

Chapter 27

Configuring Service Manager

This chapter describes how to use the Service Manager application to define, activate, and monitor networking services for your subscribers. This chapter discusses the following topics:

- Overview on page 534
- Platform Considerations on page 535
- References on page 535
- Configuration Tasks on page 536
- Service Definitions on page 537
- Referencing Policies in Service Definitions on page 541
- Referencing QoS Configurations in Service Definitions on page 542
- Configuring the Service Manager License on page 550
- Managing and Activating Service Sessions on page 551
- Using RADIUS to Manage Subscriber Service Sessions on page 551
- Using Mutex Groups to Activate and Deactivate Subscriber Services on page 557
- Configuring RADIUS Accounting for Service Manager on page 560
- Using the CLI to Manage Subscriber Service Sessions on page 564
- Configuring Service Manager Statistics on page 572
- Service Manager Performance Considerations on page 575
- Service Definition Examples on page 575

Overview

The JUNOS Service Manager application provides authentication, service selection, and service activation and deactivation to subscribers. The application also collects accounting information and statistics, and monitors subscriber and service sessions.

Service Manager supports two client types—RADIUS and CLI. Service Manager starts when it receives a request from a RADIUS or CLI client. For RADIUS clients, RADIUS Access-Accept messages and Change-of-Authorization-Request (CoA-Request) messages can create and delete Service Manager subscriber sessions and activate and deactivate service sessions. For CLI clients, CLI commands create and delete the subscriber sessions and activate and deactivate service sessions.

A subscriber's service is based on a service definition — service definitions can include profiles, policies, and quality of service (QoS) settings that define the scope of a service granted to the subscriber. Service definitions can also specify statistics configurations.

Service Manager provides convenience and flexibility to both service providers and subscribers.

- Providers are able to separate services and access technology and also to 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—subscribers can dynamically connect to and disconnect from the services, when they want and for how long they want. They are billed based on the service type and usage, rather than being charged a set rate regardless of usage.

Service Manager Terms and Acronyms

Table 107 defines terms and acronyms that are used in this discussion of the Service Manager application.

Table 107: Service Manager Terms and Acronyms

Term	Definition
Guided entrance	A service that creates a controlled Internet browsing environment by transparently directing the subscriber to a specific Web site. At the Web site, the subscriber is presented with a selection of available services. Also called <i>walled gardens</i> or <i>captive portals</i> .
Macro language	The JUNOS macro language that you use for service definitions
Mutex service	A service session that is part of a mutex group—the service definition for the service includes the mutex-group attribute.
RADIUS login method	The method that uses RADIUS VSAs in the Access-Accept packet to create a subscriber session and activate a service session when the subscriber logs in

Table 107: Service Manager Terms and Acronyms (continued)

Term	Definition
RADIUS CoA method	The method that uses RADIUS CoA-Request messages and VSAs to create a subscriber session and activate a service session for a subscriber that is already logged in
Service definition	A macro file that defines a named parameterized description of a service; used to create a service instance and the resulting subscriber service session; can include a combination of parameters such as policy lists, rate-limit profiles, QoS profiles, and interface profiles
Service instance	An instance that is created when you specify parameter values for a service definition to create a service session
Service session	A session that is created when a service instance is activated for a subscriber; a subscriber can have multiple active service sessions
Service session profile	A provider-configured profile that applies optional attributes to a service session; CLI only

Platform Considerations

Service Manager is supported on all E-series routers. For information about the modules supported on E-series routers:

- See the *ERX Module Guide* for modules supported on ERX-7xx models, ERX-14xx models, and the ERX-310 router.
- See the *E120 and E320 Module Guide* for modules supported on the E120 router and the E320 router.

References

For more information about the topics covered in this chapter, see the following documents:

- Data-Over-Cable Service Interface Specifications (DOCSIS) 2.0 Radio Frequency Interface Specification CM-SP-RF1v2.0-110-051209.
- For information about using the JUNOS software's macro language, see *JUNOS System Basics Configuration Guide, Chapter 10, Writing CLI Macros*.
- For information about setting up policy-based routing features for Service Manager, such as rate-limit profiles, classifier control lists, policy lists, and hierarchical and merged policies, see the *JUNOS Policy Management Configuration Guide*.
- For information about creating QoS profiles and QoS parameters, see the *JUNOS Quality of Service Configuration Guide*.
- For information about creating IPv4 interface profiles, see *JUNOS IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*.

