JUNIPER
NETWORKS | Engineering
Simplicity

JUNOS

Junos® OS

Broadband Subscriber Management Getting Started Guide

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Broadband Subscriber Management Getting Started Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

2 Resource Monitoring for Subscriber Management and Services

3 Dynamic Profiles for Subscriber Management

4 | **Configuration Statements and Operational Commands**

About This Guide

Use this guide to get a high-level overview of subscriber management, including AAA support, class of service (CoS) to manage appropriate service levels; resource monitoring to enhance system stability by avoiding overuse of memory and CPU resources; and dynamic profiles to configure and provision subscribers and services.

1
CHAPTER

Subscriber Management Overview

IN THIS CHAPTER

Introduction to Subscriber Management

Subscriber Management Overview

The Juniper Networks Junos OS subscriber management feature provides subscriber access, authentication, and service creation, activation, and deactivation. You can also collect accounting information and statistics for subscriber service sessions.

The subscriber access feature supports both CLI and AAA-based configuration (such as RADIUS) for subscribers. Access and services start when the router receives a message from a client (such as a DHCP discover message). For RADIUS clients, RADIUS Access-Accept messages and Change-of-Authorization-Request (CoA-Request) messages can create, modify, and delete subscriber sessions as well as activate and deactivate service sessions. You can use CLI commands to create a dynamic profile, which acts as a template of user attributes.

A subscriber service is based on the combination of a defined dynamic profile and attributes configured through authentication. Dynamic profiles can include dynamic firewall filters, class-of-service (CoS) settings, and protocol (IGMP) settings that define access limits for subscribers and the scope of a service granted to the subscriber after access is obtained.

The subscriber access feature provides the following convenience and flexibility to service providers and subscribers:

- Service providers can separate services and access technology and eliminate unprofitable flat-rate billing. They gain the ability to efficiently design, manage, and deliver services that subscribers want, and then bill subscribers based on connect time, bandwidth, and the actual service used.

- Subscribers benefit by gaining access to multiple simultaneous services. Depending on the service provider configuration, subscribers can dynamically connect to and disconnect from various services when they want and for however long they want. Subscribers can be billed based on the service level and usage, rather than being charged a set rate regardless of usage.

To understand more about Subscriber Management Licensing , see Subscriber Access Licensing Overview and Configuring the Router to Strictly Enforce the Subscriber Scaling License. Please refer to the Juniper Licensing Guide for general information about License Management. Please refer to the product Data Sheets for details, or contact your Juniper Account Team or Juniper Partner.

Subscriber Access Terms and Acronyms

Table 1 on page 3 defines terms and acronyms that are used in this discussion of subscriber access.

Table 1: Subscriber Access Terms and Acronyms

Term	Definition
AAA method for subscriber authentication	The AAA method that uses authentication (for example, including RADIUS VSAs in the Access-Accept packet) to verify a subscriber and activate a service when the subscriber logs in.
Dynamic profile	A template that defines a set of characteristics that are combined with authorization attributes and are dynamically assigned to static interfaces to provide dynamic subscriber access and services for broadband applications.
RADIUS CoA method	The method that uses RADIUS CoA-Request messages and VSAs to activate a service for a subscriber that is already logged in.
Subscriber access technology	The technology used by a subscriber to access services (for example, DHCP).

> **NOTE**: When replacing a line card with active subscribers, it is recommended to log out all the subscribers before removing the line card.

AAA Service Framework and Subscriber Management Overview

You use AAA Service Framework for authentication, authorization, accounting, address assignment, and dynamic services request that the BNG uses for network access. The framework supports authentication and authorization through external servers, such as RADIUS. The framework also supports accounting and dynamic-request CoA and disconnect operations through external servers, and address assignment through a combination of local address-assignment pools and RADIUS.

The BNG interacts with external servers to determine how individual subscribers access the broadband network. The router also obtains information from external servers for the following:

- Methods used for authentication and accounting.

- How accounting statistics are collected and used.

- How dynamic requests are handled.

Class of Service and Subscriber Management Overview

Class of service (CoS) enables you to divide traffic into classes and offer various levels of throughput and acceptable packet loss when congestion occurs. CoS also provides the option of using differentiated services when best-effort traffic delivery is insufficient. You can also configure the services router to provide hierarchical scheduling for subscribers by dynamically adding or deleting queues when subscribers require services.

By using a dynamic profile, you can provide all subscribers in your network with default CoS parameters when they log in. For example, you can configure an access dynamic profile to specify that all subscribers receive a basic data service. If you use RADIUS variables in the dynamic profile, you can enable the service to be activated for those subscribers at login. You can also use variables to configure a service profile that enables subscribers to activate a service or upgrade to different services through RADIUS change-of-authorization (CoA) messages following initial login.

Configuring Subscriber Access

This topic provides a broad overview of some of the common configuration tasks for subscriber access and management. You can find detailed information in the following Junos OS User Guides:

- Broadband Subscriber Sessions User Guide

- Broadband Subscriber Access Protocols User Guide

- Broadband Subscriber Services User Guide

- Broadband Subscriber VLANs and Interfaces User Guide

- Broadband Subscriber Management Wholesale User Guide

To configure subscriber access:

1. Configure the client access protocol.

   - Configure DHCP local server.

     See *Understanding Differences Between Legacy DHCP and Extended DHCP*.

   - Configure DHCP relay.

     See *Extended DHCP Relay Agent Overview*.

   - Configure PPP.

     See Configuring Logical Interface Properties and Configuring PPPoE

2. Configure subscriber authentication, accounting, and addressing.

   a. Configure RADIUS:

      i. Specify the RADIUS servers.

         See *Specifying RADIUS Authentication and Accounting Servers for Subscriber Access*.

      ii. Specify any optional server attributes.

         See *Configuring Authentication and Accounting Parameters for Subscriber Access*.

      iii. (Optional) Configure the CoA feature for the RADIUS dynamic-request server to change or deactivate the service after login.

         See *Configuring RADIUS-Initiated Dynamic Request Support*.

      iv. Configure subscriber accounting (RADIUS accounting).

         See *Configuring Per-Subscriber Session Accounting*.

b. Configure addressing:

- See *Address-Assignment Pool Configuration Overview*.

3. Create and manage dynamic profiles for access and service.

  a. Configure a basic dynamic profile.

    See "Configuring a Basic Dynamic Profile" on page 52.

    See *Example: Minimum PPPoE Dynamic Profile*

  b. Configure a dynamic profile for access.

    See *Configuring Dynamic DHCP Client Access to a Multicast Network*.

  c. Configure a dynamic profile for services.

    See *Defining Various Levels of Services for DHCP Subscribers*.

  d. Configure a default subscriber service.

    See *Configuring a Default Subscriber Service*.

  e. Configure the static subscriber interfaces to be referenced in the dynamic profile.

  f. Specify the interface-name and unit variables that the router uses to dynamically associate to a subscriber's incoming interface.

  g. Add, modify, or delete dynamic profile values to manage subscriber access and services.

The router dynamically activates or modifies the subscriber service using the RADIUS configuration.

- When the subscriber logs in, the router dynamically activates the service.

  See *Dynamic Service Management with RADIUS*.

- If RADIUS CoA has been configured, the router can dynamically modify the service for a subscriber.

  See *RADIUS-Initiated Change of Authorization (CoA) Overview*.

Figure 1 on page 7 shows the configuration sequence you perform for DHCP-based subscriber access. It also shows the dynamic configuration performed by the router.

Figure 1: Subscriber Access Configuration Workflow
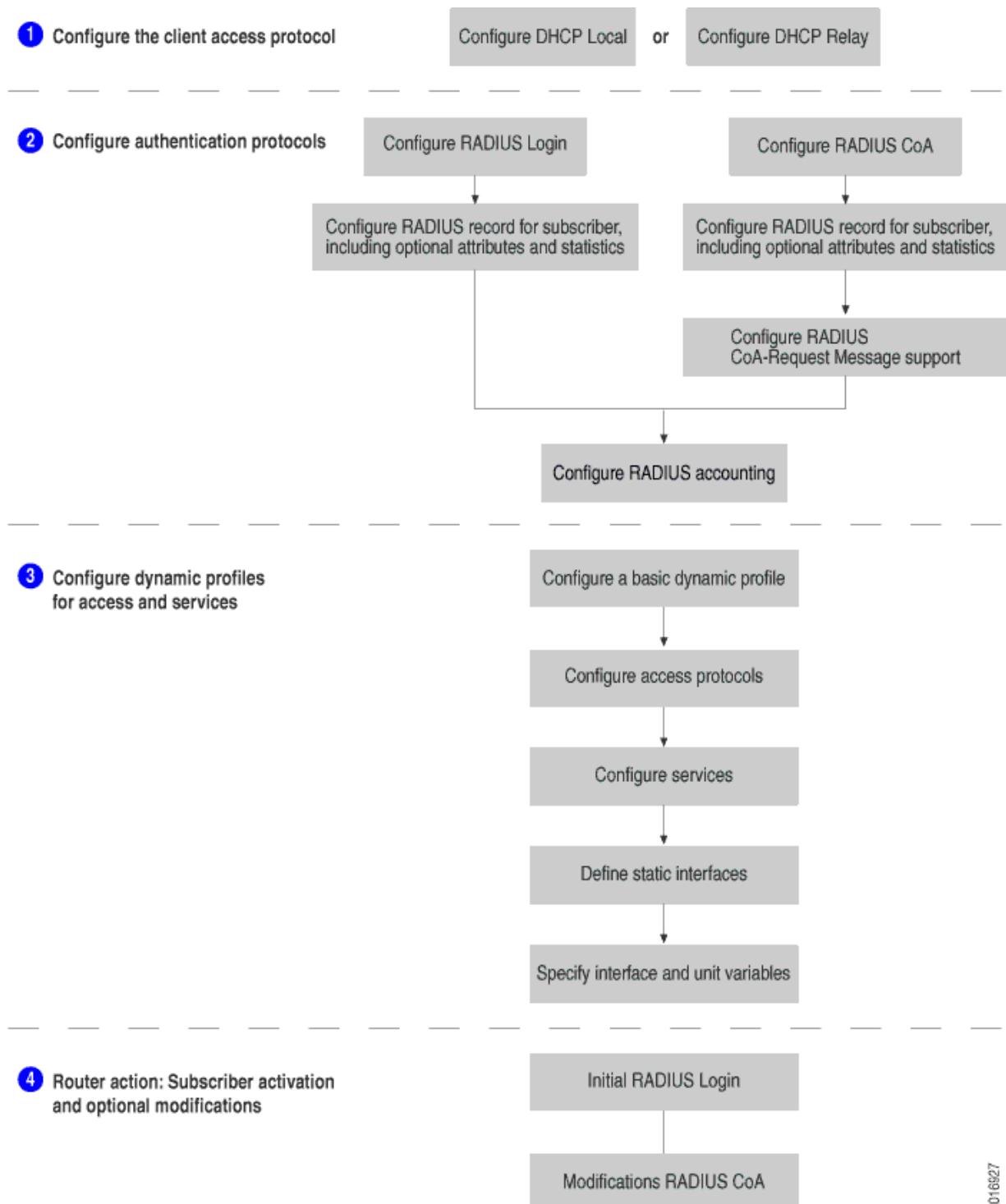
Subscriber Activation and Service Management in an Access Network

The subscriber access feature uses dynamic profiles to activate subscribers and manage services.

A dynamic profile is a set of characteristics, defined in a template, that the router uses to provide dynamic subscriber access and services.

By using dynamic profiles you can:

- Define access for your network

- Define different service levels for subscribers

- Preprovision services that you can activate later

Using AAA-based login (RADIUS-based login or RADIUS CoA) you can:

- Provide subscribers with dynamic activation and deactivation based on service selection

- Provide greater flexibility and efficient management for a large number of subscribers and services

Components of a Dynamic Profile

You can use dynamic profiles to define various router components for subscriber access.

These components include the following:

- Dynamic firewall filters—Includes input and output filters to enforce rules that define whether to permit or deny packets that are transmitting an interface on the router. To apply dynamic firewall filters to the subscriber interface, you configure static input and output firewall filters and reference those filters in dynamic profiles.

- Dynamic *Class of Service* (CoS)—Includes CoS values that define a service for a subscriber. For example, you can configure the shaping rate for traffic in a video service by referencing CoS statements in a dynamic profile.

- Dynamic signaling protocol—Includes dynamic IGMP configuration for host to router signaling for IPv4 to support IP multicasting.

Router Predefined Variables Used by Dynamic Profiles

The router contains many predefined variables. These variables enable dynamic association of certain interface-specific values to incoming subscriber requests. You must specify these predefined variables in certain statements within a dynamic profile. When a client accesses the router, the dynamic profile configuration replaces the predefined variable with the actual data from an incoming client data packet and configuration (local and RADIUS).

RELATED DOCUMENTATION

Subscriber Access Network Overview

Subscriber Access Operation Flow Using DHCP Relay

AAA Service Framework Overview

RADIUS-Initiated Change of Authorization (CoA) Overview

RADIUS-Initiated Disconnect Overview

CoS for Subscriber Access Overview

Default Services for DHCP Subscribers

Service Activation and Deactivation Using the CLI Instead of RADIUS

Dynamic Profiles for Subscriber Management | **47**

Dynamic Variables Overview | **58**

DHCP Subscriber Interface Overview

Junos OS and Junos OS Evolved Enhanced Subscriber Management

IN THIS SECTION

- Junos OS Enhanced Subscriber Management Overview | **10**
- Configuring Junos OS Enhanced Subscriber Management | **15**
- Verifying and Managing Junos OS Enhanced Subscriber Management | **21**

Junos OS Enhanced Subscriber Management Overview

Junos OS enhanced subscriber management is a next-generation broadband edge software architecture for wireline subscriber management. Enhanced subscriber management enables you to take advantage of increased scaling and performance for configuring and managing dynamic interfaces and services for subscriber management.

Enhanced subscriber management delivers optimized scaling and performance for the existing dynamic subscriber management feature set. Enhanced subscriber management provides feature parity with the legacy Junos OS subscriber management feature set, with certain exceptions. For a list of these feature exceptions, see the latest *Junos OS Release Notes for MX Series 5G Universal Routing Platforms* for your Junos OS software.

In order to use dynamic profiles to create and manage dynamic subscriber interfaces and services, you *must* explicitly configure and enable enhanced subscriber management. When enhanced subscriber management is enabled, it handles all subscriber-management control protocol traffic (DHCP, PPP, PPPoE, L2TP, and dynamic VLAN creation) to direct the creation of subscriber sessions and their associated dynamic interfaces.

If you are using only static network configurations and static services in a business edge environment, you do not need to enable enhanced subscriber management to configure these static topologies. When enhanced subscriber management is *not* enabled, the following client applications do not support the use of dynamic profiles, the creation of dynamic interfaces, or dynamic authentication services:

- Dynamic VLANs

- PPPoE

- PPP

- L2TP

- DHCP

From an operational perspective, enhanced subscriber management introduces only minimal changes to existing subscriber management configuration and verification procedures. For example, enhanced subscriber management consolidates several subscriber management components previously distributed across multiple processes into a single process. As a result, enhanced subscriber management can display consolidated information for subscriber management in a single show command.

Routing Services and Enhanced Subscriber Management

When client connections require additional routing protocols on dynamic interfaces, with the exception of IGMP and MLD, you must include routing services in the dynamic profile interface configuration. If you do not do so, then the pseudo logical interface is not created and routing services cannot be associated with the dynamic interface. The additional routing protocols cannot run on the dynamic subscriber interface.

You do not have to include routing services in the dynamic profile interface configuration when clients use only the standard access-internal routes, access routes, and framed routes. In other words, the routing service configuration is not required for simple client reachability purposes.

Routing service configuration is not required for IGMP or MLD, because these protocols are natively supported on enhanced subscriber management interfaces.

Distributed IGMP is not supported on subscriber management interfaces where routing-services are enabled.

When a dynamic profile containing the routing-services statement is instantiated, the router creates an enhanced subscriber management logical interface, also referred to as a pseudo logical interface, in the form demux0.*nnnnnnnnnn* (for example, demux0.3221225472). Any associated subscriber routes or routes learned from a routing protocol running on the enhanced subscriber management interface use this pseudo interface as the next-hop interface.

Besides enabling or disabling routing services for all subscribers on the dynamic interface, the routing-service statement enables you to use RADIUS to selectively enable or disable routing services for a specific subscriber during authentication if RADIUS returns the Routing-Services VSA (26-212) in the Access-Accept message.

This RADIUS capability requires you to specify the $junos-routing-services predefined variable in the dynamic profile. A VSA value of one enables routing services for the subscriber; a value of zero disables routing services for the subscriber. Any value other than zero or one is rejected. If you configure the variable and RADIUS does not return the VSA, then routing services are disabled for the subscriber.

You can specify the variable in the dynamic profiles for PPPoE subscribers, the underlying VLAN, or both. When you include the variable in the VLAN dynamic profile, then you must also configure the VLAN to be authenticated; otherwise, routing services remain disabled for the underlying interface and therefore also disabled for the PPPoE subscriber.

You can optionally create dedicated dynamic VLAN profiles to enable routing services for subscribers that require routing services. You can then create dedicated profiles for subscribers that do not need routing services by omitting the `routing-service` statement from the profile. In the following code sample, vlan-profile1 enables routing services; vlan-profile2 does not.

```
dynamic-profiles vlan-profile1 {
    interfaces $junos-interface-ifd-name {
        unit $junos-interface-unit {
            routing-service {
                enable;
            }
        }
    }
}
dynamic-profiles vlan-profile2 {
    interfaces $junos-interface-ifd-name {
        unit $junos-interface-unit {
        }
    }
}
```

The VLAN profile is chosen based on the VLAN range associated with the profile by the `ranges` statement at the `[edit interfaces]` hierarchy level. In the following code sample, vlan-profile1 uses VLAN IDs in the range 100 through 500; vlan-profile2 uses IDs in the range from 501 through 1000:

```
interfaces ge-0/0/1 {
    auto-configure;
        vlan-ranges {
            dynamic-profile vlan-profile1 {
                ranges 100-500;
            }
            dynamic-profile vlan-profile2 {
                ranges 501-1000;
            }
        }
    }
}
```

Address Resolution and Enhanced Subscriber Management

Junos OS supports address resolution with enhanced subscriber management. These enhancements affect only framed routes on dynamic VLANs. Framed routes associated with DHCP subscribers function the same as before this feature support.

- Dynamic layer 2 MAC address resolution is supported for non-host routes. Users deploying statically addressed IP clients or a mix of statically addressed IP clients and DHCP clients can use network (/29) framed routes or host (/32) framed routes to establish reachability. The /29 routes are coupled with the dynamic Layer 2 address associated with a host framed route. This supports business users who use routers with multiple public addresses behind CPE routers.

- This feature is enabled by default and requires no special configuration.

  In earlier releases, dynamic address resolution is supported only for host framed routes; network framed routes that resolve to an indirect next hop (such as a local gateway) are not supported.

- By default, an IPv4 framed host route is permanently associated with the source MAC address from the trigger packet that created the dynamic VLAN. You can override this behavior by enabling dynamic ARP to resolve the MAC address for the framed host routes with the `ipoe-dynamic-arp-enable` statement. ARP protocol exchange resolves the Layer 2 address for the framed route.

