

## Types of Internal Plug-Ins

---

There are two main types of plug-ins: authorization plug-ins and tracking plug-ins.

### Authorization Plug-Ins

Authorization plug-ins can perform both authentication (that is, verify the originator of a request) and authorization. Authorization can include the setting of service session parameters such as session timeout or authorizing services based on the current load of the router.

You can set up authorization plug-ins to:

- Globally authorize all subscriber sessions.
- Authenticate subscribers who belong to a particular retailer's domain.
- Globally authenticate and/or authorize all service sessions.
- Authenticate and/or authorize sessions for a particular service.
- Globally authorize DHCP address allocations.
- Authorize DHCP address allocation for subscribers who log in to a particular retailer's domain.
- Globally authorize subscribers to change their subscriptions.



**NOTE:** Event publishers send events to all configured plug-in instances. For authentication to succeed, all authentication plug-ins that receive the authentication request must grant authentication.

---

### Tracking Plug-Ins

Tracking plug-ins track activity or log accounting information. You can set up tracking plug-ins to:

- Globally track all subscribers.
- Track subscribers who belong to a particular retailer's domain.
- Globally track all service sessions.
- Track service sessions for individual services.
- Track QoS service sessions for individual services and attach the required QoS profile to the JUNOS subscriber interface.

Tracking plug-ins keep the state of active sessions and provide usage and accounting data. For each subscriber and service session, plug-ins can track when the session is activated and deactivated and can keep interim updates. For example, when the SAE activates a service, it sends a Service Session Start event to tracking plug-in instances that are registered to receive events for that service. When the service is stopped, the SAE sends a Service Session Stop event to all tracking plug-ins that received the Service Session Start event. If interim accounting is configured, service

session interim update events are sent at regular intervals to all tracking plug-ins that are registered to receive the event.

One application of tracking plug-ins is to keep usage records, such as session time and volume counters. Service-tracking plug-ins can set a timeout for a service session in response to start and interim updates that the plug-in receives for the session. When a service session is active longer than the defined timeout, the SAE stops the session and sends service session stop events to the tracking plug-ins.

Another application is to track QoS services and attach the required QoS profile to the subscriber interface. See Overview of QoS on JUNOS Routers.

### ***Customizing RADIUS Packets with Plug-Ins***

RADIUS internal plug-ins include flexible RADIUS plug-ins and custom RADIUS plug-ins that let you customize RADIUS authentication and accounting packets that the SAE sends to RADIUS servers. You can specify which fields are included in various types of RADIUS packets and what information is contained in the fields.

For example, you can specify values in authentication response packets that will set session and idle timeouts, set the RADIUS class, and set the session volume quota. For accounting packets, you can specify which fields to include in accounting records.

For DHCP subscribers, you can set up RADIUS authorization plug-ins to return to the router attributes that can be used to select a DHCP address or select a fixed address for each subscriber.

The main difference between flexible RADIUS plug-ins and custom RADIUS plug-ins is that custom plug-ins are designed to deliver better system performance than the flexible RADIUS plug-ins. To use a custom plug-in, you must provide a Java class that implements the SPI defined in the RADIUS client library. Use this SPI to specify which fields and field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core API.

To customize RADIUS packets with a flexible RADIUS plug-in, see Overview of Flexible RADIUS Plug-Ins.