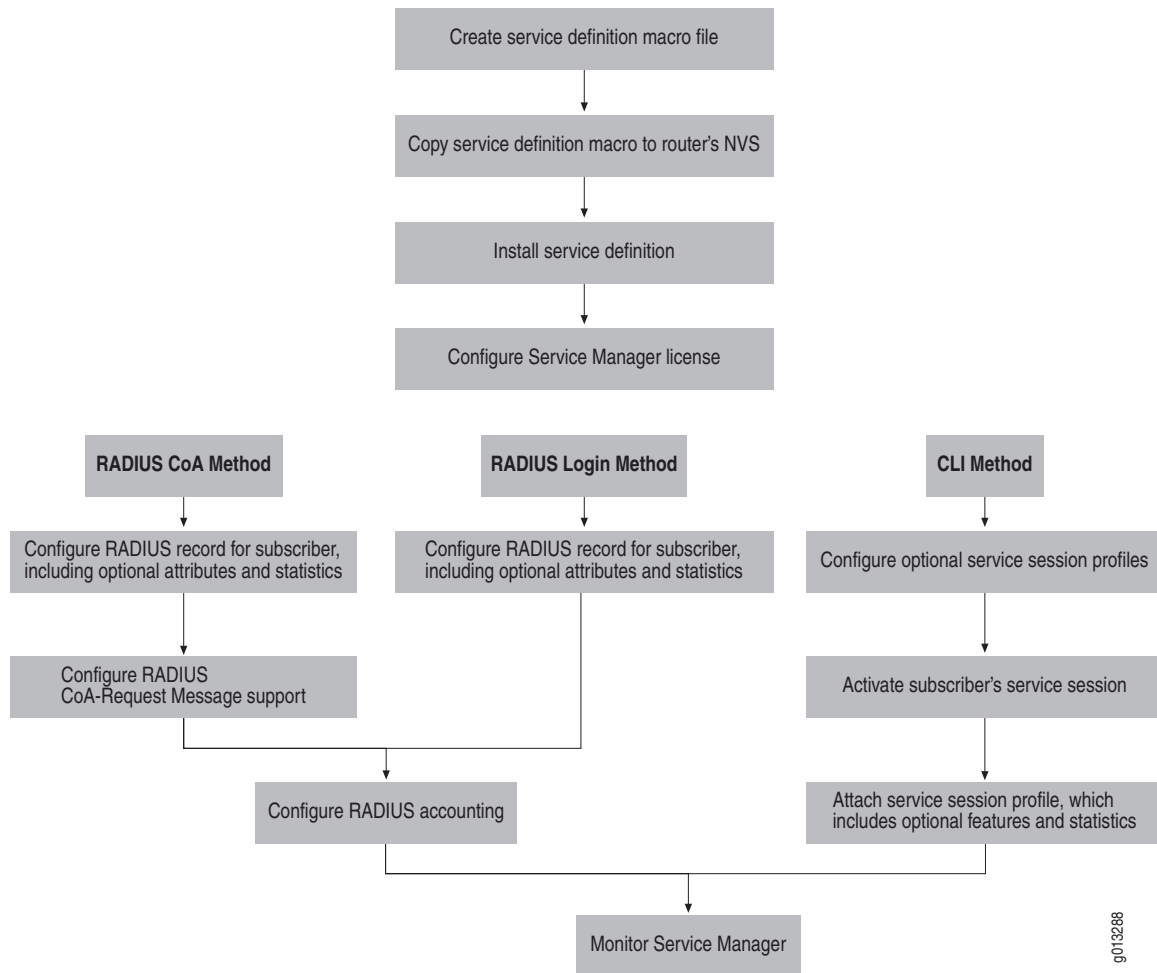
Configuration Tasks

To use the Service Manager application to create subscriber service sessions, you perform the following tasks:

- Create and manage service definitions
 - Use the macro language to define service definitions
 - Download service definition macro files to the router's nonvolatile storage (NVS)
 - Install service definitions on the router
 - Uninstall service definitions
- Configure the Service Manager license
- Configure RADIUS accounting
- Use RADIUS login and RADIUS CoA to manage subscriber service sessions
 - Specify the subscriber
 - Specify optional attributes
 - Enable statistics collection
 - Activate the service session
 - Deactivate service sessions
 - (Optional for RADIUS CoA method) Configure the CoA feature for the RADIUS dynamic-request server
- Use the CLI to manage subscriber service sessions
 - Specify the subscriber
 - Create and apply optional service session profiles
 - Enable statistics collection
 - Activate the service session
 - Deactivate service sessions

Figure 26 shows the sequence of operations you use to create and monitor subscriber service sessions.

Figure 26: Service Manager Configuration Flowchart



Service Definitions

A service definition is a high-level, platform-independent template that defines a service that you want to let your subscribers use. You use the JUNOS software's embedded macro language on your computer to create the macro file that defines the service. You copy and install the macro file on the E-Series router, and then you can associate the service definition with subscribers to create their service sessions.

Service definitions gives you flexibility by enabling you to use:

- A single service definition to create a service for multiple subscribers.
- Parameterized service definitions to create variations of a service definition.
- Different service definitions to create multiple services for a single subscriber.

A service definition might use the following types of JUNOS objects to define the characteristics and capabilities of the service you want to provide:

- Interface profiles—Specify a set of characteristics that can be dynamically assigned to IP interfaces. A service definition must use at least one interface profile.
- Policy lists—Specify policy actions for traffic traversing an interface.
- Classifier lists—Specify the criteria by which the router defines a packet flow.
- Rate-limit profiles—Specify a set of bandwidth attributes and associated actions that limit a classified packet flow or a source interface to a rate that is less than the physical rate of the port.
- QoS parameters—Specify attributes such as shaping rate, shared-shaping rate, assured rate, and scheduler weight for scheduler nodes and queues.
- QoS profiles—Specify queue, drop statistics gathering, and scheduler configuration for an interface hierarchy.

Creating Service Definitions

To create a service definition, you use the JUNOS software's macro language to specify the parameters that define the desired service. A macro file can define only one service—however, the file can have multiple templates to define characteristics of the desired service. You create service definitions independent of the Service Manager commands and operations, which are performed on the E-series router.

For detailed information about the JUNOS software's macro language, see *JUNOS System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

Figure 27 is an example of a service definition macro file that creates a tiered service. A tiered service typically provides set bandwidths for both inbound and outbound traffic for a subscriber. In this example, the input (inputBW) and output (outputBW) bandwidth values are parameterized. This example assumes that QoS profile triplePlayIP and QoS parameter maxSubscBW are configured. See *Service Definition Examples* on page 575 for additional service definition examples.

Service Manager only tracks JUNOS objects that are passed back in the `env.setResult` method when a service definition is executed. Table 108 describes the supported objects:

Table 108: JUNOS Objects Tracked by Service Manager

Name	Requirement	Description
input-stat-clacl	Optional	<ul style="list-style-type: none"> ■ Collects input statistics from policy manager ■ Can be a list of clacs
secondary-input-stat-clacl	Optional	<ul style="list-style-type: none"> ■ Collects input statistics from policy manager ■ Can be a list of clacs
output-stat-clacl	Optional	<ul style="list-style-type: none"> ■ Collects output statistics from policy manager ■ Can be a list of clacs

Table 108: JUNOS Objects Tracked by Service Manager (continued)

Name	Requirement	Description
activate-profile	Required	<ul style="list-style-type: none"> ■ Specifies the interface profile used on activation of the service ■ Deletion of the profile is Service Manager's responsibility
deactivate-profile	Optional	<ul style="list-style-type: none"> ■ Specifies the interface profile used on deactivation of the service ■ If not specified, is the same as the activation-profile ■ Deletion of the profile is Service Manager's responsibility
command-in-error	Optional	<ul style="list-style-type: none"> ■ Passes the value env.getErrorCommand ■ Service Manager displays the line in the service definition that has the error
command-error-status	Optional	<ul style="list-style-type: none"> ■ Passes the value env.getErrorStatus ■ Service Manager displays the error status for the error

Figure 27: Sample Service Definition Macro File

```

!parameterizes input and output bandwidth
<# tiered(inputBW, outputBW) #>

<# uid := app.servicemanager.getUniqueId #>
<# name := "SM-tiered-" $ uid #>
<# oname := "SM-O-tiered-" $ uid #>

classifier-list matchAll ip any any
rate-limit-profile <# name #> one-rate
    committed-rate <# inputBW; '\n' #>

policy-list <# name; '\n' #>
    classifier-group matchAll precedence 10000
    rate-limit-profile <# name; '\n' #>
    traffic-class best-effort

policy-list <# oname; '\n' #>
    classifier-group matchAll precedence 10000
    traffic-class best-effort

profile <# name; '\n' #>
    ip policy secondary-input <# name #> statistics enabled merge
    ip policy output <# oname #> statistics enabled merge
    qos-profile triplePlayIP
    qos-parameter maxSubscBW <# outputBW; '\n' #>

<# env.setResult("activate-profile", name) #>
<# env.setResult("secondary-input-stat-clacl", "matchAll") #>
<# env.setResult("output-stat-clacl", "matchAll") #>

<# endtmp1 #>