- The router can compare the source MAC address received in a gratuitous ARP request or reply packet with the value in the ARP cache. The router updates the cache with the received MAC address if it determines this address is different from the cache entry. Include the `receive-gratuitous-arp` statement to enable this feature.

  This capability is useful when an IP address moves to a different device or NIC and consequently is associated with a different MAC address than before the move. The new device broadcasts a gratuitous ARP reply that the router compares to the MAC address in the cache.

  When the statement is not included, the dynamic ARP times out. Before it is deleted from the cache, the router sends an ARP request for the target IP address. The client responds with the new MAC address, but a window may exist for the client where the MAC address does not match the NIC.

Control Plane Resiliency

Several enhancements are available to improve control plane resiliency and the reliability of session database replication and state synchronization between primary and standby Routing Engines.

- The primary and standby Routing Engines exchange detailed information about session database replication. This exchange enables the Routing Engines to better determine whether the replication is correct.

- You can configure the router to detect shared memory corruption and to automatically recover by rebooting the primary or standby Routing Engines, or both. In earlier releases, a manual reboot is

required to clear the corrupted shared memory; otherwise, it remains corrupted, causing processes that share the memory to generate core errors.

- You can monitor Routing Engine resiliency with the `show system subscriber-management resiliency` command. The `summary` version indicates whether the system is functioning normally or an unexpected condition exists. The `detail` and `extensive` versions provide detailed statistics about the session database in shared memory per Routing Engine.

Benefits of Enhanced Subscriber Management

- Optimizes scaling and performance for dynamic subscriber management features.

- Required for the creation and management of dynamic profiles, dynamic interfaces, and dynamic subscribers.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 2: Change History

Release	Description
18.4	Starting in Junos OS Release 18.4R1, BGP is supported over dynamic PPPoE interfaces for the IPv4 address family.
	Several enhancements are available for address resolution with enhanced subscriber management. These enhancements affect only framed routes on dynamic VLANs. Framed routes associated with DHCP subscribers function the same as before this feature support.
	The `routing-services` statement is deprecated and is replaced by the `routing-service` statement.
19.1	Starting in Junos OS Release 19.1R1 several enhancements are available to improve control plane resiliency and the reliability of session database replication and state synchronization between primary and standby Routing Engines.

Configuring Junos OS Enhanced Subscriber Management

Junos OS enhanced subscriber management is a next-generation broadband edge software architecture for wireline subscriber management. With enhanced subscriber management, you can take advantage of optimized scaling and performance for configuration and management of dynamic interfaces and services for subscriber management. It must be enabled to use dynamic profiles for creating and managing dynamic subscriber interfaces and services.

Enhanced subscriber management is supported on all MX Series 5G Universal Routing Platforms with Modular Port Concentrators (MPCs) installed. It is not supported for MS-DPCs. If the router has both MPC and MS-DPCs, a conflict between the MS-DPC and Enhanced Subscriber Management services can occur during ISSU that can result in an unscheduled shutdown of the device. To prevent this, do not run ISSU if the system has MS-DPCs installed, or only enable Enhanced Subscriber Management on device where no MS-DPCs are present.

Before you begin:

- Download and install Junos OS Release 15.1R4 or later.

  See *Migration, Upgrade, and Downgrade Instructions* in the *Junos OS Release 15.1R4 Release Notes*. You must reboot the router after the upgrade is validated and installed.

  > ⚠️ **CAUTION**: Because unified in-service software upgrade (unified ISSU) is not supported for subscriber management when you upgrade from a release that does not support enhanced subscriber management (Junos OS Release 14.2 or earlier) to a release that does support enhanced subscriber management (15.1R4 and later), all subscriber sessions and subscriber state are lost after the upgrade.

  > ⓘ **NOTE**: Starting in Junos OS Release 17.4R1, when enhanced IP network services and enhanced subscriber management are enabled, the amount of DRAM on the Routing Engine determines whether the subscriber management daemons on that Routing Engine all run in 32-bit mode or all run in 64-bit mode.
  >
  > - Less than 32 GB of RAM—32-bit mode
  >
  > - 32 GB or more of RAM—64-bit mode
  >
  > In releases earlier than Junos OS Release 17.4R1, only the subscriber management daemon, bbe-smgd, operates in either 32-bit or 64-bit mode depending on the DRAM.

> **NOTE**: All Routing Engines in the system must have the same amount of memory. This is universally true for subscriber management in all releases.

To configure Junos OS enhanced subscriber management for the first time:

1. Configure enhanced IP network services on the router.

   a. Specify that you want to configure chassis properties for the router.

   ```
   [edit]
   user@host# edit chassis
   ```

   b. Configure enhanced IP network services.

   ```
   [edit chassis]
   user@host# set network-services enhanced-ip
   ```

2. Enable enhanced subscriber management.

   a. Specify that you want to configure global services for the router.

   ```
   [edit]
   user@host# edit system services
   ```

   b. Enable enhanced subscriber management.

   ```
   [edit system services]
   user@host# set subscriber-management enable
   ```

3. Increase the amount of system shared memory available for enhanced subscriber management by limiting the maximum size of the configuration database.

   JUNOS OS processes map shared memory into their process space. For example, on MX240 through MX10003 routers, processes can map up to 1GB of shared memory. Enhanced subscriber management processes contend for shared memory with the JUNOS OS configuration database. By default, the configuration database tries to reserve 80 percent of the shared memory map, leaving insufficient space for subscriber management to function. The majority of configurations require much less than 300MB of mapped space. An appropriate database size enables subscriber management to operate and scale optimally. In some circumstances, you must configure a maximum size to increase the amount of shared memory available to subscriber management. In

other circumstances, we recommend that you allow the router to determine the appropriate size and that you do not configure a maximum.

> **NOTE**: Starting in Junos OS Release 20.1R1, a single memory map is used for both the Junos OS configuration database and the schema database, together. In lower releases, two separate maps are used.

- For MX5, MX10, MX40, MX80, and MX104 routers, you must always configure the maximum size to be no more than 100MB, regardless of the which Junos OS release is running and regardless of Routing Engine RAM.

```
user@host# set system configuration-database max-db-size 100M
```

- For MX240, MX480, MX960, MX2008, MX2010, MX2020, and MX10003 routers, the decision whether to explicitly configure a maximum size and what that size is, depends on the Junos OS release and the amount of RAM in the Routing Engines. Table 3 on page 17 lists conditions and the corresponding recommendations.

Table 3: Configuration Database Size for MX240, MX480, MX960, MX2008, MX2010, MX2020, and MX10003 Routers

Junos OS Release	Routing Engine RAM	Recommendation
Release 17.4R1 and earlier releases Release 18.1R1	Any	Configure maximum size to no more than 300MB.
Release 17.4R2 and higher 17.4x releases Release 18.1R2 and higher releases	Routing Engines have at least 32GB each	Allow the router to determine the appropriate size. Do not configure a maximum size.
Release 17.4R2 and higher 17.4x releases Release 18.1R2 and higher releases	Routing Engines have less than 32GB each	Configure maximum size to no more than 300MB.

4. (Optional) Enable dynamic ARP to resolve the MAC address for IPv4 framed host routes. Otherwise, an IPv4 framed host route is permanently associated with the source MAC address from the trigger packet that created the dynamic VLAN.

```
[edit system services subscriber-management overrides interfaces family inet]
user@host# set ipoe-dynamic-arp-enable
```

5. (Optional) Enable router to compare the source MAC address received in a gratuitous ARP request or reply packet with the value in the ARP cache and update the cache when this address is different from the cache entry.

```
[edit system services subscriber-management overrides interfaces family inet]
user@host# set receive-gratuitous-arp
```

6. (Optional) Force the output of the `show arp` command to display the IP address by the hostname of each device. This makes it easier to manage subscriber access if your subscriber configuration relies on the IP addresses of the devices.

```
[edit system services subscriber-management overrides interfaces family inet]
user@host# set force-show-arp-resolve
```

7. (Optional) Configure the router to automatically reboot the primary or standby Routing Engine, or both, when it detects that the shared memory has been corrupted, which is considered a catastrophic failure.

```
[edit system services subscriber-management overrides]
user@host# set event catastrophic-failure reboot master
user@host# set event catastrophic-failure reboot standby
```

8. (Optional) Enable traffic-accounting and rate-monitoring for the given interface:

```
[edit dynamic-profiles profile-name interfaces unit "$junos-interface-unit"]
user@host# set actual-transit-statistics
```

9. (Optional) Enable routing services for dynamic interfaces if you want to run routing protocols on those interfaces. This is not required for IGMP or MLD over dynamic interfaces.

```
[edit dynamic-profiles profile-name interfaces interface-name unit "$junos-interface-unit"]
user@host# set routing-services
```

Starting in Junos OS Release 18.4R1, the `routing-services` statement is deprecated and is replaced by the `routing-service` statement.

```
[edit dynamic-profiles profile-name interfaces interface-name unit "$junos-interface-unit"]
user@host# set routing-service
```

> **(i) NOTE**: When the underlying VLAN interface for PPPoE subscribers is created with a dynamic profile, you must enable routing services in both the PPPoE dynamic profile and the dynamic profile for the underlying VLAN. Otherwise the subscriber is not allowed to log in.

10. (Optional) Enable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR).

> **(i) NOTE**: For MX Series routers using enhanced subscriber management, the new backup Routing Engine (the former primary Routing Engine) will reboot when a graceful Routing Engine switchover is performed. This cold restart resynchronizes the backup Routing Engine state with that of the new primary Routing Engine, preventing discrepancies in state that might have occurred during the switchover.

> **(i) NOTE**: When graceful Routing Engine switchover is enabled for subscriber management, all Routing Engines in the router must have the same amount of DRAM for stable operation.

a. Enable GRES.

```
[edit chassis redundancy]
user@host# set graceful-switchover
```

> **(i) NOTE**: When GRES is enabled, you can either configure NSR or graceful restart. If you configure both, then committing the configuration fails.

b. Enable NSR (recommended if you enable GRES).

```
[edit routing-options]
user@host# set nonstop-routing
```

> *(i)* **NOTE**: To enable graceful restart:
>
> ```
> [edit routing-options]
> user@host# set graceful-restart
> ```

   c. Configure `commit` operations to automatically synchronize the configuration between the primary Routing Engine and the standby Routing Engine.

```
[edit system]
user@host# set commit synchronize
```

11. Commit the configuration.

After you commit the configuration, the software prompts you to initiate a system reboot.

12. Reboot the router software to enable enhanced subscriber management.

   a. Access operational mode.

```
[edit system]
user@host# exit
```

   b. Reboot the software.

```
user@host> request system reboot
```

Example

The following example shows a typical configuration to enable enhanced subscriber management.

```
[edit]
chassis {
    network-services {
        enhanced-ip;
    }
    redundancy {
        graceful-switchover;
    }
}
routing-options {
    nonstop-routing;
```

```
    }
system {
    commit synchronize;
    configuration-database {
        max-db-size 300M;
    }
    services {
        subscriber-management {
            enable;
        }
    }
}
```

> **NOTE**: If you have configured `graceful-restart`, then the following statement will be displayed in the example instead of `nonstop-routing`:
>
> ```
> routing-options {
>     graceful-restart;
> }
> ```

Verifying and Managing Junos OS Enhanced Subscriber Management

IN THIS SECTION

-
-

Purpose

View information about class of service (CoS), routing tables, active subscribers, and the subscriber database for Junos OS enhanced subscriber management.

Action

- To display dynamic subscriber interface associations for CoS classifers, rewrite rules, and scheduler maps:

```
user@host> show class-of-service interface interface-name
```

- To display CoS associations for a dynamic interface set:

```
user@host> show class-of-service interface-set interface-set-name
```

- To display the mapping of CoS schedulers to forwarding classes:

```
user@host> show class-of-service scheduler-map
```

- To display CoS traffic shaping and scheduling profiles:

```
user@host> show class-of-service traffic-control-profile
```

- To display the active entries in the routing table:

```
user@host> show route
```

- To display detailed information about active subscribers whose IP address matches the specified address:

```
user@host> show subscribers address address detail
```

- To display information about how routes are mapped to specific enhanced subscriber management interfaces:

```
user@host> show system subscriber-management route
```

- To display summary information for the subscriber management database:

```
user@host> show system subscriber-management summary
```

- To verify whether subscriber management daemons are running in 32-bit mode or 64-bit mode:

```
user@host> show system processes | grep libexec[36]
```

Starting in Junos OS Release 17.4 R1, when enhanced IP network services and enhanced subscriber management are enabled and a Routing Engine in the system has at least 32 GB of RAM, subscriber management daemons on that Routing Engine run in 64-bit mode. For consistent operation, all Routing Engines in the system must have the same amount of memory.

- 64-bit mode:

```
user@host> show system processes | grep libexec[36]
  PID TT  STAT    TIME  COMMAND
21149  -  S      0:01.37 /usr/libexec64/pfed -N
21195  -  S      0:00.46 /usr/libexec64/smid -N
21214  -  S      0:05.04 /usr/libexec64/bbe-smgd -b -N
21270  -  S      0:04.26 /usr/libexec64/authd -N
21498  -  S      0:02.37 /usr/libexec64/rpd -N
21504  -  S      0:00.84 /usr/libexec64/cosd
21539  -  S      0:00.37 /usr/libexec64/dfwd -N
21740  -  S      0:00.95 /usr/libexec64/jpppd -N
```

- 32-bit mode:

```
user@host> show system processes | grep libexec[36]
  PID TT  STAT    TIME  COMMAND
21149  -  S      0:01.37 /usr/libexec32/pfed -N
21195  -  S      0:00.46 /usr/libexec32/smid -N
21214  -  S      0:05.04 /usr/libexec32/bbe-smgd -b -N
21270  -  S      0:04.26 /usr/libexec32/authd -N
21498  -  S      0:02.37 /usr/libexec32/rpd -N
21504  -  S      0:00.84 /usr/libexec32/cosd
21539  -  S      0:00.37 /usr/libexec32/dfwd -N
21740  -  S      0:00.95 /usr/libexec32/jpppd -N
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.1R1	Starting in Junos OS Release 19.1, several enhancements are available to improve control plane resiliency and the reliability of session database replication and state synchronization between primary and standby Routing Engines.
18.4R1	Starting in Junos OS Release 18.4R1, the `routing-services` statement is deprecated and is replaced by the `routing-service` statement.
18.4R1	Starting in Junos OS Release 18.4R1, BGP is supported over dynamic PPPoE interfaces for the IPv4 address family.
18.4R1	Starting in Junos OS Release 18.4R1, several enhancements are available for address resolution with enhanced subscriber management. These enhancements affect only framed routes on dynamic VLANs. Framed routes associated with DHCP subscribers function the same as before this feature support.
18.4R1	Starting in Junos OS Release 18.4R1, the `routing-services` statement is deprecated and is replaced by the `routing-service` statement.
17.4R1	Starting in Junos OS Release 17.4R1, when enhanced IP network services and enhanced subscriber management are enabled, the amount of DRAM on the Routing Engine determines whether the subscriber management daemons on that Routing Engine all run in 32-bit mode or all run in 64-bit mode.
17.4R1	Starting in Junos OS Release 17.4 R1, when enhanced IP network services and enhanced subscriber management are enabled and a Routing Engine in the system has at least 32 GB of RAM, subscriber management daemons on that Routing Engine run in 64-bit mode.

Tracing Subscriber Management Database Events for Troubleshooting

IN THIS SECTION

The Junos OS trace feature tracks subscriber management database operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems. The operations and events are those associated with the smid process, which manages the subscriber management infrastructure.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `smid`. You can specify a different filename, but you cannot change the directory in which trace files are located.

2. When the trace log file `filename` reaches 128 kilobytes (KB), it is compressed and renamed `filename.0.gz`. Subsequent events are logged in a new file called `filename`, until it reaches capacity again. At this point, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

   You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the System Log Explorer.)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

The following topics describe how to configure all aspects of tracing subscriber management database operations:

Configuring the Subscriber Management Database Trace Log Filename

By default, the name of the file that records trace output for the subscriber management database is `smid`. You can specify a different name with the `file` option.

To configure the filename for subscriber management database tracing operations:

- Specify the name of the file used for the trace output.

```
[edit system services subscriber-management traceoptions]
user@host# set file smi_logfile_1
```

Configuring the Number and Size of Subscriber Management Database Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format `.`*number*`.gz`. The newest archived file is `.0.gz` and the oldest archived file is `.`*(maximum number)-1*`.gz`. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, *filename*, reaches 2 MB, *filename* is compressed and renamed *filename*`.0.gz`, and a new file called *filename* is created. When the new *filename* reaches 2 MB, *filename*`.0.gz` is renamed *filename*`.1.gz` and *filename* is compressed and renamed *filename*`.0.gz`. This process repeats until there are 20 trace files. Then the oldest file, *filename*`.19.gz`, is simply overwritten when the next oldest file, *filename*`.18.gz` is compressed and renamed to *filename*`.19.gz`.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit system services subscriber-management traceoptions]
user@host# set file smi_1 _logfile_1 files 20 size 2097152
```

Configuring Access to the Subscriber Management Database Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit system services subscriber-management traceoptions]
user@host# set file smi_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit system services subscriber-management traceoptions]
user@host# set file smi_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for Subscriber Management Database Messages to Be Logged

By default, the trace operation output includes all messages relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit system services subscriber-management traceoptions]
user@host# set file smi_1 _logfile_1 match regex
```

Configuring the Subscriber Management Database Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit system services subscriber-management traceoptions]
user@host# set flag flag
```

Tracing Subscriber Management Session Database Replication Events for Troubleshooting

IN THIS SECTION

The Junos OS trace feature tracks subscriber management session database replication operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems. The operations and events are those associated with the bdbrepd process, which syncs the subscriber management database between the primary and backup Routing Engines.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `bdbrepd`. You can specify a different filename, but you cannot change the directory in which trace files are located.

2. When the trace log file *filename* reaches 128 kilobytes (KB), it is compressed and renamed *filename*`.0.gz`. Subsequent events are logged in a new file called *filename*, until it reaches capacity again. At this point, *filename*`.0.gz` is renamed *filename*`.1.gz` and *filename* is compressed and renamed *filename*`.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

   You can optionally configure the maximum file size to be from 10 KB through 1 gigabyte (GB). You can also specify the number of trace files to be from 2 through 1000. (For more information about how log files are created, see the System Log Explorer.)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

The following topics describe how to configure all aspects of tracing subscriber management session database operations:

Configuring the Subscriber Management Session Database Replication Trace Log Filename

By default, the name of the file that records trace output for the subscriber management session database is `bdbrepd`. You can specify a different name with the `file` option.

