target-parameters target-parameter-name parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-level (authentication | none | privacy);
```

authentication—Provides authentication but no encryption.

none—No security. Provides no authentication and no encryption.

privacy—Provides authentication and encryption.



**NOTE:** If you are configuring the SNMPv1 or SNMPV2c security model, use none as your security level. If you are configuring the SNMPv3 (USM) security model, use the authentication or privacy security level.

---

**Configuring the Security Name**

To configure the security name to use when generating SNMP notifications, include the security-name statement at the [edit snmp v3 target-parameters target-parameter-name parameters] hierarchy level:

```
[edit snmp v3 target-parameters target-parameter-name parameters]
security-name security-name;
```

If the USM security model is used, the security-name identifies the user that is used when generating the notification. If the v1 or v2c security models are used, security-name identifies the SNMP community used when generating the notification.



**NOTE:** The access privileges for the group associated with a security name must allow this notification to be sent.

If you are using the v1 or v2 security models, the security name at the [edit snmp v3 vacm security-to-group] hierarchy level must match the security name at the [edit snmp v3 snmp-community *community-index*] hierarchy level.

### Example: Trap Configuration

Define traps:

```
[edit snmp v3]
notify n1 {
    tag router2;           # Identifies the target address
    type trap;           # Defines the type of notification
}
notify-filter nf1 {
    oid .1 include;      # Filters the type of traps that are sent to the NMS
}
target-address ta1 {    # Includes multiple addresses
    address 10.1.1.1;
    address-mask 255.255.255.0;
    port 162;
    tag-list router2;
    target-parameters tp1; # Applies configured target parameters
}
target-parameters tp1 { # Defines target parameters
    notify-filter nf1;
    parameters {
        message-processing-model v1;
        security-model v1';
        security-level none;
        security-name john;
    }
}
```

## Configuring the SNMP Community

---

The SNMP community defines the relationship between an SNMP server system and the client systems. This statement is optional.

To configure the SNMP community, include the `snmp-community` statement at the `[edit snmp v3]` hierarchy level:

```
[edit snmp v3]
snmp-community community-index;
```

*community-index* is the index for the SNMP community.

To configure the SNMP community properties, include the following statements at the `[edit snmp v3 snmp-community community-index]` hierarchy level:

```
[edit snmp v3 snmp-community community-index]
community-name community-name;
security-name security-name;
tag tag-name;
```

This section includes the following topics:

Configuring the Community Name on page 73

Configuring the Security Names on page 73

Configuring the Tag on page 73

Example: SNMP Community Configuration on page 74

## Configuring the Community Name

The community name defines the SNMP community. The SNMP community authorizes SNMPv1 or SNMPv2c clients. The access privileges associated with the configured security name define which MIB objects are available and the operations (read, write, or notify) allowed on those objects.

To configure the SNMP community name, include the `community-name` statement at the `[edit snmp v3 snmp-community community-index]` hierarchy level:

```
[edit snmp v3 snmp-community community-index]  
community-name community-name;
```

*community-name* is the community string for an SNMPv1 or SNMPv2c community. If unconfigured, it is the same as the community index.

If the community name contains spaces, enclose it in quotation marks (" ").



**NOTE:** Community names must be unique. You cannot configure the same community name at the `[edit snmp community]` and `[edit snmp v3 snmp-community community-index]` hierarchy levels.

The configured community name at the `[edit snmp v3 snmp-community community-index]` hierarchy level is encrypted. You cannot view the community name after you have configured it and committed your changes. In the CLI, the community name is concealed.

---

## Configuring the Security Names

To assign a community string to a security name, include the `security-name` statement at the `[edit snmp v3 snmp-community community-index]` hierarchy level:

```
[edit snmp v3 snmp-community community-index]  
security-name security-name;
```

*security-name* is used when performing access control. The `security-to-group` configuration at the `[edit snmp v3 vacm]` hierarchy level identifies the group.



**NOTE:** This security name must match the security name configured at the `[edit snmp v3 target-parameters target-parameters-name parameters]` hierarchy level when configuring traps.

---

## Configuring the Tag

To configure the tag, include the `tag` statement at the `[edit snmp v3 snmp-community community-index]` hierarchy level:

```
[edit snmp v3 snmp-community community-index]  
tag tag-name;
```

*tag-name* identifies the address of managers that are allowed to use a community string.

**Example: SNMP Community Configuration**

Define an SNMP community:

```
[edit snmp v3]
snmp-community index1 {
  community-name "$9$JOzi.QF/AtOz3"; # SECRET-DATA
  security-name john;
  tag router1;                        # Identifies managers that are allowed to use
}                                       # a community string
target-address ta1 {
  address 10.1.1.1;
  address-mask 255.255.255.0; # Defines the range of addresses
  port 162;
  tag-list router1;
  target-parameters tp1;             # Apply configured target parameters
}
```

## Example: SNMPv3 Configuration

---

Define an SNMPv3 configuration:

```
[edit snmp]
engine-id {
    use-fxp0-mac-address;
}
view jnxAlarms {
    oid 1.3.6.1.4.1.2636.3.4 include;
}
view interfaces {
    oid 1.3.6.1.2.1.2 include;
}
view ping-mib {
    oid 1.3.6.1.2.1.80 include;
}
[edit snmp v3]
notify n1 {
    tag router1;           # Identifies a set of target addresses
    type trap;            # Defines type of notification
}
notify n2 {
    tag host1;
    type trap;
}
notify-filter nf1 {
    oid .1 include;       # Defines which traps (or which objects for which
                        # that will be sent. In this case, include all traps.
}
notify-filter nf2 {
    oid 1.3.6.1.4.1 include; # Send enterprise-specific traps only
}
notify-filter nf3 {
    oid 1.3.6.1.2.1.1.5 include; # Send BGP traps only
}
snmp-community index1 {
    community-name "$9$JOzi.QF/AtOz3"; # SECRET-DATA
    security-name john; # Matches the security name at the target parameters
    tag host1; # Finds the addresses that are allow to be used with
} # this community string
target-address ta1 { # Associates the target address with the group
san-francisco;
    address 10.1.1.1;
    address-mask 255.255.255.0; # Defines the range of addresses
    port 162;
    tag-list router1;
    target-parameters tp1; # Apply configured target parameters
}

target-address ta2 {
    address 10.1.1.2;
    address-mask 255.255.255.0;
    port 162;
    tag-list host1;
    target-parameters tp2;
}
}
```

```

target-address ta3 {
  address 10.1.1.3;
  address-mask 255.255.255.0;
  port 162;
  tag-list [router1 host1];
  target-parameters tp3;
}
target-parameters tp1 {      # Define the target parameters
  notify-filter nf1;        # Specify which notify filter to apply
  parameters {
    message-processing-model v1;
    security-model v1';
    security-level none;
    security-name john;    # Matches the security name configured at the
                          # [edit snmp v3 snmp-community community-index]
                          #hierarchy level
  }
}
target-parameters tp2 {
  notify-filter nf2;
  parameters {
    message-processing-model v1;
    security-model v1';
    security-level none;
    security-name john;
  }
}
target-parameters tp3 {
  notify-filter nf3;
  parameters {
    message-processing-model v1;
    security-model v1';
    security-level none;
    security-name john;
  }
}
usm {
  local-engine {          #Define authentication and encryption for SNMP3 users.
    user user1 {
      authentication-md5 {
        authentication-password authentication-password;
      }
      privacy-des {
        privacy-password privacy-password;
      }
    }
    user user2 {
      authentication-sha {
        authentication-password authentication-password;
      }
      privacy-none;
    }
    user user3 {
      authentication-none;
      privacy-none;
    }
  }
}

```

```

user user4 {
  authentication-sha {
    authentication-password authentication-password;
  }
  privacy-aes128 {
    privacy-password privacy-password;
  }
}
user user5 {
  authentication-sha {
    authentication-password authentication-password;
  }
  privacy-none {
    privacy-password privacy-password;
  }
}
}
}
vacm {
  access {
    group san-francisco {      #Defines the access privileges for the group
      default-context-prefix { #san-francisco
        security-model v1 {
          security-level none {
            notify-view ping-mib;
            read-view interfaces;
            write-view jnxAlarms;
          }
        }
      }
    }
  }
}
security-to-group {
  security-model v1 {
    security-name john {      #Assigns john to the security group
      group san-francisco;    #san-francisco
    }
    security-name bob {
      group new-york;
    }
    security-name elizabeth {
      group chicago;
    }
  }
}
}
}

```