```

Managing Your Service Definitions

After you have created the macro file for your service definition, you can perform the following operations with the service definition macro file:

1. **Copy**—You must copy the service definition from the local computer that you used to create the macro file to the router's NVS card.
2. **Install**—You must install the service definition before you can use it to create a service session. During installation, Service Manager precompiles the definition and extracts the definition file's timestamp. Precompiling the service definition improves Service Manager performance. The timestamp enables the Service Manager application to track any modifications you might make while the definition is being used.
3. **Uninstall**—You can uninstall a service definition file, for example, if you no longer want to use that definition. When you uninstall a service definition file, any existing service sessions that were activated using the original service definition continue to use the original definition until you deactivate the service session.
4. **Modify**—You can update an existing service definition file at any time. To update a service definition file:
 - a. Use your text editor on your computer to make changes to the original service definition file.
 - b. Copy the updated service definition file back to your router's NVS—this overwrites the original file on the router.
 - c. Install the new service definition file.

All new service sessions will be activated using the new service definition. Any existing service sessions that were activated using the original service definition continue to use the original definition until you deactivate the service session.

copy

- Use to copy a service definition macro file from your computer to the router's NVS.
- Specify the directory containing the macro file you want to copy and the name you want to use for the file in NVS.
- Example

```
host1#copy boston:/serviceDefs/triplePlay/tiered.mac tiered.mac
```
- There is no **no** version.

service-management install

- Use to install or uninstall a service definition.
- You must include the .mac extension.
- During installation, Service Manager precompiles the service definition and extracts the definition file's timestamp.
- After you install the service definition, you can use the definition to create service sessions for subscribers.
- To update an existing service definition, you make changes to the original macro file on your computer, copy the updated file to NVS, and install the updated file. All subsequent service sessions use the new service definition file. However, currently active service sessions continue to use the original definition file until the sessions are deactivated, then reactivated.
- Example 1—Installing
`host1(config)#service-management install tiered.mac`
- Example 2—Uninstalling
`host1(config)#no service-management install tiered.mac`
- Example 3—Updating
 ! update the original macro file on the remote system

 ! copy the updated macro file to the router
`host1#copy boston:/serviceDefs/triplePlay/tiered.mac tiered.mac`
`host1#configure terminal`
 ! install the updated service definition on the router
`host1(config)#service-management install tiered.mac`
- Use the **no** version to uninstall a service definition.

Referencing Policies in Service Definitions

In Profile Configuration mode, policy interface commands for IP and L2TP allow attachments to be merged into any existing merge-capable attachment at an attachment point. Merged policies are dynamically created. Service Manager can request that multiple interface profiles be applied or removed at an interface as part of service activation or deactivation. Service Manager also specifies whether or not the attachments created from these interface profiles persist on subsequent reloads.

Service Manager can specify whether a component policy attachment is non-volatile. If the interface where the component policy is attached is volatile, then policy management makes the attachment volatile even when the Service Manager specifies otherwise. A non-volatile interface can have both volatile and non-volatile component policy attachments. The merged policy that is created is the merge of all component policies attached at a given attachment point regardless of their volatility. The merged policy and its attachments are always volatile and reconstructed on each reload operation.

For further details on merging policies, see *JUNOS Policy Management Configuration Guide, Chapter 6, Merging Policies*.

Referencing QoS Configurations in Service Definitions

You can use QoS profiles and QoS parameters to define a service for a subscriber. For example, you can configure the shaping rate for traffic in a video service by using a QoS parameter instance.

To transmit the QoS configuration to the subscriber interface (that is, the forwarding interface at the top of the interface column), you must configure the QoS profiles and QoS parameter instances in static profiles.

Specifying QoS Profiles in a Service Definition

You can configure one QoS profile per subscriber interface. We recommend that you specify the QoS profile in the first set of services applied to the subscriber's interface.

You can modify the QoS profile by modifying configurations referenced by the QoS profile, including QoS parameter instances. You can also attach a new QoS profile when activating a service, but make sure that the QoS profile is attached to the subscriber's interface.

For more information about configuring QoS profiles, see *JUNOS Quality of Service Configuration Guide, Chapter 16, Configuring and Attaching QoS Profiles to an Interface*.

Configuring a QoS Profile for Service Manager

To configure a QoS profile for Service Manager:

1. Configure the profile.

```
host1(config)#profile videoService
```

2. Configure the QoS profile.

```
host1(config-profile)#qos-profile videoBandwidth1
```

3. (Optional) Complete the QoS profile configuration described in *JUNOS Quality of Service Configuration Guide, Chapter 16, Configuring and Attaching QoS Profiles to an Interface*.

profile

- Use to create a profile and enter Profile Configuration mode.
- You specify a profile name with up to 80 alphanumeric characters.
- Example

```
host1(config)#profile iptv
host1(config-profile)#
```

- Use the **no** version to remove a profile.

qos-profile

- Use to add a QoS profile command for use with Service Manager. When the service is activated, the QoS profile is created and attached to the subscriber interface.
- Example


```
host1(config)#profile iptv
host1(config-profile)#qos-profile video
```
- Use the **no** version to remove the QoS profile from the profile.

Specifying QoS Profiles in a Service Definition

After you configure a QoS profile for Service Manager, you can reference it in a service definition. For example:

```
profile <# eastcoast ; '\n' #>
qos-profile <# video; '\n' #>
```

In this example, activating the service definition attaches the video QoS profile to the subscriber interface. Service Manager overwrites the existing QoS profile attachment at the subscriber interface.

Deactivating the service detaches the video QoS profile from the subscriber interface.

Specifying QoS Parameter Instances in a Service Definition

You can specify that Service Manager create QoS parameter instances when the subscriber logs in (during service activation) or through RADIUS QoS parameter VSAs.

You can specify up to eight parameter instance commands within a profile. When you activate a service, Service Manager creates or modifies parameter instances for the subscriber interface that matches one of the subscriber-interface types configured in the QoS parameter definition.

Deactivating a service can modify or remove QoS parameter instances.

Using a service definition, you can also configure QoS parameters instances to add value to an existing parameter instance using the **add** keyword or dynamically create new parameter instances with an initial value using the **initial-value** keyword.

For more information about configuring QoS parameters, see *JUNOS Quality of Service Configuration Guide, Chapter 23, QoS Parameter Overview*.

Creating a Parameter Instance in a Profile

To create a QoS parameter instance for Service Manager:

1. Configure the QoS parameter definition described in *JUNOS Quality of Service Configuration Guide, Chapter 23, QoS Parameter Overview*.