To configure the filename for subscriber management database tracing operations:

- Specify the name of the file used for the trace output.

  ```
  [edit system services database-replication traceoptions]
  user@host# set file bdbrep_logfile_1
  ```

Configuring the Number and Size of Subscriber Management Session Database Replication Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format `.number.gz`. The newest archived file is `.0.gz` and the oldest archived file is `.(maximum number)-1.gz`. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, `filename`, reaches 2 MB, `filename` is compressed and renamed `filename.0.gz`, and a new file called `filename` is created. When the new `filename` reaches 2 MB, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until there are 20 trace files. Then the oldest file, `filename.19.gz`, is simply overwritten when the next oldest file, `filename.18.gz` is compressed and renamed to `filename.19.gz`.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit system services database-replication traceoptions]
user@host# set file bdbrep_1 _logfile_1 files 20 size 2097152
```

Configuring Access to the Subscriber Management Session Database Replication Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit system services database-replication traceoptions]
user@host# set file bdbrep_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit system services database-replication traceoptions]
user@host# set file bdbrep_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for Subscriber Management Session Database Replication Messages to Be Logged

By default, the trace operation output includes all messages relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit system services database-replication traceoptions]
user@host# set file bdbrep_1 _logfile_1 match regex
```

Configuring the Subscriber Management Session Database Replication Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit system services database-replication traceoptions]
user@host# set flag flag
```

2

CHAPTER

Resource Monitoring for Subscriber Management and Services

IN THIS CHAPTER

Resource Monitoring for Subscriber Management and Services

Resource Monitoring for Subscriber Management and Services Overview

Junos OS supports a resource monitoring capability using both the CLI and SNMP MIB queries. You can employ this utility to provision sufficient headroom (memory space limits for the application or virtual router) to ensure system stability, especially the health and operating efficiency of I-chip-based line cards and Trio-based FPCs on MX Series routers.

When memory utilization, either the ukernel memory or ASIC memory, reaches a certain threshold, the system operations compromise on the health and traffic-handling stability of the line card. Such a trade-off on system performance can be detrimental for supporting live traffic and protocols.

Besides the ability to configure a threshold to raise error logs when a specific threshold value of resources is exceeded, you can also monitor the threshold values and resource utilization using SNMP MIB queries.

The following sections describe the types of resource monitoring available with Junos OS:

Using Watermarks for Line-Card Resource Monitoring

You can configure watermark or checkpoint values for the line-card resources, such as ukern memory (heap), next-hop (NH) memory, and firewall or filter memory, to be uniform for both Trio-based and I-chip-based line cards. The NH memory watermark is applicable only for encapsulation memory (output WAN static RAM memory). Encapsulation memory is specific to I-chips and not applicable for Trio-based chips. When the configured watermark is exceeded, error logs are triggered. If the resource has been used above a certain threshold, warning system log messages are generated to notify about the threshold value having exceeded. Based on your network needs, you can then determine whether you want to terminate any existing subscribers and services to prevent the system from being overloaded and resulting in a breakdown.

This feature gathers input from each of the line cards and transfers this statistical detail to the Routing Engine process using a well-known internal port. This information is scanned by the daemon on the Routine Engine and using the shared memory space built into the session database, warning messages are generated for exceeded threshold conditions.

You can configure the following parameters at the `[edit system services]` hierarchy level to specify the high threshold value that is common for all the memory spaces or regions and the watermark values for the different memory blocks on DPCs and MPCs:

- High threshold value, exceeding which warnings or error logs are generated, for all the regions of memory, such as heap or ukernel, next hop and encapsulation, and firewall filter memory, by using the `resource-monitor high-threshold` *value* statement.

- Percentage of free memory space used for next hops to be monitored with a watermark value by using the `resource-monitor free-nh-memory-watermark` *percentage* statement.

- Percentage of free memory space used for ukernel or heap memory to be monitored with a watermark value by using the `resource-monitor free-heap-memory-watermark` *percentage* statement.

- Percentage of free memory space used for firewall and filter memory to be monitored with a watermark value by using the `resource-monitor free-fw-memory-watermark` *percentage* statement. This feature is enabled by default and you cannot disable it manually. The default value and the configured value of the watermark value for the percentage of free next-hop memory also applies to encapsulation memory.

The default watermark values for the percentage of free ukernel or heap memory, next-hop memory, and firewall filter memory are as follows:

- free-heap-memory-watermark—20

- free-nh-memory-watermark—20

- free-fw-memory-watermark—20

Throttling Subscriber Load Based on CoS Resource Capacity

Class of service (CoS) criteria are incorporated into the throttling decision for subscriber access. Information about the availability of CoS resources, namely queue capacity, is collected from the line cards. At subscriber login, assuming that the subscriber requires CoS resources, the line cards report the CoS queue utilization as a percent of resources that are bound to a scheduling hierarchy and are not free to be bound to a new scheduling hierarchy. The `high-cos-queue-threshold` statement at the `[edit system services]` hierarchy level can be set in the range of from 0 percent to 90 percent, separately for each FPC slot. When CoS queue utilization on a given FPC reaches that FPC's configured threshold level, further subscriber logins on that FPC are not allowed. This resource monitoring mechanism provides adjustable safety margins to proactively avoid completely exhausting each FPC's available CoS queue resources. See `high-cos-queue-threshold`.This feature is only available when you enable subscriber management. For more information on enabling subscriber management, see "Configuring Junos OS Enhanced Subscriber Management" on page 15.

Examining the Utilization of Memory Resource Regions Using show Commands

You can use the `show system resource-monitor fpc` command to monitor the utilization of memory resources on the Packet Forwarding Engines of an FPC. The filter memory denotes the filter counter memory used for firewall filter counters. The asterisk (*) displayed next to each of the memory regions denotes the ones for which the configured threshold is being currently exceeded. Resource monitoring commands display the configured values of watermark for memories for different line-card applications to be monitored. The displayed statistical metrics are based on the computation performed of the current memory utilization of the individual line cards. The ukern memory is generic across the different types of line cards and signifies the heap memory buffers. Because a line card or an FPC in a particular slot can contain multiple Packet Forwarding Engine complexes, the memory utilized on the application-specific integrated circuits (ASICs) are specific to a particular PFE complex. Owing to different architecture models for different variants of line cards supported, the ASIC-specific memory (next-hop and firewall or filter memory) utilization percentage can be interpreted differently.

Load Throttling to Reduce Processing Delays

The Routing Engine can use resource monitoring to assess and reduce the processing load on a line card's Packet Forwarding Engine. It is possible for the Routing Engine to send work at a higher rate than the Packet Forwarding Engine can process. This is sometimes called overdriving the line card or Packet

Forwarding Engine. When the work load on the Packet Forwarding Engine is too high, it can cause noticeable delays in packet processing.

Resource monitoring enables the Routing Engine assess the load by evaluating the round-trip delay for packets that it sends to the Packet Forwarding Engine. A longer round-trip time indicates a higher load and therefore a greater chance of processing delays on the Packet Forwarding Engine. When appropriate, the Routing Engine reduces the percentage of subscriber sessions (client and service) that are allowed to complete.

This capability is called load throttling or round-trip time load throttling. Throttling prevents the Routing Engine from over-driving line cards to the point that processing delays become visible to operators and back-office systems. It works like this:

1. To monitor delays, the Routing Engine sends an echo request message every second to the Packet Forwarding Engine on the line card. The echo request includes both a timestamp for when it is sent and a running sequence number. The message priority is best effort, to simulate the worst-case processing delay on the line card.

2. The Packet Forwarding Engine processes the echo request and responds with an echo reply. The message priority is high to minimize jitter when the Routing Engine processes the returned packet.

3. When the Routing Engine receives the echo reply, it calculates the round trip time as the time difference between the echo request timestamp and the time it receives the echo reply for that particular sequence number.

4. The Routing Engine compares the round-trip delay time to a default round-trip threshold value of 1 second. If the measured delay is longer than the threshold for three consecutive trips, the Routing Engine denies logins for a percentage of new subscribers, reducing the number of new client and service sessions that are established. This reduction is called throttling.

    An internal algorithm derives the throttling percentage based on the threshold and the round-trip time. This percentage varies based on the round-trip delay at that point in time.

    The Routing Engine increases the throttle—denies more subscriber logins—for each successive set of three delay measurements that all exceed the threshold.

5. When the measured delay is less than the threshold for three consecutive trips, the Routing Engine removes the throttle. This allows subscribers to log in freely.

> (i) **NOTE**: RTT load throttling applies on a per-line-card basis for Ethernet interfaces (ge, xe) and pseudowire interfaces (ps) as follows:
>
>   - For aggregated Ethernet interfaces, it applies to the set of line cards associated with the aggregated Ethernet bundle.

- For pseudowire interfaces with redundant logical tunnel (RLT), it applies to the set of line cards that are associated with the anchor point.

  In both cases, the Routing Engine considers the delay value that determines throttling to be the longest round-trip delay of all the line cards in the set.

Table 4 on page 37 shows how subscriber sessions are throttled on a line card over a period of 12 seconds when the round-trip delay is greater than the internal threshold. This example has the following assumptions:

- The internal delay threshold is 1 second.

- Delay measurements occur every second.

- The session creation rate is reduced by 10 percent after 3 consecutive round-trip delay measurements that are above the round-trip delay threshold. For as long as the threshold is exceeded, the throttling is increased every 3 measurements.

- If the measured delay drops and remains below the threshold for 3 consecutive round-trip delay measurements, the session rate returns to 100 percent.

NOTE: This example is simplified. Remember that the exact throttling percentage is determined dynamically and can vary second to second.

Table 4: Example Load Throttling Due to Round-trip Delay Time

Time	Round-trip Delay (ms)	Threshold Exceeded	Percentage of Sessions Allowed
1	850	No	100
2	900	No	100
3	995	No	100
4	1021	Yes Threshold exceeded count #1	100

Table 4: Example Load Throttling Due to Round-trip Delay Time *(Continued)*

Time	Round-trip Delay (ms)	Threshold Exceeded	Percentage of Sessions Allowed
5	1130	Yes Threshold exceeded count #2	100
6	1158	Yes Threshold exceeded count #3	90 Session rate reduced by 10 %
7	1127	Yes Threshold exceeded count #1	90 Session rate reduced by 10 %
8	1135	Yes Threshold exceeded count #2	90
9	1126	Yes Threshold exceeded count #3	80 Session rate reduced by 10 %
10	1000	No Threshold not exceeded count #1	80
11	991	No Threshold not exceeded count #2	80
12	998	No Threshold not exceeded count #3	100 Throttling removed

Resource load monitoring and round-trip time throttling is enabled by default. You can use either of the following statements to disable this feature:

- `no-load-throttle` at the `[edit system services resource-monitor]` hierarchy level

- `no-throttle` at the `[edit system services resource-monitor]` hierarchy level

If you disable the feature and the Packet Forwarding Engine becomes too busy, new subscribers can log in and go active, but no traffic flows for a period of time. This delay in traffic processing might become noticeable.

You can use the following command to confirm whether the load throttling feature is enabled and see various aspects of the feature in action. The bolded fields are particularly useful.

```
user@host> show system resource-monitor summary
Resource Usage Summary
Throttle                     : Enabled
Load Throttle                : Enabled   /*RTT load throttling is enabled*/
Heap Mem Threshold           : 70  %
IFL Counter Threshold        : 95  %
Round Trip Delay Threshold(ms) : 1000   /*RTT throttle value*/
Filter Counter Threshold     : 100 %
Expansion Threshold          : 95  %
CoS Queue Threshold          : 100 %
MFS threshold                : 70  %       Used : 0


Slot # 0
    Client allowed                : Yes
    Service allowed               : Yes
    Heap memory used              : 339204848      In % : 18
    Average Round-trip Delay(ms)    : 103  (30  )      Round-trip Delay(ms) : 103   /*RTT delay
and average delay, the 30 in parentheses means that the average is for last 30 secs*/
    MAX session rate allowed(%)     : 100
    Client denied                 : 1524   /*The number of new subscribers have been denied*/
    Service Denied                : 0
    Performance Denial Client     : 1524  <--
    Performance Denial Service    : 0
    IFL Denied                    : 0
```

Limiting Subscribers with Resource Monitor

Starting in Junos OS Release 17.3R1, you can also use resource monitoring to directly limit the number of subscribers supported per hardware element. You can specify the maximum number of subscribers that can be logged in per chassis, line card (MPC), MIC, or port. You can set the limit to subscribers of only one client type (DHCP, L2TP, or PPPoE) or to subscribers of any client type.

This feature ensures that the number of subscribers logged in per hardware element does not exceed the number that your network can serve with stability at the desired service bandwidth. When the limit is reached for a hardware element, new subscriber logins are denied on that element until the number of

subscribers drops below the configured limit. New subscribers over the limit can connect to another hardware element in the same broadcast domain. When you configure the limit on one or more legs of an aggregated Ethernet interface, login is denied if the subscriber count exceeds the value on any of the legs.

Limiting subscribers this way distributes the load among hardware elements, but it does not provide any sort of load balancing. This feature can also help you map capacity in your network and determine what hardware resources you need to expand that capacity. For example, if you provide a service that needs a particular amount of memory and know how many subscribers you can service with a given set of hardware, you can determine how much memory you need. Or if you want to add a service with more memory per subscriber, you can calculate the additional amount that you need, compare it to your available memory, and determine whether you need to provision new ports, MICs, MPCs, or routers to handle the new service.

Change History on Resource Monitoring for Subscriber Management and Services

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 5: Change History on Resource Monitoring for Subscriber Management and Services

Release	Description
17.3	Starting in Junos OS Release 17.3R1, you can also use resource monitoring to directly limit the number of subscribers supported per hardware element.
17.4	Starting in Junos OS Release 17.4R1, class of service (CoS) criteria are incorporated into the throttling decision for subscriber access.
19.4	Starting in Junos OS Release 19.4R1, you can specify a value of 0 to prevent any subscriber from being throttled by queue-based throttling.

Platform-Specific Resource Monitoring for Subscriber Management and Services Behavior

Platform	Difference
MX240, MX480, and MX960 routers with MPC2E legacy, MPC2E-NG, MPC3E-NG, MPC5E, and MPC7E line cards	CoS resource monitoring feature bases admission decisions only on queues is supported for the hardware. Other CoS resources are not part of this criteria. This feature does not support throttling for subscribers arriving on pseudo-wire, logical tunnel, or redundant logical tunnel devices.
MX80, MX104 routers	Support resource monitoring configuration .

(Continued)

Platform	Difference
MX240, MX480, MX960, MX2010, and MX2020 routers	The following line cards support resource monitoring on MX240, MX480, MX960, MX2010, and MX2020 routers: • MX-MPC1-3D • MX-MPC1-3D-Q • MX-MPC2-3D • MX-MPC2-3D-Q • MX-MPC2-3D-EQ • MPC-3D-16XGE-SFPP • MPC3E • MPC3E-3D-NG • MPC4E-3D-2CGE-8XGE • MPC4E-3D-32XGE • MPC5EQ-40G10G • MPC5EQ-100G10G • MPC5E-100G10G • MPC5E-40G10G • MPC10E-10C-MRATE • MPC10E-15C-MRATE • MX2K-MPC6E • MX2K-MPC11E • DPCE • MS-DPC

(Continued)

Platform	Difference
	• MX Series Flexible PIC Concentrators (MX-FPCs) • NG-MPC3E

Limiting Subscribers by Client Type and Hardware Element with Resource Monitor

In addition to using resource monitoring to monitor and manage system memory usage, you can use it to directly limit the number of subscribers supported per hardware element: chassis, line card (MPC), MIC, and port. You can specify the maximum number of subscribers that can be logged in to each of those elements. You apply the limit to subscribers of only one client type (DHCP, L2TP, or PPPoE) or to subscribers of any of these client types. In the latter case, the limit applies to the sum of sessions for all three client types.

Subscriber limiting can ensure that the number of subscribers logged in per hardware element does not exceed the number that your network can serve with stability at the desired service bandwidth. When the limit is reached for a hardware element, new subscriber logins are denied on that element until the number of subscribers drops below the configured limit. New subscribers over the limit connect to another hardware element in the same broadcast domain. When you configure the limit on one or more legs of an aggregated Ethernet interface, login is denied if the subscriber count exceeds the value on any of the legs.

Limiting subscribers this way distributes the load among hardware elements, but it does not provide any sort of load balancing. This feature can also help you map capacity in your network and determine what hardware resources you need to expand that capacity. For example, if you provide a service at a particular bandwidth and know how many subscribers you can service with a given set of hardware, you can determine how much bandwidth you need. Or if you want to add a service with more bandwidth per subscriber, you can calculate the additional bandwidth that you need, compare it to your available bandwidth, and determine whether you need to provision new ports, MICs, MPCs, or routers to handle the new service.

> **NOTE**: The CLI uses the terms `fpc` and `pic`. For this feature, `fpc` corresponds to MPC and `pic` corresponds to MIC.

To place a limit on the maximum number of subscribers allowed for a hardware element:

1. Configure the client type for the subscribers.

   ```
   [edit system services resource-monitor subscribers-limit]
   user@host# edit client-type type
   ```

2. (Optional) Configure a subscriber limit on the chassis.

   ```
   [edit system services resource-monitor subscribers-limit client-type type]
   user@host# set chassis limit limit
   ```

3. (Optional) Configure a subscriber limit on an MPC.

   ```
   [edit system services resource-monitor subscribers-limit client-type type]
   user@host# edit fpc slot-number
   [edit system services resource-monitor subscribers-limit client-type type fpc slot-number]
   user@host# set limit limit
   ```

4. (Optional) Configure a subscriber limit on a MIC.

   ```
   [edit system services resource-monitor subscribers-limit client-type type fpc slot-number]
   user@host# edit pic number
   [edit system services resource-monitor subscribers-limit client-type type fpc slot-number pic number]
   user@host# set limit limit
   ```

5. (Optional) Configure a subscriber limit on a port.

   ```
   [edit system services resource-monitor subscribers-limit client-type type fpc slot-number pic number]
   user@host# set port number limit limit
   ```

For example, the following configuration sets chassis and MPC limits for PPPoE subscribers:

```
[edit system services resource-monitor subscribers-limit]
user@host# edit client-type pppoe
[edit system services resource-monitor subscribers-limit client-type pppoe]
user@host# set chassis limit 112000
user@host# set fpc 0 limit 28000
user@host# set fpc 1 limit 28000
user@host# set fpc 2 limit 28000
user@host# set fpc 3 limit 28000
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, class of service (CoS) criteria are incorporated into the throttling decision for subscriber access.
17.3R1	Starting in Junos OS Release 17.3R1, you can also use resource monitoring to directly limit the number of subscribers supported per hardware element.

RELATED DOCUMENTATION

Diagnosing and Debugging System Performance by Configuring Memory Resource Usage Monitoring on MX Series Routers

Resource Monitoring Usage Computation Overview

3
CHAPTER

Dynamic Profiles for Subscriber Management

IN THIS CHAPTER

Dynamic Profiles for Subscriber Management

Dynamic Profiles Overview

A dynamic profile is a set of characteristics that acts as a kind of template that enables you to create, update, or remove a configuration that you can use to provide dynamic subscriber access and services for broadband applications. Using these profiles enables you to consolidate all of the common attributes of a client or a group of clients and apply the attributes or dynamically created objects simultaneously. After profiles are created, they reside on the router in a profile library.

You can manage subscribers dynamically with two kinds of dynamic profiles: *client profiles* and *service profiles*. Both profile types are configured at the `[edit dynamic-profiles]` hierarchy level and are independent of each other. Whether you use dynamic service profiles in addition to your dynamic client profiles depends on how you support differentiation among subscribers and how you package your subscriber services.