You must configure at least one controlled-interface type and one subscriber-interface type. The range specified in the parameter definition controls the available value of the parameter instance.

2. Configure the QoS profile.

```
host1(config)#profile video
```

3. Configure the QoS parameter instance command in the profile.

```
host1(config-profile)#qos-parameter videoBandwidth1 add 40000
```

qos-parameter

- Use to create a parameter instance command in a profile. When the service is activated, the parameter instances are created for the subscriber interface.
- Use the **add** keyword in Profile Configuration mode to add a value to an existing parameter instance.
- Use the **initial-value** keyword to create a new instance with the specified value.
- Examples


```
host1(config)#profile video
host1(config-profile)#qos-parameter max-subscriber-bandwidth initial-value 15000
```
- In Profile Configuration mode, the **no** version removes the QoS parameter instance command in the profile.

Specifying QoS Parameter Instances in a Service Definition

After you configure a QoS parameter instance for Service Manager, you can reference it in a service definition. For example:

```
<# qosserviceone(bandwidth1, bandwidth2) #>
  profile <# profileName ; '\n' #>
    qos-parameter <# qosParameterName1 ; ' ' ; bandwidth1 ; '\n' #>
    qos-parameter <# qosParameterName2 ; ' ' ; bandwidth2 ; '\n' #>
  <# endtpl #>
```

When you activate a service, Service Manager creates the parameter instance and overwrites previous parameter instances. For example, activating the qosserviceone service definition creates a profile containing two QoS parameter instances. Service Manager creates the qosParameterName1 parameter instance with the value of bandwidth1, and creates qosParameterName2 with a value of bandwidth2.

If you activate the service definition using `qoserviceone(2000000,3000000)`, Service Manager creates `qosParameterName1` with a value of 2000000 and `qosParameterName2` instance with a value of 3000000.

Specifying the Add and Initial-Value Keywords

You can use the **add** keyword to add value to an existing parameter instance. For example:

```
<# qoserviceone(bandwidth1, bandwidth2) #>
  profile <# profileName ; '\n' #>
    qos-parameter <# qosParameterName3 ; ' add ' ; bandwidth2 ; '\n' #>

<# endtmpl #>
```

When you specify parameter instances using the **add** keyword, you can also use the **initial-value** keyword to specify an initial value. For example:

```
<# qoserviceone(bandwidth1, bandwidth2) #>
  profile <# profileName ; '\n' #>
    qos-parameter <# qosParameterName4 ; ' add ' ; bandwidth2 ;
                  ' initial-value 1000000' ; '\n' #>

<# endtmpl #>
```

When you activate the service, Service Manager locates the existing QoS parameter instance in the interface column. If Service Manager does not find a parameter instance, it creates one with a value specified in the **initial-value** keyword (in this case, 100000). The value in the command is then added to the initial value. If an existing parameter instance is found, Service Manager adds the value to the existing interface.

For example, when you activate `qosServiceOne` as `qosServiceOne(2000000,3000000)`, Service Manager attempts to locate the parameter instance `qosParameterName4` for the subscriber's interface. If it finds a parameter instance, it adds `bandwidth2` (3,000,000) to the current value. If Service Manager does not find a parameter instance, it creates one with an initial value of 1,000,000 and adds 3,000,000. The final parameter instance value is 4,000,000.

When deactivating the service, Service Manager locates the QoS parameter instance and subtracts the value in the command from the existing instance value. If the parameter is no longer referenced, the parameter instance is removed.

Modifying QoS Configurations with Service Manager

This section describes how to modify QoS configurations with Service Manager.

Modifying Parameter Instances

Service Manager activates services without considering current parameter instance values. For example, when you deactivate a video service, Service Manager can add 5 Mbps to a parameter associated with the shaping rate of a video queue.

Similarly, Service Manager can deactivate services and restore parameter instances to their previous value. For example, when you deactivate a video service, Service Manager can subtract 5 Mbps from a parameter associated with the shaping rate of a video queue.

Table 109 lists the results of a series of activations and deactivations of parameters using the **add** and **initial-value** keywords.

Table 109: Sample Modifications Using the Add and Initial-Value Keywords

Action	QoS Parameter Instance	Result
Activate	qos-parameter video-bw add 5000000 initial-value 0	Parameter instance video-bw is created with a value of 5000000
Activate	qos-parameter video-bw add 1000000 initial-value 0	Parameter instance video-bw is increased by 1000000, for a total of 6000000
Deactivate	qos-parameter video-bw add 1000000 initial-value 0	Parameter instance video-bw is decreased by 1000000, for a total of 500000
Deactivate	qos-parameter video-bw add 5000000 initial-value 0	Parameter instance video-bw is removed

Removing a parameter instance using profiles is based on the number of times a parameter instance is modified, not the value added.

Modifying parameter instances in profiles and modifying explicit parameter instances can cause invalid parameter instance values. Table 110 lists a series of activations and deactivations using parameter instances in profiles and explicit parameter instances. By the second deactivation, the parameter has a negative value (-4000000).



NOTE: We recommend that you do not configure negative values for Service Manager.

Table 110: Sample Modifications Using Parameter Instances

Action	QoS Parameter Instance	Result
Activate	qos-parameter video-bw add 5000000 initial-value 0	Parameter instance video-bw is created with a value of 5000000
Activate	qos-parameter video-bw add 1000000 initial-value 0	1000000 is added to parameter instance video-bw, for a total of 6000000
Activate	qos-parameter video-bw 2000000	Parameter instance video-bw is set to 2000000
Deactivate	qos-parameter video-bw add 1000000 initial-value 0	1000000 is subtracted from parameter instance video-bw for a total of 1000000
Deactivate	qos-parameter video-bw add 5000000 initial-value 0	5000000 is subtracted from parameter instance video-bw for a total of -4000000
Deactivate	qos-parameter video-bw 2000000	Parameter instance video-bw is removed

Modifying QoS Configurations in a Single Service Manager Event

QoS accepts QoS profile attachments and parameter instances created using multiple sources (profiles, RADIUS, or Service Manager) within a single Service Manager event. Events include:

- Subscriber login
- Subscriber logout
- RADIUS Change of Authority (CoA)

QoS prioritizes the creation of QoS profiles and parameter instances within events. Table 111 lists the sources that overwrite QoS profiles and parameter instances created by other sources. Each row represents new QoS profiles and parameter instances; columns represent existing QoS profiles and parameter instances.

Table 111: Configuration Within a Single Service Manager Event

	Profile	RADIUS	Service Manager
Profile	a	–	–
RADIUS	a	a	–
Service Manager	a	a	a

Modifying QoS Configurations Using Other Sources

You can modify QoS configurations with Service Manager by using other QoS sources. For example, you can modify a parameter instance that was created with Service Manager by using the CLI. Similarly, you can use SNMP to detach a QoS profile attached by Service Manager.