A dynamic client profile can also correctly be referred to as a dynamic subscriber profile.

Dynamic client profiles are sometimes referred to as client access profiles. However, they are different from the access profiles configured at the `[edit access profile profile-name]` hierarchy level. Access profiles

are used to configure authentication, accounting, and authorization parameters for subscriber access, some session attributes, and client-specific properties for L2TP and PPP sessions. Access profiles are applied at various configuration levels with the access-profile statement.

Dynamic Client Profiles and Dynamic Service Profiles

The major differences between dynamic client and dynamic service profiles are the following:

- A dynamic client profile is provisioned and applied to the client application configuration; for example, DHCP, DHCPv6, L2TP LNS, PPPoE, static subscribers, and VLANs. The contents of the profile are applied to the logical interface for the subscriber session. Most often, dynamic client profiles enable the dynamic instantiation of logical interfaces to which the profile is applied, but client profiles can also be applied to static subscriber logical interfaces.

  A dynamic client profile can include any of the stanzas under `[edit dynamic-profiles` *`profile-name`*`]`, except for `variables` *`variable-name`*.

- Dynamic service profiles include only service-related configurations, which are a subset of the configurations available in dynamic client profiles. They do not include other configuration attributes for a subscriber session. You cannot use a service profile to create or modify a logical interface. A dynamic service profile functions as a supplement to dynamic client profiles that is used after the creation of logical interfaces.

  A dynamic service profile can include the following stanzas under `[edit dynamic-profiles` *`profile-name`*`]`: `class-of-service`, `firewall`, `protocols`, `services`, and `variables`.

Dynamic client profiles and dynamic service profiles also differ in the types of variables they can use:

- Dynamic client profiles can include predefined-variable-defaults, which define default values for Juniper Networks predefined variables that are included in the profile. The default values in the profile are used when RADIUS does not return a value for the variable. See "Dynamic Variables Overview" on page 58 and "Configuring Default Values for Predefined Variables in a Dynamic Profile" on page 116 for information about predefined variables.

- Dynamic service profiles can include user-defined variables that act like parameters in a function call. The variable values can be provided by the RADIUS server to support more specialized customization per subscriber. You can also set default values for these variables to be used when RADIUS does not provide the value. See "User-Defined Variables in Dynamic Profiles" on page 119 for information about user-defined variables.

- Dynamic client profiles do not include user-defined variables. Dynamic service profiles do not include predefined-variable-defaults.

Table 6 on page 49 lists the types of variables supported by access profiles and service profiles.

Table 6: Types of Variables Supported in Dynamic Profiles

Type of Dynamic Profile	Junos OS Predefined Variable (Local)	Junos OS Predefined Variable (RADIUS)	User-Defined Variable
Access Profile	Yes	Yes	Yes
Service Profile	Yes	No	Yes

Table 7 on page 49 lists the default values, expressions, and unique identifiers supported by access profiles and service profiles.

Table 7: Default Values and Expressions Supported in Dynamic Profiles

Type of Dynamic Profile	Default Values	Expressions	Unique Identifiers
Access Profile	Yes (RADIUS predefined variables only)	No	Yes (Schedulers and Scheduler maps only)
Service Profile	Yes (User-defined variables only)	Yes (Service activation only)	Yes (Firewall filters only)

Dynamically Applying Services to Subscriber Sessions

You can configure services to be applied to subscriber sessions in several ways:

- Include service configurations for the subscriber session in a dynamic client profile. For example, you can configure Layer 2 services such as Class of Service (CoS) and Layer 3 services such as dynamic firewall filters. Layer 3 services are applied for the negotiated address family for DHCP, DHCPv6, and PPPoE subscribers. See *Changing CoS Services Overview*.

> ⓘ **NOTE**: A dynamic client profile cannot reference a dynamic service profile. It can only directly include service configurations.

- Apply a dynamic service profile using your RADIUS configuration. The Juniper Networks Activate-Service VSA (26-65), returned in the RADIUS Access-Accept message when the subscriber authenticates, can reference a dynamic service profile and optionally pass additional parameters for the service. For DHCP and PPPoE sessions, this service profile is applied when the session's address family is activated. See *Dynamic Service Management with RADIUS*.

You can use another Juniper Networks VSA, Deactivate-Service (26-66), to deactivate services in the Access Accept message.

- Apply a service profile with a Juniper Networks VSA in a RADIUS Change of Authorization (CoA) message. You can use a CoA message to activate (VSA 26-65) or deactivate (VSA 26-66) services. For example, a subscriber may opt in or out of a service after the session is established. See *RADIUS-Initiated Change of Authorization (CoA) Overview*.

- Apply a dynamic service profile by including the `service-profile` statement to reference the profile in the configurations for DHCP local server, DHCP relay agent, L2TP, or static subscribers. For example, see *Specifying the Static Subscriber Group Service Profile*, *Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces*, and *Configuring an L2TP Access Profile on the LNS*.

Dynamic Profile Overrides

Starting in Junos OS Release 14.1, you can specify a different dynamic profile in the RADIUS Client-Profile-Name VSA [26-174] to have RADIUS override a configured client dynamic profile. RADIUS returns this VSA to AAA with other client session attributes in the Access-Accept message. AAA subsequently overrides the corresponding profile name attribute in the session database entry for the client, and this new profile is instantiated instead of the originally configured profile.

Dynamic Profile Version Creation

You can create new versions of dynamic profiles that are currently in use by subscribers. Dynamic profile version creation is enabled at the `[edit system]` hierarchy level. When enabled, you can create multiple versions of any dynamic profiles on the router. Any subscriber that logs in following a dynamic profile modification uses the latest version of the dynamic profile. Subscribers that are already active continue to use the older version of the dynamic profile until they log out or their session terminates.

When creating versions of dynamic profiles, keep the following in mind:

- You must enable or disable dynamic profile version creation before creating or using any dynamic profiles on the router. Enabling or disabling dynamic profile version creation after dynamic profiles are configured is not supported.

> **NOTE**: Before you enable or disable dynamic profile version creation for a router on which any dynamic profiles are configured, you must first remove all dynamic profiles from the router configuration.

- Each version of a dynamic profile is stored in the profile database as a new profile.

- The name of the new profile version is derived by appending a string to the original base dynamic profile name. This string contains two dollar sign ($) characters to identify the version field of the

profile name. These two characters are followed by numerical characters that represent the "version number" of the dynamic profile (for example, 01).

- The version number of the dynamic profile is automatically generated by the system.

- The dynamic profile that you modify is always stored as the latest version. You cannot create a modified dynamic profile and save it as an earlier version. For example, if you modify version three of a dynamic profile while it is in use, the dynamic profile is saved as version four.

- You can only modify the latest version of a dynamic profile.

- The maximum value for the version number is 99999. However, for each profile, only 10 active versions are supported at a time.

- If the dynamic profile version that you modify is not in use by any subscriber, the profile is overwritten with committed changes without creating a new version.

- After reaching the 99999th modified version of a dynamic profile, any further modifications to the dynamic profile result in overwriting that final version. If the final version is in use, any modification attempts fail upon commit.

- You can delete a dynamic profile only when none of its versions are in use.

- The dynamic profile version feature supports graceful restart and unified ISSU.

Dynamic Profile Semantic Checking

Variables are applied to dynamic profiles dynamically and cannot be checked with existing CLI commands. Semantic checking validates some variables in dynamic profiles to help identify potential configuration errors.

Semantic checks are performed during commit and during profile instantiation. Commit time checks ensure that variables appear in the correct location within the dynamic profile. Checks performed before profile instantiation ensure that the values that replace the variables are correct. The checks performed on the values include the following:

- Range validation

- Variable type validation

- Existence of variables where they are mandatory

- Variable matching to regular expressions

A commit time check failure results in an error message being displayed and logged in the `/var/log/messages` file and the commit failing. An instantiation failure results in an error being logged in the `/var/log/messages` file and the profile instantiation failing.

Configuring a Basic Dynamic Profile

This topic describes how to create a basic dynamic profile. A basic profile must contain a profile name and have both an interface variable name (such as `$junos-interface-ifd-name`) included at the `[edit dynamic-profiles` *profile-name* `interfaces` hierarchy level and logical interface variable name (such as `$junos-underlying-interface-unit` or `$junos-interface-unit`) at the `[edit dynamic-profiles` *profile-name* `interfaces` *variable-interface-name* `unit]` hierarchy level.

Before you configure dynamic profiles for initial client access:

1. Configure the necessary router interfaces that you want DHCP clients to use when accessing the network.

    See *DHCP Subscriber Interface Overview* for information about the types of interfaces you can use with dynamic profiles and how to configure them.

2. Configure all RADIUS values that you want the profiles to use when validating DHCP clients for access to the multicast network.

    See *RADIUS Servers and Parameters for Subscriber Access*

To configure a basic dynamic profile:

1. Name the profile.

    ```
    [edit]
    user@host# edit dynamic-profiles basic-profile
    ```

2. Define the `interface-name` statement with the internal `$junos-interface-ifd-name` variable used by the router to match the interface name of the receiving interface.

    ```
    [edit dynamic-profiles basic-profile]
    user@host# edit interfaces $junos-interface-ifd-name
    ```

3. Define the `unit` statement with the internal variable:

    - When referencing an existing interface, specify the `$junos-underlying-interface-unit` variable used by the router to match the unit value of the receiving interface.

- When creating dynamic interfaces, specify the `$junos-interface-unit` variable used by the router to generate a unit value for the interface.

```
[edit dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name"]
user@host# set unit $junos-underlying-interface-unit
```

or

```
[edit dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name"]
user@host# set unit $junos-interface-unit
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can specify a different dynamic profile in the RADIUS Client-Profile-Name VSA [26-174] to have RADIUS override a configured client dynamic profile.

RELATED DOCUMENTATION

Unique Identifiers for Firewall Variables

Per-Subscriber Support of Maximum Transmission Unit for Dynamic Profiles

You can create per subscriber support of maximum transmission unit (MTU) for dynamic profiles. For more information, see the following topics:

Understanding Per-subscriber Support of Maximum Transmission Unit for Dynamic Profiles

Maximum transmission unit (MTU) is used to determine the maximum size of each packet in any TCP or IP transmission. MTU cannot be greater than the payload size that is the encapsulations at the assigned layer and any lower layers are excluded. You can specify the MTU for statically configured logical interfaces. Starting in Junos OS Release 18.2R1, you can configure an MTU value for a subscriber logical interface in a dynamic profile. This feature is required in customer applications requiring per-subscriber MTU for logical interfaces on the same underlying physical interface (from which the MTU is inherited by default). The use case is primarily dynamic VLANs for DHCP or DHCPv6 or IPoE or IPv6oE. The dynamic logical interface MTU must be no greater than the physical interface MTU minus the VLAN header size. The per-subscriber MTU feature is provided by extending dynamic-profiles to allow MTU to

be configured, either with a static value or the predefined variable, $junos-interface-mtu, whose value is provided by RADIUS.

MTU Per-Subscriber for Dynamic Profiles

A dynamic profile is a set of characteristics, defined in a type of template, that you can use to provide dynamic subscriber access and services for broadband applications. These services are assigned dynamically to interfaces. You can identify subscribers statically or dynamically. To identify subscribers statically, you can reference a static VLAN interface in a dynamic profile. To identify subscribers dynamically, you need to create variables for demux interfaces that are dynamically created when the subscribers log in. Junos OS allows you to create MTU for each subscriber for dynamic profiles. The value can be static or can be represented through a new variable, $junos-interface-mtu. By default, the variable value is the MTU of the payload, that is, the MTU of the physical interface minus the VLAN header size. A specific value can be returned through the RADIUS authentication in the Framed-MTU attribute (12). The attribute includes a single value which is applied to both the inet and inet6 protocol families if both are configured with $junos-interface-mtu variable. While applying the MTU on the subscriber logical interface during dynamic profile instantiation, a check is made to ensure that the MTU of the logical interface does not exceed what is supported on its physical interface along with the family protocol overhead. The value of the static MTU should be within the acceptable MTU range. If RADIUS does not return a Framed MTU value for $junos-interface-mtu variable, the default value for `interface-mtu` is used. You configure this value at the `[edit dynamic profiles` *dynamic-profiles* `predefined-variable-defaults]` hierarchy level. If neither is provided, then the profile request is NACKed.

Benefits of Per Subscriber Support of MTU for Dynamic Profiles

- Provides network scalability if each subscriber uses different dynamic profile or different subscriber name.

- Allows each subscriber to send traffic with different traffic rate.

Limitations

The following are the limitations:

- MTU for a dynamic logical interface is applied using the same rules as static logical interfaces.

- Framed-MTU returned by RADIUS is applicable only to the authenticated session. In other words, an authenticated dynamic VLAN (DVLAN) profile affects only the MTU for the vlan logical interface, and an authenticated DHCP profile affects only the MTU for the DHCP subscriber logical interface.

- If the RADIUS does not return a value in the Framed-MTU attribute (12), the profile request is NACKed.

- A commit check ensures that the mtu is specified for inet and inet6 address family, they must both be configured as explicit values if not the same value, or both must be configured with the $junos-interface-mtu predefined variable (in which case they are set to the same value). Otherwise, the configuration is forbidden.

Configuring Per-subscriber Maximum Transmission Unit for Dynamic Profiles

The maximum transmission unit (MTU) can be configured per subscriber for dynamic profiles. The value of MTU can be static or represented through the $junos-interface-mtu predefined variable. By default, the variable value is the MTU of the payload, which must be less than or equal to the MTU of the physical interface minus the VLAN header size. A specific value can be returned through RADIUS authentication through the Framed-MTU attribute (12). If the RADIUS server fails to return a value in the Framed MTU attribute, then the default value configured with `interface-mtu` statement at the `[edit dynamic-profiles profile-name predefined-variable-defaults]` hierarchy level is used. You can configure the MTU value with the `mtu` statement at the `[edit dynamic-profiles name interfaces name unit name family inet]` hierarchy level or at the `[edit dynamic-profiles name interfaces name unit name family inet6]` hierarchy level.

Before you begin, configure the device interfaces.

To configure per-subscriber MTU for dynamic profiles:

1. Configure per-subscriber MTU for dynamic profiles by hardcoding the mtu value for inet or inet 6 family.

```
[edit dynamic-profiles dynamic-profiles-name interfaces "$junos-interface-ifd-name" unit unit-name family inet]
user@host# set mtu mtu-value
```

```
[edit dynamic-profiles dynamic-profiles-name interfaces "$junos-interface-ifd-name" unit unit-name family inet6]
user@host# set mtu mtu-value
```

Configure pre-subscriber MTU for dynamic profiles by hardcoding an mtu value of 1450 for family inet of dynamic profile vlan-profile.

```
[edit dynamic-profiles vlan-profile interfaces "$junos-interface-ifd-name" unit 100 family
inet]
user@host# set mtu 1450
```

2. Configure the value for default for junos-interface-mtu.

```
[edit dynamic-profiles dynamic-profiles-name predefined-variable-defaults]
user@hots# interface-mtu value
```

Configure interface mtu value of 1450 for predefined variable defaults of dynamic profile vlan-profile.

```
[edit dynamic-profiles vlan-profile predefined-variable-defaults]
user@hots# interface-mtu 1450
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can configure an MTU value for a subscriber logical interface in a dynamic profile.

RELATED DOCUMENTATION

Dynamic Variables Overview

Variables constitute the dynamic component of a dynamic profile. You use variables in dynamic profiles as placeholders for dynamically obtained or dynamically generated values that the dynamic profiles use to configure subscriber interfaces and provision subscribers.

How Dynamic Variables Work

Dynamic variables are data placeholders that you define and place in dynamic profiles. When a particular event occurs on an interface (for example, a DHCP client accesses the interface), the dynamic profiles obtain data to fill these placeholders from one of three sources—the interface receiving an incoming client data packet, an externally configured server (for example, RADIUS), or a value associated with each user-configurable variable.

For your convenience, Junos OS provides predefined variables that you can use within a dynamic profile. Most of these variables relate to interface-specific data obtained directly from the interface that receives an incoming client data packets (for example, interface name, interface unit value, and so on). When a client accesses the interface, the router software extracts the necessary interface data, propagates this data to the dynamic profile, and then uses the dynamic profile to configure the interface for the accessing client.

You can define user-defined variables for individual dynamic profiles at the `[dynamic-profiles` *profile-name* `variables]` hierarchy level. At this hierarchy level, you create an association between a variable value (for example, `$junos-igmp-version`) that appears in the body of the dynamic profile and data associated with that call value that is managed in an externally configured server (for example, a RADIUS VSA managed on a RADIUS server) or defined as a value in the `variables` stanza. When an event occurs on an interface to trigger the instantiation of a dynamic profile for the interface, Junos OS obtains values for each variable from an external server (for example, from RADIUS authentication and authorization VSAs) during the subscriber authentication process. At run time, the variables are replaced by these actual values and are used to configure the subscriber interface.

Default Values for Predefined Variables

You can optionally configure default values for many predefined variables. The Junos OS uses the default value in the following cases:

- When the external RADIUS server is not available

- When the VSA returned by the RADIUS server does not contain a value for the predefined variable

Unique Identifier (UID) for Parameterized Filters

Parameterized filters use unique identifiers (UIDs) in dynamic profiles created for services. The generated UIDs enable you to identify and configure separate parameter values for filters with the same variable name. In addition, assigning a UID improves performance of the router.

For service profiles, you can request the generation of a UID for a user-defined variable by including the `uid` statement at the [`dynamic-profiles` *profile-name* `variables` *variable-name*] hierarchy level. You then reference the variable name in the filter.

To enable selection of a particular filter in a dynamic profile that contains multiple variables of the same parameter and criteria type, you must indicate that the variable refers to a UID. To configure, include the `uid-reference` statement at the [`dynamic-profiles` *profile-name* `variables` *variable-name*] hierarchy level. For example, if the variable `$in-filter` receives the value of "filter1" from RADIUS, the filter definition named $filter is used.

RELATED DOCUMENTATION

Parameterized Filters Overview

RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework

Predefined Variables in Dynamic Profiles

Predefined Variables

The dynamic profile obtains and replaces values for predefined variables from an incoming client data packet and configuration (local and RADIUS). These variables are predefined—you use them in the body of a dynamic profile without first having to define the variables at the `[dynamic-profiles *profile-name* variables]` hierarchy level. Table 8 on page 60 provides a list of predefined variables, their descriptions, and where in the Junos OS hierarchy you can configure them.

Table 8: Predefined Variables and Definitions

Variable	Definition
Access and Access-Internal Routes	

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-framed-route-cost	Cost metric of an IPv4 access route. You specify this variable with the `metric` statement at the [`edit dynamic-profiles` *profile-name* `routing-options access route` *address*] hierarchy level.
$junos-framed-route-distance	Distance of an IPv4 access route. You specify this variable with the `preference` statement at the [`edit dynamic-profiles` *profile-name* `routing-options access route` *address*] hierarchy level.
$junos-framed-route-ip-address-prefix	Route prefix of an IPv4 access route. You specify this variable at the [`edit dynamic-profiles` *profile-name* `routing-options access`] hierarchy level for the `route` statement.
$junos-framed-route-ipv6-address-prefix	Route prefix of an IPv6 access route. You specify this variable with the `route` statement at either of the following hierarchy levels: • [`edit dynamic-profiles` *profile-name* `routing-instances $junos-routing-instance routing-options rib $junos-ipv6-rib access`] • [`edit dynamic-profiles` *profile-name* `routing-options rib $junos-ipv6-rib access`]

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-framed-route-ipv6-cost	Cost metric of an IPv6 access route. You specify this variable with the `metric` statement at either of the following hierarchy levels: • `[edit dynamic-profiles profile-name routing-instances $junos-routing-instance routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]` • `[edit dynamic-profiles profile-name routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]`
$junos-framed-route-ipv6-distance	Distance of an IPv6 access route. You specify this variable with the `preference` statement at either of the following hierarchy levels: • `[edit dynamic-profiles profile-name routing-instances $junos-routing-instance routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]` • `[edit dynamic-profiles profile-name routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]`
$junos-framed-route-ipv6-nexthop	IPv6 next-hop address of an access route. You specify this variable with the `next-hop` statement at either of the following hierarchy levels: • `[edit dynamic-profiles profile-name routing-instances $junos-routing-instance routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]` • `[edit dynamic-profiles profile-name routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]`

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-framed-route-ipv6-tag	Tag value of an IPv6 access route. You specify this variable with the `tag` statement at either of the following hierarchy levels: • `[edit dynamic-profiles profile-name routing-instances $junos-routing-instance routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]` • `[edit dynamic-profiles profile-name routing-options rib $junos-ipv6-rib access route $junos-framed-route-ipv6-address-prefix]`
$junos-framed-route-nexthop	IPv4 next-hop address of an access route. You specify this variable at the `[edit dynamic-profiles profile-name routing-options access route address]` hierarchy level for the `next-hop` statement.
$junos-framed-route-tag	Tag value of an IPv4 access route. You specify this variable at the `[edit dynamic-profiles profile-name routing-options access route address]` hierarchy level for the `tag` statement.
$junos-framed-route-tag2	Tag2 value for static routes. You specify this variable at the `[edit dynamic-profiles profile-name routing-options access route $junos-framed-route-ip-address-prefix]` hierarchy level for the `tag2` statement.
$junos-interface-name	Logical interface of an access-internal route. DHCP or PPP supplies this information when the subscriber logs in. You specify this variable at the `[edit dynamic-profiles profile-name routing-options access-internal route address]` hierarchy level for the `qualified-next-hop` statement. This variable is also used for creating dynamic IP demux interfaces.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-ipv6-rib	Routing table for an IPv6 access route. You specify this variable with the `rib` statement at the `[edit dynamic-profiles` *profile-name* `routing-instances $junos-routing-instance routing-options]` hierarchy level. You can use this variable to specify a nondefault routing instance for the route.
$junos-subscriber-ip-address	IP address of a subscriber identified in an access-internal route. You specify this variable at the `[edit dynamic-profiles` *profile-name* `routing-options access-internal]` hierarchy level for the `route` statement. This variable is also used for creating dynamic IP demux interfaces.
$junos-subscriber-mac-address	MAC address for a subscriber identified in an access-internal route. You specify this variable at the `[edit dynamic-profiles` *profile-name* `routing-options access-internal route` *address* `qualified-next hop` *underlying-interface*`]` hierarchy level for the `mac-address` statement.
Dynamic Protocols	
$junos-igmp-access-group-name	Specifies the access list to use for the source (S) filter.
$junos-igmp-access-source-group-name	Specifies the access list to use for the source-group (S,G) filter.
$junos-igmp-enable	Ensures that IGMP is not disabled on the interface by an AAA-based authentication and management method (for example, RADIUS). You specify this variable at the `[dynamic-profiles` *profile-name* `protocols igmp]` hierarchy level for the `interface` statement.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-igmp-immediate-leave	Enables IGMP immediate leave on the interface. You specify this variable at the [dynamic-profiles *profile-name* protocols igmp] hierarchy level for the interface statement.
$junos-igmp-version	IGMP version configured in a client access profile. Junos OS obtains this information from the RADIUS server when a subscriber accesses the router. The version is applied to the accessing subscriber when the profile is instantiated. You specify this variable at the [dynamic-profiles *profile-name* protocols igmp] hierarchy level for the interface statement.
$junos-interface-name	Name of the dynamic interface to which the subscriber access client connects. Its use is in dynamically enabling IGMP on the subscriber interface. You specify this variable at the [dynamic-profiles *profile-name* protocols igmp] hierarchy level for the interface statement. The interface name is derived from concatenating the $junos-interface-ifd-name and the $junos-underlying-interface-unit variables obtained when a subscriber is created dynamically at the [dynamic-profiles *profile-name* interfaces] hierarchy level.
$junos-ipv6-ndra-prefix	Prefix value for the router advertisement interface. Junos OS obtains this information from the RADIUS server when a subscriber accesses the router. The prefix value is applied to the accessing subscriber when the profile is instantiated. You specify this variable at the [dynamic-profiles *profile-name* protocols router-advertisement interface *$junos-interface-name*] hierarchy level.
$junos-mld-access-group-name	Specifies the access list to use for the group (G) filter.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-mld-access-source-group-name	Specifies the access list to use for the source-group (S,G) filter.
$junos-mld-enable	Ensures that MLD is not disabled on the interface by an AAA-based authentication and management method (for example, RADIUS). You specify this variable at the `[dynamic-profiles profile-name protocols mld]` hierarchy level for the `interface` statement.
$junos-mld-immediate-leave	Enables MLD immediate leave on the interface. You specify this variable at the `[dynamic-profiles profile-name protocols mld]` hierarchy level for the `interface` statement.
$junos-mld-version	MLD version configured in a client access profile. Junos OS obtains this information from the RADIUS server when a subscriber accesses the router. The version is applied to the accessing subscriber when the profile is instantiated. You specify this variable at the `[dynamic-profiles profile-name protocols mld]` hierarchy level for the `interface` statement.

Dynamic CoS — Traffic-Control Profile Parameters

Variable	Definition
$junos-cos-adjust-minimum	Minimum adjusted shaping rate configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `adjust-minimum` statement at the `[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-byte-adjust	Byte adjustment value configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `bytes` option with the `overhead-accounting` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service traffic-control-profiles` *profile-name*`]` hierarchy level.
$junos-cos-byte-adjust-cell	Overhead bytes when downstream ATM traffic is in cell-mode. **NOTE**: Do not configure the $junos-cos-byte-adjust-cell variable when the $junos-cos-byte-adjust variable is configured.
$junos-cos-byte-adjust-frame	Overhead bytes when downstream ATM traffic is in frame-mode. **NOTE**: Do not configure the $junos-cos-byte-adjust-frame variable when the $junos-cos-byte-adjust variable is configured.
$junos-cos-delay-buffer-rate	Delay-buffer rate configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `delay-buffer-rate` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service traffic-control-profiles` *profile-name*`]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-excess-rate	Excess rate configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `excess-rate` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.
$junos-cos-excess-rate-high	Rate configured for excess high-priority traffic in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `excess-rate-high` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.
$junos-cos-excess-rate-low	Rate configured for excesslow-priority traffic in a traffic-control profile in a dynamic profile for subscriber access. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `excess-rate-low` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-guaranteed-rate	Guaranteed rate configured in a traffic-control profile in a dynamic profile Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `guaranteed-rate` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service traffic-control-profiles` *profile-name*`]` hierarchy level.
$junos-cos-guaranteed-rate-burst	Burst size for the guaranteed rate that is configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable with the `burst-size` option in the `guaranteed-rate` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service traffic-control-profiles` *profile-name*`]` hierarchy level.
$junos-cos-scheduler-map	Scheduler-map name configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `scheduler-map` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service traffic-control-profiles` *profile-name*`]` hierarchy level. The scheduler map can be defined dynamically (at the `[edit dynamic-profiles` *profile-name* `class-of-service scheduler-maps]` hierarchy level) or statically (at the `[edit class-of-service scheduler-maps]` hierarchy level).

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-shaping-mode	Shaping mode configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `overhead-accounting` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.
$junos-cos-shaping-rate	Shaping rate configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `shaping-rate` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.
$junos-cos-shaping-rate-burst	Burst size for the shaping rate configured in a traffic-control profile in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable with the `burst-size` option in the `shaping-rate` statement at the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles *profile-name*] hierarchy level.
$junos-cos-shaping-rate-excess-high	Shaping rate configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-shaping-rate-excess-high-burst	Shaping rate burst size configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-excess-low	Shaping rate configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-excess-low-burst	Shaping rate burst size configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-high	Shaping rate configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-high-burst	Shaping rate burst size configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-low	Shaping rate configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-shaping-rate-priority-low-burst	Shaping rate burst size configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-medium	Shaping rate configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-medium-burst	Shaping rate burst size configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-traffic-control-profile	Traffic-control profile configured in a dynamic profile for subscriber access. The Junos OS obtains the profile information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `traffic-control-profiles` statement at the [edit dynamic-profiles *profile-name* class-of-service] hierarchy level.

Dynamic CoS — Scheduler Parameters

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-scheduler	Name of a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable at the `[edit dynamic-profiles` *profile-name* `class-of-service schedulers]` hierarchy level.
$junos-cos-scheduler-bs	Buffer size as a percentage of total buffer, specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `buffer-size` statement with the `percent` option at the `[edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*`]` hierarchy level.
$junos-cos-scheduler-pri	Packet-scheduling priority value specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `priority` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*`]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-scheduler-dropfile-any	Name of the drop profile for random early detection (RED) for loss-priority level `any` specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `drop-profile` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority any protocol any]` hierarchy level. The drop profile must be configured statically (at the `[edit class-of-service drop-profiles]` hierarchy level).
$junos-cos-scheduler-dropfile-high	Name of the drop profile for random early detection (RED) for loss-priority level `high` specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `drop-profile` statement at the `[edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority high protocol any]` hierarchy level. The drop profile must be configured statically (at the `[edit class-of-service drop-profiles]` hierarchy level).

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-scheduler-dropfile-low	Name of the drop profile for random early detection (RED) for loss-priority level `low` specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `drop-profile` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority low protocol any`] hierarchy level. NOTE: The drop profile must be configured statically (at the [`edit class-of-service drop-profiles`] hierarchy level) for loss-priority `low`.
$junos-cos-scheduler-dropfile-medium-high	Name of the drop profile for random early detection (RED) for loss-priority level `medium-high` specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `drop-profile` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority medium-high protocol any`] hierarchy level. The drop profile must be configured statically (at the [`edit class-of-service drop-profiles`] hierarchy level).

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-scheduler-dropfile-medium-low	Name of the drop profile for random early detection (RED) for loss-priority level `medium-low` specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `drop-profile` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name* `drop-profile-map loss-priority medium-low protocol any`] hierarchy level. NOTE: The drop profile must be configured statically (at the [`edit class-of-service drop-profiles`] hierarchy level).
$junos-cos-scheduler-excess-priority	Priority value of the excess rate specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `excess-priority` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*] hierarchy level.
$junos-cos-scheduler-excess-rate	Value of the excess rate specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `excess-rate` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-cos-scheduler-shaping-rate	Value of the shaping rate specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `shaping-rate` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*] hierarchy level.
$junos-cos-scheduler-tx	Transmit rate specified for a scheduler configured in a dynamic profile. Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the dynamic profile is attached. You reference this variable in the `transmit-rate` statement at the [`edit dynamic-profiles` *profile-name* `class-of-service schedulers` *scheduler-name*] hierarchy level.

Dynamic CoS — Dynamic Interface Sets

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-aggregation-interface-set-name	Name of a hierarchical CoS L2 interface set that represents a logical intermediate node (DPU-C or PON tree) in the access network. This is also known as the dynamic aggregation interface set. The variable take one of the following values: • When the `hierarchical-access-network-detection` option is configured for the access lines and the value of the Access-Aggregation-Circuit-ID-ASCII attribute (TLV 0x03) received either in the ANCP Port Up message or PPPoE PADR IA tags begins with a # character, then the variable takes the value of the remainder of the string after the # character. The # character identifies the string as a backhaul identifier. • When the `hierarchical-access-network-detection` option is not configured, or if the string does not begin with the # character, then the variable takes the value specified with the `predefined-variable-defaults` statement. You reference this variable in the `interface-set` statement at the [`edit dynamic-profiles` *profile-name* `interfaces`] hierarchy level.
$junos-interface-set-name	Name of the interface set obtained from the RADIUS server Access-Accept message when a subscriber authenticates over the interface to which the dynamic profile is attached. You reference this variable in the `interface-set` statement at the [`edit dynamic-profiles` *profile-name* `interfaces`] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-phy-ifd-interface-set-name	Locally generated name of an interface set that is associated with the underlying physical interface in a dynamic profile. This predefined variable enables you to group all the subscribers on a specific physical interface so that you can apply services to the entire group of subscribers. This interface set is a default level 2 interface set for four-level hierarchies and a default level 3 interface set for five-level hierarchies. You can use this predefined variable to conserve CoS resources in a mixed business and residential topology by collecting the residential subscribers into an interface set associated with the physical interface. This causes a level 2 node to be used for the interface set rather than one for each residential interface. Otherwise, because the business and residential subscribers share the same interface and business subscribers require three levels of CoS, then three levels are configured for each residential subscriber. That results in an unnecessary level 2 node being consumed for each residential connection, wasting CoS resources. You reference this variable in the `interface-set` statement at the `[edit dynamic-profiles `*`profile-name`*` interfaces]` hierarchy level.
$junos-phy-ifd-underlying-intf-set-name	Name of a default, topology-based interface set that is based on the physical interface name with a suffix of "-underlying" to conserve hierarchical CoS L2 nodes. This interface is used as a default level 2 interface set. You reference this variable in the `interface-set` statement at the `[edit dynamic-profiles `*`profile-name`*` interfaces]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-svlan-interface-set-name	Locally generated name of an interface set for use by dual-tagged VLAN (S-VLAN) interfaces. The name is based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical_interface_name-outer_vlan_tag*. For example, an Ethernet interface of ge-1/1/0, with a dual-tagged VLAN interface that has an outer tag of 111, results in a value of ge-1/1/0-111 for $junos-svlan-interface-set-name. You reference this variable in the `interface-set` statement at the `[edit dynamic-profiles` *profile-name* `interfaces]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-tagged-vlan-interface-set-name	Locally generated name of an interface set that groups logical interfaces stacked over logical stacked VLAN demux interfaces. You can use this variable for either a 1:1 (dual-tagged; individual client) VLAN or N:1 (single tagged; service) VLAN. The format of the generated variable differs with VLAN type as follows: • Dual-tagged (client) VLAN—The format is *physical_interface_name-outer_vlan_tag-inner_vlan_tag*. For example, an Ethernet interface of ge-1/1/0, with a dual-tagged VLAN interface that has an outer tag of 111 and an inner tag of 200, results in a value of ge-1/1/0-111–200 for $junos-tagged-vlan-interface-set-name. • Single tagged (service) VLAN—The format is *physical_interface_name-vlan_tag* For example, an Ethernet interface of ge-1/1/0, with an N:1 VLAN that has a single tag of 200, results in a value of ge-1/1/0-200 for $junos-tagged-vlan-interface-set-name. You reference this variable in the `interface-set` statement at the [edit dynamic-profiles *profile-name* interfaces] hierarchy level.

Dynamic Connectivity Fault Management Parameters

$junos-action-profile	Name of the action profile configured in a dynamic profile.
$junos-ccm-interval	Continuity check interval time configured in a dynamic profile.
$junos-loss-threshold	The number of continuity check messages lost before marking the remote MEP as down, configured in a dynamic profile.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-ma-name-format	Name of the maintenance association name format configured in a dynamic profile.
$junos-md-name-format	Name of the maintenance domain format configured in a dynamic profile.
$junos-ma-name	Name of the maintenance association configured in a dynamic profile.
$junos-md-level	Value of 'Level', configured in a dynamic profile.
$junos-md-name	Name of the maintenance domain configured in a dynamic profile.
$junos-mep-id	The 'MEP' value configured in the dynamic profile.
$junos-remote-mep-id	The 'Remote MEP' value configured in the dynamic profile.

Enhanced Hierarchical Policers (Junos OS Evolved)

$junos-hpolicer-high-cir	The committed information rate (CIR) for high-priority traffic in a hierarchical policer.
$junos-hpolicer-high-max-cir	The maximum committed information rate for high-priority traffic, which allows rate limiting up to this value.
$junos-hpolicer-high-committed-burst	The committed burst size for high-priority traffic, which limits the amount of burst traffic allowed.
$junos-hpolicer-med-high-cir	The CIR for medium-high priority traffic.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-hpolicer-med-high-max-cir	The maximum CIR for medium-high priority traffic.
$junos-hpolicer-med-high-committed-burst	The burst size for medium-high priority traffic.
$junos-hpolicer-med-low-cir	The CIR for medium-low priority traffic.
$junos-hpolicer-med-low-max-cir	The maximum CIR for medium-low priority traffic.
$junos-hpolicer-med-low-committed-burst	The burst size for medium-low priority traffic.
$junos-hpolicer-low-cir	The CIR for low-priority traffic.
$junos-hpolicer-low-max-cir	The maximum CIR for low-priority traffic.
$junos-hpolicer-low-committed-burst	The burst size for low-priority traffic.