Table 112 lists the sources that you can use to modify QoS profile attachments and parameter instances.

Table 112: Modifying QoS Configurations with Other Sources

	QoS Profile Attachment	QoS Parameter Instances
Service Manager	a	a
RADIUS	a	a
SNMP	a	–
SRC software	a	–
CLI	a	a

The following sections describe the precedence of each source when modifying configurations.

Service Manager

QoS profile attachments and parameter instances created through Service Manager have precedence over all other sources. For example, Service Manager can overwrite a QoS profile attachment modified through RADIUS, SNMP, the SRC software, or the CLI.

Conversely, Service Manager configurations can be overwritten through SNMP, the SRC software, and the CLI, but not by RADIUS.

Service Manager counts references of parameter instances. You can modify parameter instances created by Service Manager using other sources without affecting the reference counts. For more information, see *QoS Statistics* on page 550.

RADIUS

QoS profile attachments and parameter instances configured through RADIUS can overwrite QoS profile attachments and parameter instances configured through the SNMP, the SRC software, and the CLI, but not those created by Service Manager.

Conversely, QoS profiles and parameter instances configured through RADIUS can be overwritten by any source (SNMP, the SRC software, CLI, and Service Manager).

SNMP, the SRC Software, and the CLI

QoS profile attachments and parameter instances configured through the CLI can overwrite QoS profile attachments and parameter instances configured through any source.

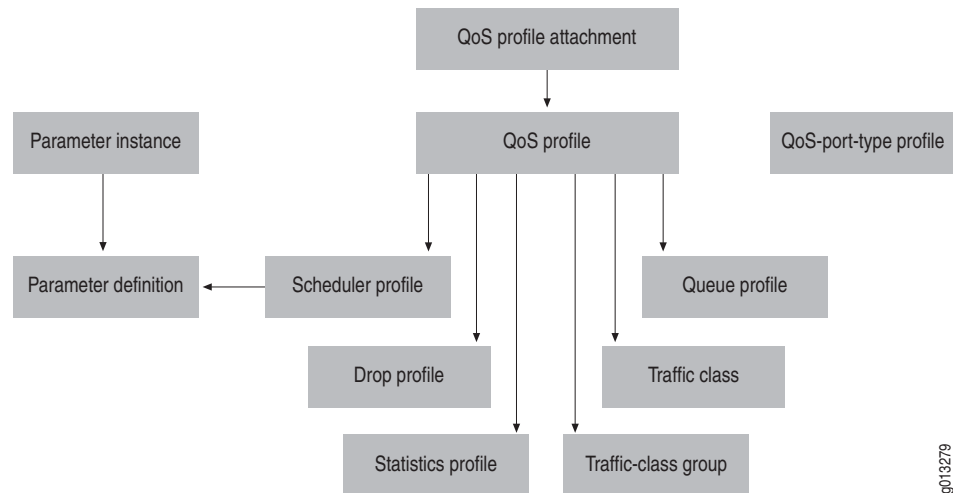
QoS profiles attached through SNMP and the SRC software can also overwrite QoS profile attachments configured through any source.

Conversely, QoS profiles and parameter instances configured through the CLI, SNMP, or the SRC software can be overwritten by any source.

Removing QoS Configurations Referenced by Service Manager

When Service Manager no longer references a QoS configuration, it must be removed from the service definition.

Figure 28 on page 549 shows the references for QoS configurations.

Figure 28: QoS Configuration Dependency Chain

Service Manager automatically removes QoS profiles and parameter instances. After removing the QoS profile and parameter instances, Service Manager automatically removes the following QoS configurations in the following order:

1. QoS profiles
2. Scheduler profiles
3. Queue profiles
4. Drop profiles
5. Statistics profiles

Service Manager does not automatically remove the following QoS configurations:

- Parameter definitions
- Traffic classes
- Traffic-class groups
- QoS-port-type profiles

QoS for Service Manager Considerations

When you specify QoS configurations in Service Manager, the following considerations apply.

RADIUS or Service Manager

We recommend that you choose either RADIUS or Service Manager to create a single parameter instance. If you use both RADIUS and Service Manager, parameter instances activated using Service Manager take precedence.

Interoperability with Other Service Components

Service Manager removes QoS profiles and parameter instances if other components in the service definition (for example, policies) cause an error.

QoS Statistics

Service Manager counts references of parameter instances in profiles. The reference count is incremented each time the parameter is configured through the CLI, RADIUS, or Service Manager. The reference count is decremented each time the parameter is unconfigured, such as through service deactivation. Modifications to parameter instances are also reference counted, using a separate reference count. Parameter instances are removed when both reference counts reach zero.

Service Manager also counts references of modified parameters in profiles using the **add** keyword. The reference count is incremented each time the parameter is modified through service activation with the **add** keyword. The reference count is decremented each time the parameter is modified through service deactivation. References of regular parameter instances are also counted, using a separate reference count. Parameter instances are removed when both reference counts reach zero.

Ranges

You can verify ranges for parameter instances by specifying a range in the parameter definition using the **range** command.

When activating the service or modifying parameters, Service Manager verifies the value of the parameter instance to be within the specified range and generates an informational log message indicating the value is outside the range. Service Manager does not verify ranges when you specify the parameter instances within profiles at the time of configuration.

Configuring the Service Manager License

Use the Service Manager license to enable full Service Manager application support. You can create a maximum of 10 subscriber sessions when the Service Manager license is not enabled. If you disable the Service Manager license and more than 10 subscriber sessions exist, you cannot enable any new sessions—however, all existing active subscriber sessions continue to function.

For information about the maximum number of subscriber sessions supported, see *JUNOS Release Notes, Appendix A, System Maximums*.

license service-management

- Use to specify the Service Manager license and enable full Service Manager application support—if the license is not enabled, you are limited to 10 subscriber sessions.
- The license is a unique string of up to 15 alphanumeric characters.



NOTE: Obtain the license from Juniper Networks Customer Service or your Juniper Networks sales representative.

- Example
host1(config)#**license service-management 123456789**
- Use the **no** version to disable the license.

Managing and Activating Service Sessions

You can use either RADIUS or the CLI to manage, activate, and deactivate service sessions. The following list describes some of the differences between using RADIUS and the CLI to manage the Service Manager application.