Filters — RADIUS-obtained Policies

Variable	Definition
$junos-input-filter	Name of an input filter to be attached; filter name is derived from RADIUS VSA 26-10 (Ingress-Policy-Name) or RADIUS attribute 11 (Filter-ID) to the interface.
$junos-input-interface-filter	Name of an input filter to be attached to a `family any` interface; filter name is derived from RADIUS VSA 26-191 (Input-Interface-Filter) to the interface. You can also specify the filter name with the $`junos-input-interface-filter` statement at the [`edit dynamic-profiles` *profile-name* `interfaces` *interface-name* `unit` *logical-interface-number* `filter input`] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-input-ipv6-filter	Name of an IPv6 input filter to be attached; filter name is derived from RADIUS VSA 26-106 (IPv6-Ingress-Policy-Name) to the interface.
$junos-output-filter	Name of an output filter to be attached; filter name is derived from RADIUS VSA 26-11 (Egress-Policy-Name) to the interface.
$junos-output-interface-filter	Name of an output filter to be attached to a `family any` interface; filter name is derived from RADIUS VSA 26-191 (Output-Interface-Filter) to the interface. You can also specify the filter name with the `$junos-output-interface-filter` statement at the `[edit dynamic-profiles profile-name interfaces interface-name unit logical-interface-number filter output]` hierarchy level.
$junos-output-ipv6-filter	Name of an IPv6 output filter to be attached; filter name is derived from RADIUS VSA 26-107 (IPv6-Egress-Policy-Name) to the interface.
Services	
$junos-input-ipv6-service-filter	Starting in Junos OS Release 17.2R1, name of an IPv6 input service filter to be attached. The filter name is derived from RADIUS-VSA 26-202 (IPv6 input service filter) to the interface. You specify this variable at the `[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number family inet6 service input service-set service-set-name service-filter]` hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-input-ipv6-service-set	Starting in Junos OS Release 17.2R1, name of an IPv6 service set to be attached. The service set name is derived from RADIUS-VSA 26-200 (IPv6 input service set) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet6 service input service-set] hierarchy level.
$junos-input-service-filter	Starting in Junos OS Release 17.2R1, name of an IPv4 input service filter to be attached. The filter name is derived from RADIUS-VSA 26-198 (IPv4 input service filter) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet service input service-set *service-set-name* service-filter] hierarchy level.
$junos-input-service-set	Starting in Junos OS Release 17.2R1, name of an IPv4 input service set to be attached. The service set name is derived from RADIUS-VSA 26-196 (IPv4 input service set) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet service input service-set] hierarchy level.
$junos-output-ipv6-service-filter	Starting in Junos OS Release 17.2R1, name of an IPv6 service filter to be attached. The filter name is derived from RADIUS-VSA 26-203 (IPv6 output service filter) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet6 service output service-set *service-set-name* service-filter] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-output-ipv6-service-set	Starting in Junos OS Release 17.2R1, name of an IPv6 service set to be attached. The service set name is derived from RADIUS-VSA 26-201 (IPv6 output service set) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet6 service output service-set] hierarchy level.
$junos-output-service-filter	Starting in Junos OS Release 17.2R1, name of an IPv4 service filter to be attached. The filter name is derived from RADIUS-VSA 26-199 (IPv4 output service filter) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet service output service-set *service-set-name* service-filter] hierarchy level.
$junos-output-service-set	Starting in Junos OS Release 17.2R1, name of an IPv4 output service set to be attached. The service set name is derived from RADIUS-VSA 26-197 (IPv4 output service set) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet service output service-set] hierarchy level.
$junos-pcef-profile	Starting in Junos OS Release 17.2R1, name of a PCEF profile to be attached. The profile name is derived from RADIUS-VSA 26-204 (PCEF profile) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* service] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-pcef-rule	Starting in Junos OS Release 17.2R1, name of a PCC rule to activate. The rule name is derived from RADIUS-VSA 26-205 (PCEF rule) to the interface. You specify this variable at the [edit dynamic-profile *profile-name* interfaces *interface-name* unit *logical-unit-number* service pcef *pcef-profile-name* activate] hierarchy level.
Subscriber Interfaces — Dynamic Demux Interfaces	
$junos-interface-ifd-name	Name of the device to which the subscriber access client connects. All interfaces are created on this device. Its primary use is in creating single or multiple subscribers on a statically created interface. You specify this variable at the [dynamic-profiles *profile-name* interfaces] hierarchy level. When creating a logical underlying interface for a dynamic VLAN demux interface, you must also specify this variable at the [dynamic-profiles *profile-name* interfaces *demux0* unit *$junos-interface-unit* demux-options underlying-interface] hierarchy level.
$junos-interface-target-weight	Weight for an interface to associate it with an interface set and thus with the set's aggregated Ethernet member link for targeted distribution. When an interface set does not have a weight, then the interface weight value for the first authorized subscriber interface is used for the set. The value is derived from RADIUS VSA 26-214 Interface-Target-Weight. Also associated with Diameter AVP 214.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-interface-unit	Creates a unit number assigned to the logical interface. The router supplies this information when the subscriber accesses the network. You specify this variable at the [dynamic-profiles *profile-name* interfaces *interface-name*] hierarchy level for the unit statement.
$junos-ipv6-address	Selects the IPv6 address of the interface the subscriber uses. You specify this variable at the [edit dynamic-profiles *profile-name* interfaces *interface-name* unit *logical-unit-number* family *family*], [edit dynamic-profiles *profile-name* interfaces demux0 unit *logical-unit-number* family *family*], [edit dynamic-profiles *profile-name* interfaces pp0 unit "$junos-interface-unit" family *family*], and [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level for the address statement.
$junos-loopback-interface	Selects the loopback interface the subscriber uses. You specify this variable at the [dynamic profiles *profile-name* interfaces demux0 unit "$junos-interface-unit" family inet] hierarchy level for the unnumbered-address statement.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-preferred-source-address	Selects the preferred IPv4 source address (family inet) associated with the loopback address used for the subscriber. You specify this variable at the `[dynamic profiles `*`profile-name`*` interfaces demux0 unit "$junos-interface-unit" family inet unnumbered-address "$junos-loopback-interface"]` hierarchy level for the `preferred-source-address` statement. When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the `$junos-routing-instance` predefined variable, you must not configure an IPv4 preferred source address. This constraint applies whether you use the `$junos-preferred-source-address` predefined variable or the `preferred-source-address` statement. Configuring the preferred source address in this circumstance causes a commit failure.
$junos-preferred-source-ipv6-address	Selects the preferred IPv6 source address (family inet6) associated with the loopback address used for the subscriber. You specify this variable at the `[dynamic profiles `*`profile-name`*` interfaces demux0 unit "$junos-interface-unit" family inet6 unnumbered-address "$junos-loopback-interface"]` hierarchy level for the `preferred-source-address` statement. When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the `$junos-routing-instance` predefined variable, you must not configure an IPv6 preferred source address. This constraint applies whether you use the `$junos-preferred-source-ipv6-address` predefined variable or the `preferred-source-address` statement. Configuring the preferred source address in this circumstance causes a commit failure.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-subscriber-demux-ip-address	IP address of the subscriber. Use this variable instead of $junos-subscriber-ip-address when the IP demux subscribers require a framed route returned from the RADIUS server. You specify this variable at the [dynamic-profiles *profile-name* interfaces *demux0* unit family *inet* demux-source] hierarchy level.
$junos-subscriber-ip-address	IP address of the subscriber. You specify this variable at the [dynamic-profiles *profile-name* interfaces *demux0* unit family *inet* demux-source] hierarchy level. This variable is also used for creating access-internal routes.
$junos-subscriber-ipv6-address	IPv6 address for subscriber. You specify this variable at the [dynamic-profiles *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet6 demux-source] hierarchy level.
$junos-subscriber-ipv6-multi-address	Expands the demux-source into multiple addresses; for example, the IPv6 prefix and /128 address for the subscriber. You specify this variable at the [dynamic-profiles *profile-name* interfaces *interface-name* unit *logical-unit-number* family inet6 demux-source] hierarchy level.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-underlying-interface	Creates a logical underlying interface for a dynamic IP demux interface. The client logs in on this interface. You specify this variable at the [dynamic profiles *profile-name* interfaces demux0 unit "$junos-interface-unit" demux-options] hierarchy level for the underlying-interface statement.
	When configured, the underlying interface is used to determine the $junos-underlying-interface, $junos-underlying-interface-unit, and $junos-ifd-name variables. For example, if the receiving logical interface is ge-0/0/0.1, the $junos-underlying-interface variable is set to ge-0/0/0 and the $junos-underlying-interface-unit variable is set to 1.
	This variable is also used for creating access-internal routes.

Subscriber Interfaces — Static VLAN Interfaces

Variable	Definition
$junos-interface-ifd-name	Name of the device to which the subscriber access client connects. All interfaces are created on this device. Its primary use is in creating single or multiple subscribers on a statically created interface. You specify this variable at the [dynamic-profiles *profile-name* interfaces] hierarchy level.
$junos-underlying-interface-unit	Obtains the unit number for the underlying interface. It specifies the use of the underlying interface for the subscriber. You specify this variable at the [dynamic-profiles *profile-name* interfaces *$junos-interface-ifd-name*] hierarchy level for the unit statement.

Subscriber Interfaces — Dynamic PPPoE Interfaces

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-interface-unit	Specifies the logical unit number when the router dynamically creates a PPPoE logical interface. The `$junos-interface-unit` predefined variable is dynamically replaced with the unit number supplied by the network when the PPPoE subscriber logs in. You specify this variable at the [`edit dynamic-profiles` *profile-name* `interfaces pp0`] hierarchy level for the `unit` statement.
$junos-underlying-interface	Specifies the name of the underlying Ethernet interface on which the router dynamically creates the PPPoE logical interface. The `$junos-underlying-interface` predefined variable is dynamically replaced with the name of the underlying interface supplied by the network when the PPPoE subscriber logs in. You specify this variable at the [`edit dynamic-profiles` *profile-name* `interfaces pp0 unit "$junos-interface-unit" pppoe-options`] hierarchy level for the `underlying-interface` statement.
Subscriber Interfaces — Dynamic Interface Sets	
$junos-interface-set-name	Name of an interface set configured in a dynamic profile. To represent the name of a dynamically created agent circuit identifier (ACI) interface set, use the `$junos-interface-set-name` predefined variable in the `interface-set` statement at the [`edit dynamic-profiles` *profile-name* `interfaces`] hierarchy level.
$junos-interface-set-target-weight	Weight for an interface set to associate it and its member links with an aggregated Ethernet member link for targeted distribution. The value is derived from RADIUS VSA 26-213 Interface-Set-Target-Weight. Also associated with Diameter AVP 213.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-phy-ifd-interface-set-name	Name of an interface set associated with the underlying physical interface in a dynamic profile.
	In a heterogeneous topology where residential and business subscribers share the same physical interface, although only two levels of CoS are required for residential access, business access requires three levels. Because they share the same physical interface, three levels are configured for both, causing an unnecessary level 2 node to be consumed for each residential connection.
	You can reduce the CoS resources wasted on residential access by collecting the residential subscribers into an interface set associated with the physical interface. In this way, a level 2 node is used for the interface set rather than for each residential interface. To do so, specify the $junos-phy-ifd-interface-set-name predefined variable with the interface-set statement at the [edit dynamic-profiles *profile-name* interfaces] hierarchy level to create the interface set based on the underlying physical interface.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-pon-id-interface-set-name	Locally generated interface set name used to associate individual customer circuits in a passive optical network (PON) to deliver CoS and other services to the set of interfaces.
	The name is extracted from the DHCPv4 (Option 82, suboption 2) or DHCPv6 (Option 37) agent remote ID string inserted by an optical line terminal (OLT) in a PON. The OLT must format the agent remote ID string with a pipe symbol (|) as the delimiter between substrings. The substring extracted for the interface set name consists of the characters following the last delimiter in the agent remote ID string.
	The extracted substring identifies individual customer circuits. You determine the format and contents of the substring, and configure your OLT to insert the information. Typically, the substring might include the name and port of the OLT accessed by the CPE optical network terminal (ONT).
$junos-svlan-interface-set-name	Locally generated interface set name for use by dual-tagged VLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical_interface_name - outer_VLAN_tag*.

Wholesale Networking

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-interface-name	Name of the dynamic interface to which the subscriber access client connects. Its use is in identifying the subscriber interface. You specify this variable at the [dynamic-profiles *profile-name* routing-instance *$junos-routing-instance*] hierarchy level for the interface statement. The interface name is derived from concatenating the $junos-interface-ifd-name and the $junos-underlying-interface-unit variables obtained when a subscriber is created dynamically at the [dynamic-profiles *profile-name* routing-instance *$junos-routing-instance* interface] hierarchy level.
$junos-routing-instance	Name of the routing instance to which the subscriber is assigned. This variable triggers a return value from the RADIUS server for Virtual-Router (VSA 26-1). You reference this variable in the statement at the [dynamic-profiles *profile-name*] hierarchy level for the routing-instance statement. When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure a preferred source address. This constraint applies whether you use the $junos-preferred-source-address predefined variable, the $junos-preferred-source-ipv6-address predefined variable, or the preferred-source-address statement. Configuring the preferred source address in this circumstance causes a commit failure.

Table 8: Predefined Variables and Definitions *(Continued)*

Variable	Definition
$junos-inner-vlan-map-id	Define identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs.The VLAN tag is allocated from the inner VLAN ID swap ranges that are provisioned on the core-facing physical interface. The inner VLAN tag is swapped with (replaces) the outer VLAN tag when the subscriber traffic is tunneled to the NSP. You specify this variable with the `inner-vlan-id` statement at the [`edit dynamic-profiles` *profile-name* `interfaces $junos-interface-ifd-name unit $junos-interface-unit input-vlan-map`] hierarchy level.
$junos-vlan-map-id	Identifier for a VLAN that is rewritten at the input or output interface as specified by a VLAN map. You specify this variable with the `vlan-id` statement at the [`edit dynamic-profiles` *profile-name* `interfaces $junos-interface-ifd-name unit $junos-interface-unit input-vlan-map`] or [`edit dynamic-profiles` *profile-name* `interfaces $junos-interface-ifd-name unit $junos-interface-unit input-vlan-map`] hierarchy levels.

Change History for Predefined Variables

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 9: Change History for Predefined Variables

Release	Description
16.1	Starting in Junos OS Release 16.1: • When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure an IPv4 preferred source address. • You can reduce the CoS resources wasted on residential access by collecting the residential subscribers into an interface set associated with the physical interface. • When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure a preferred source address. • Starting in Junos OS Release 16.1R4, identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs. • When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure an IPv6 preferred source address. • You can reduce the CoS resources wasted on residential access by collecting the residential subscribers into an interface set associated with the physical interface. • When you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure a preferred source address. • identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs. Starting in Junos OS Release 16.1R4: • Define identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs. • Define identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs.

Table 9: Change History for Predefined Variables *(Continued)*

Release	Description
17.2	Starting in Junos OS Release 17.2R1: • predefined variables added for name of an IPv6 input service filter to be attached. • predefined variables added for name of an IPv6 service set to be attached. • predefined variables added for name of an IPv4 input service filter to be attached. • predefined variables added for name of an IPv6 service set to be attached. • predefined variables added for name of an IPv6 service filter to be attached. • predefined variables added for name of an IPv6 service set to be attached. • predefined variables added for name of an IPv4 service filter to be attached. • predefined variables added for name of an IPv4 output service set to be attached. • predefined variables added for name of a PCEF profile to be attached. • predefined variables added for name of a PCC rule to activate.
25.4R1Evo	Starting in Junos OS Evolved Release 25.4R1, specify the predefined variables when you configure enhanced hierarchical policer.

Predefined Variables That Correspond to RADIUS Attributes and VSAs

Table 10 on page 99 lists the RADIUS attributes and Juniper Networks VSAs and their corresponding predefined variables that are used in dynamic profiles. When the router instantiates a dynamic profile following subscriber access, the system uses the predefined variable to specify the RADIUS attribute or VSA for the information obtained from the RADIUS server.

Some predefined variables support the configuration of default values. The configured default value is used in the event that RADIUS fails to return a value for the variable. You configure default values with the `predefined-variable-defaults` *predefined-variable default-value* statement at the `[edit dynamic-profiles]` hierarchy level. When you specify the *predefined-variable*, you use the name of the predefined variable, but you omit the leading `$junos-` prefix.

Table 10: RADIUS Attributes and Corresponding Predefined Variables

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
RADIUS Standard Attributes			
8 Framed-IP-Address	$junos-framed-route-ip-address	Address for the client	No
11 Filter-ID	$junos-input-filter **NOTE**: Variable is also used for VSA 26-10.	Input filter to apply to client IPv4 interface	Yes
12 Framed-MTU	$junos-interface-mtu	Maximum size of the packet; maximum transmission unit	Yes
22 Framed-Route	$junos-framed-route-ip-address-prefix	(Subattribute 1): Route prefix for access route	No
	$junos-framed-route-nexthop	(Subattribute 2): Next hop address for access route	No
	$junos-framed-route-cost	(Subattribute 3): Metric for access route	No

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-framed-route-distance	(Subattribute 5): Preference for access route	No
	$junos-framed-route-tag	(Subattribute 6): Tag for access route	No
97 Framed-IPv6-Prefix	$junos-ipv6-ndra-prefix	Prefix value in IPv6 Neighbor Discovery route advertisements	No
99 Framed-IPv6-Route	$junos-framed-route-ipv6-address-prefix	(Subattribute 1): Framed IPv6 route prefix configured for the client	No
	$junos-framed-route-ipv6-cost	(Subattribute 3): Metric for access route	No
	$junos-framed-route-ipv6-distance	(Subattribute 5): Preference for access route	No
	$junos-framed-route-ipv6-nexthop	(Subattribute 2): IPv6 routing information configured for the client	No
	$junos-framed-route-ipv6-tag	(Subattribute 6): Tag for access route	No

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
Juniper Networks VSA			
26-1 Virtual-Router	$junos-routing-instance	Routing instance to which subscriber is assigned	Yes
26-10 Ingress-Policy-Name	$junos-input-filter **NOTE**: Variable is also used for RADIUS attribute 11.	Input filter to apply to client IPv4 interface	Yes
26-11 Egress-Policy-Name	$junos-output-filter	Output filter to apply to client IPv4 interface	Yes
26-23 IGMP-Enable	$junos-igmp-enable	Enable or disable IGMP on client interface	Yes
26-71 IGMP-Access-Name	$junos-igmp-access-group-name	Access list to use for the group (G) filter	Yes
26-72 IGMP-Access-Src-Name	$junos-igmp-access-source-group-name	Access List to use for the source group (S,G) filter	Yes
26-74 MLD-Access-Name	$junos-mld-access-group-name	Access list to use for the group (G) filter	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
26-75 MLD-Access-Src-Name	$junos-mld-access-source-group-name	Access List to use for the source group (S,G) filter	Yes
26-77 MLD-Version	$junos-mld-version	MLD protocol version	Yes
26-78 IGMP-Version	$junos-igmp-version	IGMP protocol version	Yes
26-97 IGMP-Immediate-Leave	$junos-igmp-immediate-leave	IGMP immediate leave	Yes
26-100 MLD-Immediate-Leave	$junos-mld-immediate-leave	MLD immediate leave	Yes
26-106 IPv6-Ingress-Policy-Name	$junos-input-ipv6-filter	Input filter to apply to client IPv6 interface	Yes
26-107 IPv6-Egress-Policy-Name	$junos-output-ipv6-filter	Output filter to apply to client IPv6 interface	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
26-108 CoS-Parameter-Type	$junos-cos-scheduler-map	(T01: Scheduler-map name) Name of scheduler map configured in traffic-control profile	Yes
	$junos-cos-shaping-rate	(T02: Shaping rate) Shaping rate configured in traffic-control profile	Yes
	$junos-cos-guaranteed-rate	(T03: Guaranteed rate) Guaranteed rate configured in traffic-control profile	Yes
	$junos-cos-delay-buffer-rate	(T04: Delay-buffer rate) Delay-buffer rate configured in traffic-control profile	Yes
	$junos-cos-excess-rate	(T05; Excess rate) Excess rate configured in traffic-control profile	Yes
	$junos-cos-traffic-control-profile	(T06: Traffic-control profile) Name of the traffic-control profile configured in a dynamic profile	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-cos-shaping-mode	(T07; Shaping mode) CoS shaping mode configured in a dynamic profile	Yes
	$junos-cos-byte-adjust	(T08; Byte adjust) Byte adjustments configured for the shaping mode in a dynamic profile	Yes
	$junos-cos-adjust-minimum	(T09; Adjust minimum) Minimum adjusted value allowed for the shaping rate in a dynamic profile	Yes
	$junos-cos-excess-rate-high	(T10; Excess rate high) Excess rate configured for high-priority traffic in a dynamic profile	Yes
	$junos-cos-excess-rate-low	(T11; Excess rate low) Excess rate configured for low-priority traffic in a dynamic profile	Yes
	$junos-cos-shaping-rate-burst	(T12; Shaping rate burst) Burst size configured for the shaping rate in a dynamic profile	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-cos-guaranteed-rate-burst	(T13; Guaranteed rate burst) Burst size configured for the guaranteed rate in a dynamic profile	Yes
26-130 Qos-Set-Name	$junos-interface-set-name	Name of an interface set configured in a dynamic profile	Yes
26-146 CoS-Scheduler-Pmt-Type	$junos-cos-scheduler	(Null: Scheduler name) Name of scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-tx	(T01: CoS scheduler transmit rate) Transmit rate for scheduler configured in a dynamic profile	Yes Available for multiple parameters: • Percent • Rate
	$junos-cos-scheduler-bs	(T02: CoS scheduler buffer size) Buffer size for scheduler configured in a dynamic profile	Yes Available for multiple parameters: • Percent • Temporal