- RADIUS-based login and RADIUS CoA support:
 - Provides dynamic activation and deactivation based on subscriber service selection
 - Provides greater flexibility and efficient management for a large number of subscribers and services
 - Enables you to use mutual exclusion (mutex) groups to create mutex services (RADIUS CoA only)
- CLI-based support:
 - Provides static activation and deactivation for subscribers who are always logged in
 - Is useful for testing new service definitions
 - Enables you to preprovision services that you can activate later

Using RADIUS to Manage Subscriber Service Sessions

Service Manager supports two RADIUS-based methods for dynamically activating subscriber service sessions. Dynamic service sessions that RADIUS activates are not stored in NVS. Both methods can also apply optional statistics and session threshold (volume and time) configurations. The two methods differ in how Service Manager activates a subscriber service session:

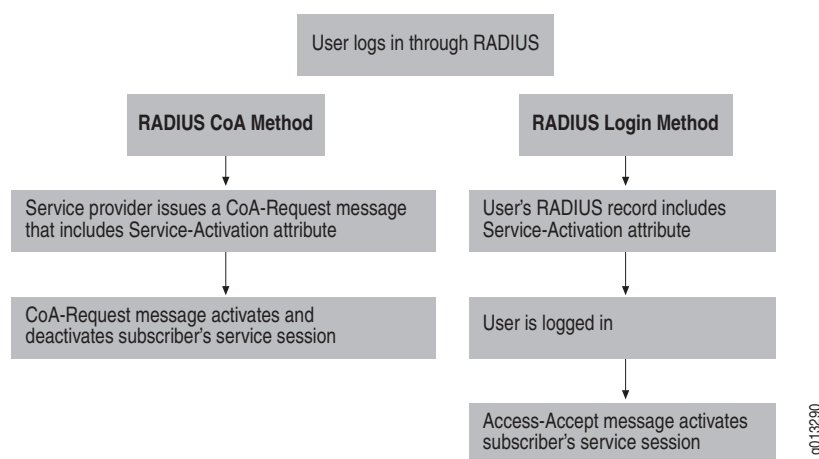
- RADIUS login method—The service session is activated when the subscriber logs in. At login, RADIUS verifies that the Activate-Service attribute is configured in the subscriber's RADIUS record. RADIUS then uses vendor-specific attributes (VSAs) in the Access-Accept packet to activate the service session for the subscriber. This method is useful when your subscribers are not currently logged in.
- RADIUS CoA method—Supports dynamic service selection for subscribers. For example, the subscriber might have logged in without a service, or might have used the RADIUS login method to activate a service at login. If no service was activated at login (because of no Activate-Service attribute in the user's RADIUS record), you can later use the CoA method and a separate RADIUS record to create a subscriber session and activate a service session for the subscriber. Or, if the RADIUS login method was used and the subscriber already has an active

service session, you can use the CoA method and a new RADIUS record to activate a new service session for the subscriber (and optionally deactivate the existing service session). The RADIUS CoA method is useful when you have a large number of users already logged in through RADIUS and you want to activate new services for them. This method is also used for the guided entrance service described in *Guided Entrance Service Definition Example* on page 579.

The RADIUS CoA method also supports the use of mutex groups to create mutex services. See *Using Mutex Groups to Activate and Deactivate Subscriber Services* on page 557.

Figure 29 compares the two RADIUS-based methods.

Figure 29: Comparing RADIUS Login and RADIUS CoA Methods



Using RADIUS to Activate Subscriber Service Sessions

To use RADIUS to activate subscriber service sessions, you create a RADIUS record that includes the Activate-Service VSA. For the RADIUS login method, this RADIUS record is used by the Access-Accept message to start Service Manager and activate the service when the subscriber logs in.

For the RADIUS CoA method, the service provider uses a CoA-Request message to activate and deactivate the service for the subscriber who is already logged in.

To configure a service session that will be activated by RADIUS:

1. Create the RADIUS record for the subscriber and service:
 - For RADIUS login—Create the RADIUS record for the subscriber and include the Activate-Service VSA in the record. Specify values for the parameters defined in the service template name of the definition macro file.

- For RADIUS CoA—Format the CoA message to create the RADIUS record for the subscriber. Include the Activate-Service VSA in the record. Optionally, include the Deactivate-Service VSA if the subscriber has an active service session that you want to deactivate. Specify values for the parameters defined in the service template name of the definition macro file.



NOTE: You specify the parameter values in the order in which the parameters appear in the template name of the service definition file. For example, in the tiered service that is defined in Figure 27 on page 539, the template name is:

```
<# tiered(inputBW, outputBW) #>
```

For the RADIUS Activate-Service VSA, you specify values for the input and output bandwidth:

```
tiered(1280000, 5120000)
```

2. Specify optional VSAs for the service session as needed:

- Service-Volume
- Service-Timeout
- Service-Statistics

Service Manager RADIUS Attributes

For the RADIUS login method, the RADIUS VSAs for service activation, threshold configuration, statistics configuration, and interim accounting in Access-Accept messages at subscriber login are used by Service Manager to activate the appropriate service session. For the RADIUS CoA method, Service Manager uses the VSAs for service activation and deactivation, threshold configuration, statistics configuration, and interim accounting in CoA-Request messages to activate the service session. The accounting-related VSAs are included in RADIUS accounting messages.

Table 113 lists the Service Manager-related attributes and indicates which are tagged VSAs. See *Using Tags with RADIUS Attributes* on page 555 for a discussion about using tagged VSAs to group attributes for a service.

Table 113: Service Manager RADIUS Attributes

Attribute Number	Attribute Name	RADIUS Message Type	VSA Description
[1]	User-Name (used with Virtual-Router, Juniper Networks VSA 26-1)	Access-Accept	Uniquely identifies the subscriber session
[8]	Framed-IP-Address (used with Virtual-Router, Juniper Networks VSA 26-1)	Access-Accept	Uniquely identifies the subscriber session

Table 113: Service Manager RADIUS Attributes (continued)

Attribute Number	Attribute Name	RADIUS Message Type	VSA Description
[26-65]	Activate-Service	Access-Accept and CoA-Request	Name of the service to be activated; includes parameter values; a tagged VSA
[26-66]	Deactivate-Service	Access-Accept and CoA-Request	Name of the service to be deactivated Note: This VSA is only used by CoA.
[26-67]	Service-Volume	Access-Accept and CoA-Request	Number of MB of traffic that the service can consume; the service is terminated when output byte count exceeds this value; a tagged VSA
[26-68]	Service-Timeout	Access-Accept and CoA-Request	Number of seconds that the service is to remain active; the service is terminated when the time expires; a tagged VSA
[26-69]	Service-Statistics	Access-Accept and CoA-Request	Statistics configuration; a tagged VSA: 0 = disable 1 = timestamp only 2 = timestamp and volume
[26-83]	Service-Session	For service sessions only: Acct-Start Acct-Stop Interim-Acct	Name of the service (including parameter values) with which the statistics are associated
[26-140]	Service-Interim-Acct-Interval	Access-Accept and CoA-Request	Number of seconds between accounting updates for a service; a tagged VSA
[31]	Calling-Station-ID	Access-Accept	Uniquely identifies the subscriber session
[44]	Acct-Session-ID	Acct-Start Acct-Stop Interim-Acct	Accounting identifier that makes it easy to match start and stop records in a log file; the format is extended to include a colon-separated value that uniquely identifies the subscriber session



NOTE: Service Manager statistics collection is a two-part procedure. You must configure statistics information in the service definition macro file and also enable statistics collection in the RADIUS record.