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-cos-scheduler-pri	(T03: CoS scheduler priority) Packet-scheduling priority for scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-dropfile-low	(T04: CoS scheduler drop-profile low) Name of drop profile for RED loss-priority level low for scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-dropfile-medium-low	(T05: CoS scheduler drop-profile medium-low) Name of drop profile for RED loss-priority level medium-low for scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-dropfile-medium-high	(T06: CoS scheduler drop-profile medium-high) Name of drop profile for RED loss-priority level medium-high for scheduler configured in a dynamic profile	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-cos-scheduler-dropfile-high	(T07: CoS scheduler drop-profile high) Name of drop profile for RED loss-priority level `high` for scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-dropfile-any	(T08: CoS scheduler drop-profile any) Name of drop profile for RED loss-priority level `any` for scheduler configured in a dynamic profile	Yes
	$junos-cos-scheduler-excess-rate	(T09: CoS scheduler excess rate) Excess rate configured for a scheduler in a dynamic profile	Yes Available for multiple parameters: • Percent • Proportion
	$junos-cos-scheduler-shaping-rate	(T10: CoS scheduler shaping rate) Shaping rate configured for a scheduler in a dynamic profile	Yes Available for multiple parameters: • Percent • Rate

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-cos-scheduler-excess-priority	(T11: CoS scheduler excess priority) Excess priority configured for a scheduler in a dynamic profile	Yes
26-165 Hierarchical-Policer-Parameter-Type (Junos OS Evolved)	$junos-hpolicer-high-cir	(T01: high committed information rate) The committed information rate (CIR) for high-priority traffic in a hierarchical policer.	Yes
	$junos-hpolicer-high-max-cir	(T02: high max committed information rate) The maximum committed information rate for high-priority traffic, which allows rate limiting up to this value.	Yes
	$junos-hpolicer-high-committed-burst	(T03: high committed burst size) The committed burst size for high-priority traffic, which limits the amount of burst traffic allowed.	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-hpolicer-med-high-cir	(T04: med-high committed information rate) The CIR for medium-high priority traffic.	Yes
	$junos-hpolicer-med-high-max-cir	(T05: med-high max committed information rate) The maximum CIR for medium-high priority traffic.	Yes
	$junos-hpolicer-med-high-committed-burst	(T06: med-high committed burst size) The burst size for medium-high priority traffic.	Yes
	$junos-hpolicer-med-low-cir	(T07: med-low committed information rate) The CIR for medium-low priority traffic.	Yes
	$junos-hpolicer-med-low-max-cir	(T08: med-low max committed information rate) The maximum CIR for medium-low priority traffic.	Yes

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
	$junos-hpolicer-med-low-committed-burst	(T09: med-low committed burst size) The burst size for medium-low priority traffic.	Yes
	$junos-hpolicer-low-cir	(T10: low committed information rate) The CIR for low-priority traffic.	Yes
	$junos-hpolicer-low-max-cir	(T11: low max committed information rate) The maximum CIR for low-priority traffic.	Yes
	$junos-hpolicer-low-committed-burst	(T12: low committed burst size) The burst size for low-priority traffic.	Yes
26-191 Input-Interface-Filter	$junos-input-interface-filter	Name of an input filter to be attached to a `family any` interface.	No
26-192 Output-Interface-Filter	$junos-output-interface-filter	Name of an output filter to be attached to a `family any` interface.	No

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
26-196 IPv4-Input-Service-Set	$junos-input-service-set	Name of an IPv4 input service set to be attached.	No
26-197 IPv4-Output-Service-Set	$junos-output-service-set	Name of an IPv4 output service set to be attached.	No
26-198 IPv4-Input-Service-Filter	$junos-input-service-filter	Name of an IPv4 input service filter to be attached.	No
26-199 IPv4-Output-Service-Filter	$junos-output-service-filter	Name of an IPv4 output service filter to be attached.	No
26-200 IPv6-Input-Service-Set	$junos-input-ipv6-service-set	Name of an IPv6 input service set to be attached.	No
26-201 IPv6-Output-Service-Set	$junos-output-ipv6-service-set	Name of an IPv6 output service set to be attached.	No
26-202 IPv6-Input-Service-Filter	$junos-input-ipv6-service-filter	Name of an IPv6 input service filter to be attached.	No

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
26-203 IPv6-Output-Service-Filter	$junos-output-ipv6-service-filter	Name of an IPv6 output service filter to be attached.	No
26-204 Adv-Pcef-Profile-Name	$junos-pcef-profile	Name of a PCEF profile to be attached.	No
26-205 Adv-Pcef-Rule-Name	$junos-pcef-rule	Name of a PCC rule to activate.	No
26-211	$junos-inner-vlan-tag-protocol-id	Name of VLAN map to activate	Yes
26-212	$junos-routing-services	Enables or disables routing services capability. If you enable this variable in your configuration and RADIUS does not return the VSA, then routing services are disabled for the subscriber.	No

Table 10: RADIUS Attributes and Corresponding Predefined Variables *(Continued)*

Attribute Number and Name	Junos OS Predefined Variable	Description	Default Value Support for Junos OS Predefined Variable
26-213	$junos-interface-set-target-weight	Specify a weight for an interface set to associate it and its member links with an aggregated Ethernet member link for targeted distribution.	Yes
26-214	$junos-interface-target-weight	Specify a weight for an interface to associate it with an interface set and thus with the set's aggregated Ethernet member link for targeted distribution. When an interface set does not have a weight, then the interface weight value for the first authorized subscriber interface is used for the set.	Yes

Predefined Variable Defaults for Dynamic Client Profiles

IN THIS SECTION

-

You can optionally configure default values for many predefined variables. The Junos OS uses the default value in the following cases:

- When the external RADIUS server is not available

- When the VSA returned by the RADIUS server does not contain a value for the predefined variable

The RADIUS value for a predefined variable takes precedence over the default value. For example, if you have configured a default for a predefined variable, but RADIUS also returns a value, the system uses the value from RADIUS instead.

The default value must be appropriate to the variable, such as an integer or an alphanumeric string. Starting in Junos OS Release 19.3R1, you can also configure the default value of a predefined variable to be another predefined variable by using a variable expression. In earlier releases, the default value must be fixed; it cannot be a variable.

Expressions are typically configured for user-defined variables and dynamic service profiles. See for more information.

When you use a variable expression, you are setting up a condition that determines the default value of the predefined variable. The value of the default is different when the condition is matched than when it is not matched. In dynamic client profiles, you can configure any of the following operators for variable expressions:

- `equals`—Assigns a predefined variable as the default value.

- `ifNotZero(`*parameter-1*`, `*parameter-2*`)`—Sets a condition to be matched. Assigns the value from *parameter-2* as the default value only when *parameter-1* is nonzero, meaning that the parameter resolved to some value.

- `ifZero(`*parameter-1*`, `*parameter-2*`)`—Sets a condition to be matched. Assigns the value from *parameter-2* as the default value only when *parameter-1* is zero, meaning that the parameter did not resolve to any value. If *parameter-1* did resolve to a value (therefore it is not zero), then the value from *parameter-1* is assigned as the default.

You can also nest expressions, which provides additional conditions for setting the variable value. For example, a dynamic profile for a subscriber in a heterogeneous network might have the following configuration for the `predefined-variable-defaults` statement:

```
predefined-variable-defaults {
    aggregation-interface-set-name equals "$junos-phy-ifd-underlying-intf-set-name";
    interface-set-name equals "ifZero($junos-default-interface-set-name, $junos-phy-ifd-
interface-set-name)";
    default-interface-set-name equals "ifZero($junos-interface-set-name, ifNotZero($junos-
```

```
aggregation-interface-set-name, $junos-aggregation-interface-set-name##'-default'))";
 }
```

See *Dynamic Level 2 and Level 3 Interface Set Naming with Predefined Variables* for a detailed explanation of how to evaluate these expressions in the context of a heterogeneous network.

Change History: Predefined Variable Defaults for Dynamic Client Profiles

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 11: Change History: Predefined Variable Defaults for Dynamic Client Profiles

Release	Description
19.3	Starting in Junos OS Release 19.3R1, you can also configure the default value of a predefined variable to be another predefined variable by using a variable expression.

Configuring Predefined Dynamic Variables in Dynamic Profiles

This topic discusses how to configure predefined variables in a dynamic profile. The dynamic profile obtains and replaces data for these variables from an incoming client data packet. You can specify these variables in the body of a dynamic profile without having to first define the variables at the `[edit dynamic-profiles` *profile-name* `variables]` hierarchy level.

Before you configure dynamic variables:

1. Create a basic dynamic profile.

   See "Configuring a Basic Dynamic Profile" on page 52.

2. Ensure that the router hardware is configured in the network to accept subscriber access.

To configure predefined variables in a dynamic profile:

1. Access the desired dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
[edit dynamic-profiles profile-name]
```

2. Configure the necessary variables (in this example, for a protocol interface).

```
[edit dynamic-profiles profile-name]
user@host# set protocols protocol-name interface predefined-variable-name
```

For example, the following simple configuration uses a predefined variable to dynamically create the interface accessed by the IGMP client, enabling IGMP on the subscriber interface:

```
[edit]
user@host# set dynamic-profiles igmp1-prof protocols igmp interface $junos-interface-name
```

For a complete list of supported predefined variables, see "Junos OS Predefined Variables" on page 60.

Configuring Default Values for Predefined Variables in a Dynamic Profile

For any Junos OS predefined variable that can be sourced from RADIUS, you can specify a default value in a dynamic client profile. These default values are used when RADIUS does not supply a value.

Defining default values for these predefined variables enables you to determine whether to source values locally from the profile instead of only from RADIUS. This enables you to use RADIUS as a way to selectively override predefined variable values, instead of being the sole source of those values.

For a list of predefined variables and options for which you can configure default values, see "Junos OS Predefined Variables That Correspond to RADIUS Attributes and VSAs" on page 98.

To configure default values for Junos predefined variables:

1. Specify that you want to configure the dynamic client profile.

```
[edit]
user@host# edit dynamic-profile profile-name
```

2. Configure the default value for a predefined variable or for a specific option within a predefined variable.

```
[edit dynamic-profiles profile-name]
user@host# set predefined-variable-defaults predefined-variable variable-option default-value
```

> **ℹ NOTE**: When you specify the *predefined-variable*, you use the name of the Junos OS predefined variable, but you omit the leading `$junos-` prefix.

For example, consider the behavior when you have the following configuration to specify a default value for the $junos-routing-instance predefined variable:

```
[edit dynamic-profiles prof1]
user@host# set predefined-variable-defaults routing-instances RI-def
```

- When RADIUS does not return a routing instance, the subscribers come up in the RI-def routing instance.

- When RADIUS returns routing-instance RI-res, the subscribers come up in the RI-res routing instance.

When you do not configure a default value for the $junos-routing-instance predefined variable and RADIUS does not return a value, the subscribers come up in the master routing instance, which is the Junos OS default.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.4R1Evo	Starting in Junos OS Evolved Release 25.4R1, specify the predefined variables when you configure enhanced hierarchical policer.
25.4R1Evo	Starting in Junos OS Evolved Release 25.4R1, you can configure default value for a predefined variable.
19.3R1	Starting in Junos OS Release 19.3R1, you can also configure the default value of a predefined variable to be another predefined variable by using a variable expression.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv6 input service filter to be attached.

17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv6 service set to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv4 input service filter to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv4 input service set to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv6 service filter to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv6 service set to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv4 service filter to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of an IPv4 output service set to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of a PCEF profile to be attached.
17.2R1	Starting in Junos OS Release 17.2R1, name of a PCC rule to activate.
16.1R4	Starting in Junos OS Release 16.1R4, identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs.
16.1R4	Starting in Junos OS Release 16.1R4, identifier for the inner VLAN tag for Layer 2 wholesale, ANCP-triggered, autosensed dynamic VLANs.
16.1	Starting in Junos OS Release 16.1, when you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure an IPv4 preferred source address.
16.1	Starting in Junos OS Release 16.1, when you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure an IPv6 preferred source address.
16.1	Starting in Junos OS Release 16.1, you can reduce the CoS resources wasted on residential access by collecting the residential subscribers into an interface set associated with the physical interface.
16.1	Starting in Junos OS Release 16.1, when you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the $junos-routing-instance predefined variable, you must not configure a preferred source address.

16.1	Starting in Junos OS Release 16.1, when you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the `$junos-routing-instance` predefined variable, you must not configure an IPv6 preferred source address.
16.1	Starting in Junos OS Release 16.1, you can reduce the CoS resources wasted on residential access by collecting the residential subscribers into an interface set associated with the physical interface.
16.1	Starting in Junos OS Release 16.1, when you specify a static logical interface for the unnumbered interface in a dynamic profile that includes the `$junos-routing-instance` predefined variable, you must not configure a preferred source address.

RELATED DOCUMENTATION

User-Defined Variables in Dynamic Profiles

IN THIS SECTION

User-Defined Variables

In dynamic service profiles, the Junos OS enables you to configure custom variables at the `[edit dynamic-profiles` *`profile-name`* `variables]` hierarchy level and use those variables in the `[edit dynamic-profiles]` hierarchy. The dynamic profile obtains and replaces data for these variables from an external server (for example, RADIUS) during the subscriber authentication process. At run time, the variables are replaced by actual values and used to configure subscriber interfaces.

You can use any of the following statements to configure user-defined variables:

- `default-value`—Configure a default value for a user-defined variable in a dynamic profile. The values that the system uses for these variables are applied when the subscriber authenticates. Specifying a default value provides a standalone configuration for the associated statement or a backup for the statement configuration if the external server is inaccessible or does not contain a value for the variable.

- `equals`—Configure an expression for a user-defined variable that is evaluated at run time and returned as the variable value.

- `mandatory`—Specify that an external server (for example, RADIUS) must return a value for the user-defined variable. If the external server does not return a value for the variable, the dynamic profile fails.

- `uid`—Configure a unique ID for parameterized filters and CoS in a dynamic profile created for services.

- `uid-reference`—Configure a variable that references a unique ID for parameterized filters or CoS in a dynamic profile created for services.

> (i) **NOTE**: The order in which you define how variables are obtained is important. To ensure that you obtain any mandatory variables from an external server, and not derive values from defaults or through variable expressions, you must define any mandatory variables first.

Configuring User-Defined Dynamic Variables in Dynamic Profiles

This topic discusses how to configure a user-defined dynamic variable in a dynamic service profile. You can define a variable at the `[edit dynamic-profiles` *`profile-name`* `variables]` hierarchy level that is used elsewhere in the dynamic service profile. You can optionally specify a default value for any dynamic variable that appears in the body of the dynamic profile. The default variable values are used in the event the router is unable to access an external server (for example, RADIUS) or otherwise obtain a

value for use as the dynamic variable. Alternatively, you can specify that using a RADIUS-returned value is mandatory; if that value is not received, then the profile fails.

Before you configure any dynamic variable default values:

1. Create a basic dynamic profile.

   See "Configuring a Basic Dynamic Profile" on page 52.

2. Ensure that the router is configured to enable communication between the client and the RADIUS server.

   See *Specifying the Authentication and Accounting Methods for Subscriber Access*.

3. Configure all RADIUS values that you want the profiles to use when validating subscribers.

   See *RADIUS Servers and Parameters for Subscriber Access*

To configure variables in a dynamic service profile:

1. Access the `variables` stanza in the desired dynamic service profile.

   ```
   [edit]
   user@host# edit dynamic-profiles Profile1 variables
   ```

2. Define the variable.

   ```
   [edit dynamic-profiles Profile1 variables]
   user@host# set video-filter equals "'Filter1'"
   ```

3. (Optional) Specify a default value for use by the variable in the event the router cannot contact the external server or if the external server does not contain a value for the assigned attribute.

   ```
   [edit dynamic-profiles Profile1 variables]
   user@host# set video-filter default-value Filter_default
   ```

4. (Optional) Specify that the external server must return a value for a user-defined variable. When you include the `mandatory` statement, if the external server does not return a value for the variable, the dynamic service profile fails.

   ```
   [edit dynamic-profiles Profile1 variables]
   user@host# set video-filter mandatory
   ```

Using Variable Expressions in User-Defined Variables

IN THIS SECTION

Junos OS enables you to create expressions—groups of arithmetic operators, string operators, and operands—for use as variables within dynamic profiles. You configure variable expressions at the `[dynamic-profiles `*`profile-name`*` variables]` hierarchy level. At run time, the variable expressions are calculated and used as variable values to configure dynamic subscriber interfaces.

When configuring expressions in dynamic profiles, you must adhere to the following rules:

- You can configure expressions only within a variable stanza of a dynamic profile.

- Dynamic profiles that contain expressions for user-defined variables must be used only for service activation.

- You generally assign expressions only to user-defined variables. You cannot assign expressions to internal variables or predefined variables.

- Expression values are given precedence over default values.

- Entire expressions must be contained within quotation marks (" ").

- Strings within the expressions must be quoted within single quotation marks (' ') and the single quotation marks can contain only strings.

- White space is treated as a delimiter for all operands and operators. Strings containing spaces that you create within expressions are treated as single strings and include any leading or trailing white space. For example, to create a service profile:

```
dynamic-profiles {
    service profile {
        variables {
            scheduler-name;
            video-filter equals   " '  Filter 1   ' "    # Everything within the single
quotation marks is considered a string, including the leading and trailing white space
        }
```

```
    }
 }
```

- The expression must be either all arithmetic operators or all string operators; mixing arithmetic operators and string operators is not allowed unless properly converted to the correct type.

- Expressions can refer to other system predefined variables or other user-defined variables. However, no circular referencing between variables is allowed. For example, the following reference is incorrect:

```
dynamic-profiles {
    Service_Profile_1 {
        variables {
            scheduler-name;
            transmit-rate2 equals   " ( $transmit-rate1 * 2)/3"   # refers to transmit-rate1
            transmit-rate1 equals   " ( $transmit-rate2 * 2)/3"     # refers to transmit-rate2
        }
    }
}
```

- To create a dynamic interface and interface-set for BNG-CUPS user plane, use the following format:

```
dynamic-profile {
    <dprof_name> {
        interfaces {
            interface-set <dynamic interface-set Name> {
                interface <interface-name> {
                    unit "$junos-interface-unit";
                }
            }
         …
        class-of-service {
        …
            interfaces {
                interface-set <dynamic interface-set Name> {
                    output-traffic-control-profile <TCP Name>;
                }
            }
```

```
     }
  }
```

You can configure service filter precedence for BNG-CUPS user plane, to control the service
evaluation order for the packets. You can add the precedence configuration to the dynamic profile.

```
dynamic-profiles {
    <dprof_name> {
        ...
        interfaces {
            pp0 {
                unit "$junos-interface-unit" {
                    ...
                    family inet {
                        filter {
                            input "$in-filter" precedence 20;
                            output "$out-filter" precedence 20;
                        }
                        unnumbered-address "$junos-loopback-interface";
                    }
                    family inet6 {
                        filter {
                            input "$inv6-filter" precedence 20;
                            output "$outv6-filter" precedence 20;
                        }
                        unnumbered-address "$junos-loopback-interface";
                    }
                }
            }
        }
        ...
    }
}
```

- Any mandatory variable that does not contain a "default" value or an "equals" expression must
  contain a value as a part of service activation. For example, a RADIUS service VSA like "service-
  video(value1, value2)" that contains two or fewer mandatory variables in the dynamic service profile
  definition "service-video" succeeds. The service activation fails if at least one mandatory variable
  does not have any value associated with it, either through "default" or "equals" attribute evaluation.

Table 12 on page 125 lists supported operators and functions you can use to create expressions.
Precedence 5 is the highest level.

Table 12: Operators and Functions

Operation	Operator	Associativity	Precedence	Action
Arithmetic Addition	+	Left	1	Adds the elements to the right and left of the operator together.
Arithmetic Subtraction	-	Left	1	Subtracts the element to the right of the operator from the element to the left of the operator.
Arithmetic Multiplication	*	Left	2	Multiplies the element to the left of the operator by the element to the right of the operator.
Arithmetic Division	/	Left	2	Divides the element to the left of the operator by the element to the right of the operator.
Arithmetic Modulo	%	Left	2	Divides the element to the left of the operator by the element to the right of the operator and returns the integer remainder. If the element to the left of the operator is less than the element to the right of the operator, the result is the element to the left of the operator.
Concatenation	##	Left	3	Creates a new string by joining the string values to the left of the operator and the values to the right of the operator together.
Maximum	max(param1,param2)	Left	4	Takes the maximum of the two values passed as parameters.
Minimum	min(param1,param2)	Left	4	Takes the minimum of the two values passed as parameters.

Table 12: Operators and Functions *(Continued)*

Operation	Operator	Associativity	Precedence	Action
Round	round(param1)	-	4	Rounds the value to the nearest integer.
Truncate	trunc(param1)	-	4	Truncates a non-integer value to the value left of the decimal point.
Convert to String	toStr(param1)	-	4	Converts the variable inside the parentheses to a null terminated string.
Convert to Integer	toInt(param1)	-	4	Converts the parameter to an integer. A single string or variable is allowed as a parameter.
Random	rand()	-	4	Generates a random numerical value.
If Not Zero	ifNotZero(param1, param2)	Left	4	Returns the second parameter if the first parameter is not zero. Returns NULL if first parameter is zero.
Parentheses	( )	-	5	Groups operands and operators to achieve results different from simple precedence; effectively has the highest precedence.

Expressions are evaluated after variables are populated with values. The evaluation is conducted immediately before profile instantiation and includes value checking. If the computed values are not acceptable, or rules governing expression syntax are broken, the expression evaluation fails, profile instantiation does not occur, and messages are logged to describe the errors.

lists the possible expression error scenarios and the action taken by the router software.

Table 13: Expression Errors and Actions

Error	Occurance	Action	Variable Value
Parsing error	Commit check phase	Commit fails	not applicable
Circular variable dependency error	Commit check phase	Commit fails	not applicable
Variables inside the expressions are not defined	Commit check phase	Commit fails	not applicable
Divide by zero	Profile Instantiation	Profile instantiation fails	Zero (0)
Adding string to a number	Profile Instantiation	Profile instantiation fails	Zero (0)
Overflow error	Profile Instantiation	Profile instantiation fails	Undefined
Underflow error	Profile Instantiation	Profile instantiation fails	Undefined

You can also configure the user-defined variables with a default value. The default value provides a standalone configuration for the associated statement or a backup for the statement configuration if the RADIUS server is inaccessible or the VSA attribute does not contain a value.

Change History for Variable Expressions in User-Defined Variables

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 14: Change History for Variable Expressions in User-Defined Variables

Release	Description
19.3	Starting in Junos OS Release 19.3R1, you can configure expressions in the `predefined-variable-defaults` statement in a dynamic profile. See "Predefined Variable Defaults for Dynamic Client Profiles" on page 113. You can also configure a limited number of expressions to establish default values for predefined variables. See "Predefined Variable Defaults for Dynamic Client Profiles" on page 113.
25.2 Junos OS Evolved	Starting in Junos OS Evolved Release 25.2R1, you can configure interface-set and family (inet, inet6) input or output filter precedence for dynamic profiles in the BNG-CUPS user plane mode.

SEE ALSO

interface-set (Dynamic Profiles)

Defining Dynamic Filter Processing Order

precedence

Configuring Variable Expressions in Dynamic Profiles

You can create expressions—groups of arithmetic operators, string operators, and operands—for use as variables within dynamic profiles. These expressions are used as variable values to configure dynamic subscriber interfaces.

To configure dynamic profile variable expressions:

1. Access the dynamic profile for which you want to create variable expressions.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Access the `variables` hierarchy for the dynamic profile.

```
[edit dynamic-profiles profile-name]
user@host# edit variables
```

3. Define the variable using the expression operators and operands described in "Using Variable Expressions in User-Defined Variables" on page 122.

```
[edit dynamic-profiles profile-name variables]
user@host# set expression
```

Table 15 on page 129 provides several examples of expressions that you can create using the supported operators and functions.

Table 15: Expression Examples

Example	Description
video-filter equals "' Filter1' "	Assigns the string " Filter1" to the dynamic $video-filter variable.
video-filter2 equals "$video-filter ## ' Filter2' "	Converts dynamic variable "$video-filter" to a string and concatenates the new string with the string " Filter2". The result is the string "$video-filter Filter2" assigned to the $video-filter2 variable.
tempvar equals "120"	Converts "120" to an integer and assigns the integer to the $tempvar variable.
transmit-rate2 equals " ($transmit-rate1 * 2)/3 + $tempvar)"	Multiplies the "transmit-rate1" variable by 2 and divides that value by the sum of 3 and the value of "$tempvar". The result is assigned to the $transmit-rate2 variable.
host-ip equals " '203.0.113.2' "	Assigns the string "203.0.113.2" to the $host-ip variable.
max-val "max($max1,$max2)"	Assigns the greater of value "max1" or "max2" to the $max-val variable.
min-val "$min($var1,30)"	Assign the smaller of value "var1" and "30" to the $min-val variable.

Table 15: Expression Examples *(Continued)*

Example	Description
rounded-var equals "round($var1)"	Rounds off the value of the variable "$var1" to the nearest integer and assigns the value to the $rounded-var variable.
trunc-var equals "trunc(1234.5)"	Truncates the value in parentheses to the left side of the decimal and assigns the resulting value to the $trunc-var variable.
bwg-shaping-rate equals "$ancp-downstream - ($ancp-downstream % 2 * (1 - $sp-qos-cell-mode))"	Evaluates the expression as per the precedence set in the parentheses.
temp-filter1 equals " 'Filter1' ## toStr($filter)"	Converts the "$filter" variable to a string value and concatenates the converted string to the string "Filter1". The resulting combined string is assigned to the $temp-filter1 variable.

Conditional Configuration for Dynamic Profile Overview

You can configure conditional configuration statements for dynamic profiles to dynamically obtain subscriber information for a client or service.

Conditional configuration involves two main steps:

1. Defining the conditional variable

2. Referencing the conditional variable in a *configuration statement*

A conditional variable is defined as an expression **ifNotZero** *(param1, param2)*. In this expression, *param1* is a user-defined variable whose value is derived from an external server such as RADIUS and *param2* can be a user-defined variable, a function, operation, number, or string. A conditional variable can be user-defined or Unique ID (UID) *reference* variable. It cannot be a predefined or UID variable. In Junos OS, conditional variables are supported only for the service dynamic profiles.

The configuration statements in which the conditional variables are referenced are called *conditional* configuration statements. After the conditional variable are defined, they are referenced in **dynamic-profiles** configuration statements and are processed when the service profile is instantiated. The following service profile configuration statements support conditional variables:

- dynamic-profiles *profile-name* interfaces *interface-name* unit *unit-no* family *type* filter input *filter-name*

- dynamic-profiles *profile-name* interfaces *interface-name* unit *unit-no* family *type* filter output *filter-name*

- dynamic-profiles *profile-name* firewall family *type* filter *filter-name* term *term-name*

- dynamic-profiles *profile-name* firewall family *type* filter *filter-name* term *term-name* then policer *policer-name*

- dynamic-profiles *profile-name* firewall family *type* filter *filter-name* term *term-name* then hierarchical-policer *policer-name*

- dynamic-profiles *profile-name* class-of-service scheduler-maps *map-name* forwarding-class *class-name* scheduler *scheduler-name*

The system follows the following set of rules while evaluating the conditional variables and conditional configuration statements during service profile instantiation:

- In the function ifNotZero(*param1*, *param2*), if the value of a *param1* is not received from an external server and if the default value is not configured, the value of the variable is treated as non-zero and *param2* is evaluated.

- If the value of *param1* in the function ifNotZero(*param1*, *param2*) is 0, then NULL is returned as the value of the expression and *param2* is not evaluated. In this case, the value of the conditional variable becomes NULL and the configuration statement in which the conditional variable is referenced is ignored.

- If the value of *param1* is non-zero, then *param2* is evaluated and its value is returned as the value of the expression.

The following **filter-service** and **cos-service** configuration examples show how the rules are applied:

Filter Service Configuration Example

```
filter-service {
    variables {
        input-filter-var mandatory;
        output-filter-var mandatory;
        bw-limit-var  mandatory;
        term1-var default-value term1;
        input-filter-ref {
            equals "ifNotZero($input-filter-var,$input-filter-var)";
            uid-reference;
        }
        output-filter-ref {
            equals "ifNotZero($output-filter-var,$output-filter-var)";
            uid-reference;
```

```
            }
        policer1-ref {
            equals "ifNotZero($bw-limit-var,'policer1')";
            uid-reference;
        }

        term1 equals "ifNotZero($term1-var,$term1-var)";
        input-filter uid;
        output-filter uid;
        policer1 uid;
    }
    interfaces {
        pp0 {
            unit "$junos-interface-unit" {
                family inet {
                    filter {
                        input "$input-filter-ref" precedence 50;
                        output "$output-filter-ref" precedence 50;
                    }
                }
            }
        }
    }
    firewall {
        family inet {
            filter "$input-filter" {
                interface-specific;
                term $term1 {
                    then {
                        policer "$policer1-ref";
                        service-accounting;
                    }
                }
                term rest {
                    then accept;
                }
            }
            filter "$output-filter" {
                interface-specific;
                term rest {
                    then accept;
                }
            }
```

```
        }
        policer "$policer1" {
            if-exceeding {
                bandwidth-limit "$bw-limit-var";
                burst-size-limit 15k;
            }
            then discard;
        }
    }
 }
```

In the **filter-service** configuration example, `input-filter-ref`, `output-filter-ref`, `policer1-ref`, and `term1` are conditional variables while `input "$input-filter-ref" precedence 50`, `output "$output-filter-ref" precedence 50`, `term $term1`, and `policer "$policer1-ref"` are conditional configuration statements. In this example, if the value of `input-filter-var` is 0, the value of the conditional variable `input-filter-ref` becomes NULL. Thus the entire configuration statement, `input "$input-filter-ref" precedence 50`, in which the conditional variable is referenced, is ignored. If , however, the value of the variable is non-zero, the configuration statement is processed during the service profile instantiation.

CoS Service Configuration Example

```
cos-service {
    variables {
        sch1_var mandatory;
        sch2_var mandatory;
        sch1_ref {
            equals "ifNotZero($sch1_var,$sch1_var)";
            uid-reference;
        }
        sch2_ref {
            equals "ifNotZero($sch2_var,$sch2_var)";
            uid-reference;
        }
        smap1 uid;
        sch1 uid;
        sch2 uid;
    }
    class-of-service {
        scheduler-maps {
            "$smap1" {
                forwarding-class best-effort scheduler "$sch1_ref";
                forwarding-class assured-forwarding scheduler "$sch2_ref";
```

```
                }
            }
        schedulers {
            "$sch1" {
                transmit-rate percent 30;
                buffer-size percent 30;
                priority low;
            }
            "$sch2" {
                transmit-rate percent 10;
                buffer-size percent 10;
                priority high;
            }
        }
    }
 }
```

In the **cos-service** configuration example, `sch1_ref` and `sch2_ref` are conditional variables while `forwarding-class best-effort scheduler "$sch1_ref"` and `forwarding-class assured-forwarding scheduler "$sch2_ref"` are conditional configuration statements. Similar to the evaluation in the **filter-service** configuration example, if the value of any variable, referenced in a conditional variable is 0, the configuration statement in which the conditional variable is referenced is ignored and not processed during CoS service profile instantiation.

RELATED DOCUMENTATION

Standard and Vendor-Specific RADIUS Attributes

RADIUS Servers and Parameters for Subscriber Access

Versioning for Dynamic Profiles

Enabling Dynamic Profiles to Use Multiple Versions

You can create new versions of dynamic profiles that are currently in use by subscribers. Any subscriber that logs in following a dynamic profile modification uses the latest version of the dynamic profile. Subscribers that are already active continue to use the older version of the dynamic profile until they log out or their session terminates.

You must enable or disable dynamic profile version creation before creating or using any dynamic profiles on the router. Enabling or disabling dynamic profile version creation after dynamic profiles are configured is not supported.

To configure versioning for dynamic profiles:

1. Access the router system hierarchy level.

   ```
   [edit]
   user@host# edit system
   ```

2. Access the global dynamic profile options.

   ```
   [edit system]
   user@host# edit dynamic-profile-options
   ```

3. Enable version creation for dynamic profiles on the router.

   ```
   [edit system dynamic-profile-options]
   user@host# set versioning
   ```

Modifying Dynamic Profiles with Versioning Disabled

You use dynamic profiles to configure large groups of subscribers. However, after you have configured and applied dynamic profiles, be cautious when modifying any dynamic profiles that are in use by active subscribers on the router if you have not enabled the router to use dynamic profile versioning. This section provides guidelines and procedures for modifying existing profiles and applying them to subscriber interfaces if dynamic profile versioning is not enabled on the router.

When modifying dynamic profiles, keep the following considerations in mind:

- Do not modify a dynamic profile when dynamic profile versioning is disabled and the dynamic profile is in use by active subscribers.

- Modifying a dynamic profile when dynamic profile versioning is disabled and when the dynamic profile is in use by active subscribers can lead to unpredictable behavior.

When a dynamic profile is modified and committed when dynamic profile versioning is not enabled, the router:

1. Logs a warning that the profiles are being modified and committed.

2. Determines whether the profile is currently being use by any subscriber.

3. If the profile is in use by a subscriber, the commit fails and the router logs errors to report the conflict.

We recommend that you only modify dynamic profiles when you have enabled dynamic profile versioning on the router. However, to properly modify a dynamic profile when dynamic profile versioning is disabled on the router:

1. Ensure that no subscribers are using the dynamic profile.
2. Create a new dynamic profile with a different name that contains the desired changes:
   Original Profile

```
profile1 {
    interfaces {
        "$junos-interface-ifd-name" {
            unit "$junos-underlying-interface-unit" {
                family inet {
                    filter {
                        input "$junos-input-filter";
                    }
                }
            }
```

```
        }
    }
}
```

Original DHCP Configuration

```
forwarding-options {
    dhcp-relay {
        traceoptions {
            flag all;
        }
        .......
        dynamic-profile profile1;
        ........
    }
}
```

New Profile

```
profile2 {
    interfaces {
        "$junos-interface-ifd-name" {
            unit "$junos-underlying-interface-unit" {
                family inet {
                    filter {
                        input "$junos-input-filter";
                        output "$junos-output-filter; /* added output filter variable */
                    }
                }
            }
        }
    }
}
```

Modified DHCP Configuration

```
forwarding-options {
    dhcp-relay {
        traceoptions {
            flag all;
        }
```

```
      .......
      dynamic-profile profile2; /* Name changed from profile1 */
      ........
   }
}
```

3. Commit the configuration containing the modified profile.

   The modified profile is used for any new subscribers that access the router.

Distinguishing Profile Versions with a Configurable Alias

You can configure a version alias to identify a specific configuration variant of a base dynamic client profile. The version alias is a text description that lets you decide how to name different profile variants, so they have an identifier independent of the dynamic version name that is automatically created by the BNG.

The need for a version alias results from the practice of using a given base dynamic profile across multiple BNGs in a network. Dynamic versioning enables you to modify a base dynamic profile to provide specific capabilities to subscribers that subsequently log in with the base dynamic profile. The different variations might be for subscribers on different BNGs or to new subscribers on a given BNG.

Dynamic versioning assigns a name to each new variation in the base profile. Consequently, the version name may vary for subscribers on one BNG or across multiple BNGs. In either case, RADIUS cannot determine which version of a profile is in use by any subscriber. This creates an operational challenge because RADIUS is unable to return corresponding attributes and VSAs in a CoA message that are compatible with that version of the profile.

When you configure a version alias for a dynamic client profile, the BNG sends the version alias to the RADIUS server during authentication. It is conveyed in the Juniper Networks client-profile-name VSA (26–4874–174). The version alias is an independent tag that enables you to track which profile variations are in use. Because RADIUS can distinguish the different profile versions, you can normalize the RADIUS back-end configuration for efficient use of CoA messages.

By default, the Client-Profile-Name VSA carries the name of the base dynamic profile. The version alias string is concatenated to the end of the profile name in the VSA, like this:

client-profile-name:*version-alias-string*

- To configure a version alias for a dynamic client profile:

```
[edit dynamic-profiles profile-name]
user@host# set version-alias version-alias-string
```

- To display the alias for a dynamic client profile:

```
user@host> show subscribers detail

Type: PPPoE
User Name: DEFAULTUSER
IP Address: 192.0.2.21
IP Netmask: 255.255.255.255
IPv6 Address: 2001:db8::17
Logical System: default
Routing Instance: default
Interface: pp0.3221225720
Interface type: Dynamic
Underlying Interface: demux0.3221225719
Dynamic Profile Name: pppoe-client-profile
Dynamic Profile Version Alias: profile-version1a
MAC Address: 00:00:5E:00:53:38
State: Active
Radius Accounting ID: 288
Session ID: 288
PFE Flow ID: 344
VLAN Id: 1
Login Time: 2019-09-23 10:40:56 IST
```

RELATED DOCUMENTATION

Dynamic Profiles for Subscriber Management | 47

4

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements

- Operational Commands