The Service-Volume and Service-Timeout VSAs rely on the values captured by the Service Manager statistics feature to determine when a threshold is exceeded. Therefore, you must configure and enable statistics collection to use these attributes. Service-Volume For detailed information about Service Manager statistics see *Configuring Service Manager Statistics* on page 572.

Table 114 describes a partial RADIUS Access-Accept packet that activates a service session for subscriber `client1@isp1.com`. (Figure 27 on page 539 shows the service definition macro file that creates the tiered service.) The session enables the subscriber to use the tiered service with an input bandwidth of 1280000 and output bandwidth of 5120000. The subscriber can use the service for 5 hours (18000 seconds), and Service Manager captures both timestamp and volume statistics during the session (service-statistics value of 2). Also, accounting for the service is updated every 600 seconds (10 minutes).

Table 114: Sample RADIUS Access-Accept Packet

RADIUS Attribute	Tag	Value
username	none	client1@isp1.com
class	none	(binary data)
service-activation	6	tiered(1280000, 5120000)
service-timeout	6	18000
service-statistics	6	2
service-interim-acct-interval	6	600

Using Tags with RADIUS Attributes

Service Manager uses tagged RADIUS VSAs to enable a single RADIUS record to activate multiple service sessions for a subscriber, with each session having unique attributes. A particular tag identifies a specific Activate-Service attribute and all other RADIUS attributes that are associated with that Activate-Service attribute.

You can specify a maximum of 8 tags (1–8), which enables you to activate up to eight unique service sessions for a subscriber in a single RADIUS record. The following are tagged VSAs—they must always have a tag in their RADIUS entry:

- Activate-Service
- Service-Statistics
- Service-Timeout
- Service-Volume
- Service-Interim-Acct-Interval

Table 115 describes an Access-Accept packet that activates the two services, tiered and voice, for subscriber `client1@isp1.com`. Each service has its own unique tag, enabling you to assign attributes for one service, but not the other. For example, the two services have different timeout settings and different interim accounting intervals, and statistics are enabled only for the tiered service.

Table 115: Using Tags

RADIUS Attribute	Tag	Value
username	none	client1@isp1.com
class	none	(binary data)
service-activation	2	tiered(1280000, 5120000)
service-timeout	2	18000
service-statistics	2	1
service-interim-acct-interval	2	600
service-activation	6	voice(100000)
service-timeout	6	1440
service-interim-acct-interval	6	1200

Using RADIUS to Deactivate Service Sessions

A service session can be deactivated by a CoA-Request message or when a subscriber logs out of a RADIUS-activated service session. If the subscriber logs off the router, Service Manager deactivates that subscriber session and all associated service sessions.

RADIUS also supports attributes that you can use to manage deactivation of service sessions. You can:

- Set time or volume thresholds for the service
- Use the Deactivate-Service RADIUS attribute

Setting Thresholds

You can set a threshold for the session by including one or both of the following attributes in the RADIUS record:



NOTE: The Service-Timeout and Service-Volume attributes use values captured by the Service Manager statistics feature to determine when a threshold is exceeded. Therefore, you must configure and enable statistics collection to use these attributes. See *Configuring Service Manager Statistics* on page 572.

- **Service-Timeout**—The number of seconds that the service session is active. You can specify a number in the range 0–16777251. A value of 0 indicates that the session never times out. A particular Service-Timeout VSA can be used by a maximum of 2000 services.

The service-timeout threshold accuracy is within 30 seconds of the specified value.

- **Service-Volume**—The total number of MB of traffic that can use the service session. You can specify a number in the range 0–16777251 MB. A value of 0 indicates that there is no limit to the amount of traffic for the session. A particular Service-Volume VSA can be used by a maximum of 1000 services.



Service Manager terminates a session when the *output* byte count exceeds the configured service-volume threshold. The output byte count is captured by the **output-stat-clacl** string in the classifier list variable that you configure to collect statistics. See *Configuring Service Manager Statistics* on page 572.

The service-volume threshold accuracy is based on a 10-second period. Service Manager does not immediately deactivate a service session when the output byte count reaches the service-volume threshold. Instead, Service Manager checks the volume in 10-second intervals and deactivates a service session at the end of the 10-second period in which the output byte count reaches the volume threshold. For example, if a threshold is reached 4 seconds into the 10-second interval, the session continues for the remaining 6 seconds in the measuring period and is then terminated. Therefore, the total volume equals the threshold plus the volume during the additional 6 seconds.

When the output byte count reaches the threshold, RADIUS deactivates the service session. You must use tags to associate threshold attributes with the Activate-Service attribute for the service session.

Using the Deactivate-Service Attribute

You can also include the Deactivate-Service attribute in the subscriber's RADIUS record. The format for this attribute is the same as the format of the Activate-Service attribute—the name of the service, including parameters. The Deactivate-Service attribute is used by RADIUS CoA messages, such as in a guided entrance service. See *Guided Entrance Service Example* on page 578 for more information.

Using Mutex Groups to Activate and Deactivate Subscriber Services

Service Manager supports two methods that use RADIUS CoA-Request messages to activate and deactivate subscriber services and that can also dynamically change a service that is currently provided to a subscriber.

In the first method, you use a CoA message with the Activate-Service VSA to activate the new service; you can optionally include the Deactivate-Service VSA to deactivate the subscriber's existing service. This method is described in *Using RADIUS to Activate Subscriber Service Sessions* on page 552.

The second method uses mutual exclusion (mutex) groups to create mutex services. With this method, you group services together in a mutex group. When you use a CoA message to activate a service that is in a mutex group, Service Manager activates the new service and implicitly deactivates any existing service that it is a member of the same mutex group as the newly activated service. Service Manager does not deactivate an existing service that is a member of a different mutex group or is not a member of a mutex group.

Using mutex services results in a more reliable activation and deactivation process than the original CoA-Request method. With mutex services, Service Manager always activates the new service before deactivating the existing service. This ensures that the subscriber is never without an active service. In the original CoA-Request method, the order of activation and deactivation is random—in some cases the existing service might be deactivated before the new service is activated, or the new activation might fail. In these cases, the subscriber might be without an active service.

If statistics are enabled when you activate a mutex service, Service Manager sends a RADIUS Acct-Stop message for the deactivated service.

Activating and Deactivating Multiple Services

The Service Manager mutex service feature enables you to activate and deactivate multiple services with a single CoA-Request message. A CoA-Request message can have more than one service activation request—the multiple service requests might be from the same mutex group or from different groups. The following examples describe how you might use mutex groups to activate and deactivate multiple services.

■ Example 1—Multiple mutex services of the same mutex group

Service Manager activates the multiple mutex services, which are in the same group, then deactivates all previously existing services that are also members of that mutex group. Active services that are members of different mutex groups are unaffected.

■ Example 2—Multiple mutex services of different mutex groups

Service Manager activates the mutex services, which are members of different mutex groups. Service Manager then deactivates all previously existing services that are members of the same mutex groups as any of the newly activated services. Active services that are members of different mutex groups are unaffected.

Configuring a Mutex Service

To configure and enable a mutex service, you complete the following steps:

1. Create the new service definition and configure the service as a member of a mutex group.

When you create the service definition, include the following service attribute in the service definition, where *groupIndex* identifies the mutex group for this service.

```
<# env.setResult("mutex-group", "groupIndex") #>
```

For example (the mutex group attribute is highlighted in bold text):

```
!parameterizes input and output bandwidth
<# tiered(inputBW, outputBW) #>

<# uid := app.servicemanager.getUniqueId #>
<# name := "SM-tiered-" $ uid #>
```

```

<# oname := "SM-0-tiered-" $ uid #>

classifier-list matchAll ip any any
rate-limit-profile <# name #> one-rate
    committed-rate <# inputBW; '\n' #>

policy-list <# name; '\n' #>
    classifier-group matchAll precedence 10000
    rate-limit-profile <# name; '\n' #>
    traffic-class best-effort

policy-list <# oname; '\n' #>
    classifier-group matchAll precedence 10000
    traffic-class best-effort

profile <# name; '\n' #>
    ip policy secondary-input <# name #> statistics enabled merge
    ip policy output <# oname #> statistics enabled merge
    qos-profile triplePlayIP
    qos-parameter maxSubscBW <# outputBW; '\n' #>

<# env.setResult("mutex-group", "12") #>
<# env.setResult("activate-profile", name) #>
<# env.setResult("secondary-input-stat-clac1", "matchAll") #>
<# env.setResult("output-stat-clac1", "matchAll") #>

<# endtmpl #>

```

2. Activate the mutex service

Use a RADIUS CoA-Request message and the new service definition to create the mutex service. The new service is considered a mutex service because it belongs to a mutex group.

Service Manager activates the new service and deactivates any existing active service that is a member of the same mutex group as the new service.

3. (Optional) Verify the status of the new service.

```

host1#show service-management subscriber-session client1@isp.com interface ip
192.168.0.1
User Name: CLIENT1@ISP.COM, Interface: ip 192.168.0.1
Id: 1
Owner: AAA 4194326
Non-volatile: False
State: Active
ServiceSessions:

```

Name	mutex	Owner/Id	State	Operation
tiered(2000000,3000000)	12	AAA 4194326	ConfigApplySuccess	Activate
Name	Non-volatile			
tiered(2000000,3000000)	False			

Configuring RADIUS Accounting for Service Manager

The Service Manager application supports RADIUS accounting and interim accounting for subscriber service sessions that are activated by the RADIUS login and RADIUS CoA methods. When RADIUS accounting is enabled, RADIUS generates:

- An Acct-Start message when a service session is activated
- An Acct-Stop message when a service session is deactivated
- Interim-Acct messages

RADIUS accounting messages always include Service Manager time statistics. You must enable Service Manager volume statistics for a service session.

When you terminate a subscriber session, Service Manager first sends RADIUS Acct-Stop messages for any active services associated with the subscriber session, and then sends the Acct-Stop message for the subscriber session.



NOTE: Service Manager statistics collection is a two-part procedure. You must configure statistics information in the service definition macro file and also enable statistics collection. For detailed information about Service Manager statistics, see *Configuring Service Manager Statistics* on page 572.

To support RADIUS accounting for Service Manager, the RADIUS Acct-Session-ID attribute [44] has been extended to include a colon-separated identifier, which uniquely identifies a service for a subscriber. For example:

```
erx FastEthernet 12/0:0001048580:002478
```

The Service-Session attribute (VSA 26-83) identifies the name of the service. This attribute is the value of the Activate-Service or Deactivate-Service attribute (including parameter values) that was used in the RADIUS Access-Accept message to activate or deactivate the service session. For example:

```
tiered(1280000, 5120000)
```

Table 116 lists the RADIUS accounting attributes used by the Service Manager application.

Table 116: Service Manager RADIUS Accounting Attributes

Attribute Number	Attribute Name	RADIUS Message Type	VSA Description
[26-83]	Service-Session	For service sessions only: Acct-Start Acct-Stop Interim-Acct	Name of the service (including parameter values) with which the statistics are associated
[26-140]	Service-Interim-Acct-Interval	Access-Accept and CoA-Request	Number of seconds between accounting updates for a service; a tagged VSA
[44]	Acct-Session-ID	Acct-Start Acct-Stop Interim-Acct	Accounting identifier that makes it easy to match start and stop records in a log file; the format is extended to include a colon-separated value that uniquely identifies the subscriber session

Configuring Service Interim Accounting

Interim accounting determines how often accounting information is updated and sent to an accounting server. In addition to the user-based interim accounting supported on the router, Service Manager supports service-related interim accounting—you can configure an interim accounting interval for services that are created during a user RADIUS-based login and services that are activated by a CoA operation.

The service interim accounting interval is specified by the RADIUS Service-Interim-Acct-Interval attribute (VSA 26-140) that is included in the RADIUS Access-Accept message or CoA-Request message that activates a service session. Because the Service-Interim-Acct-Interval attribute is a tagged attribute, you can configure different interim accounting intervals for a particular user's various services.

You can use the **aaa service accounting interval** command to specify the default service interim accounting interval. Service Manager uses this interval value for service accounting when the Service-Interim-Acct-Interval attribute is not configured.



NOTE: You can also configure interim accounting for users. A user interim accounting interval is configured in the Acct-Interim-Interval RADIUS attribute (RADIUS attribute 85). You use the **aaa user accounting interval** command to specify the default user interim accounting interval, which is used when RADIUS attribute 85 is not configured. See *Chapter 1, Configuring Remote Access* for information about configuring user interim accounting.

When the Service-Interim-Acct-Interval attribute is configured for a service, Service Manager uses the guidelines shown in Table 117 to determine the correct interim accounting interval to use for the service.

Table 117: Determining the Service Interim Accounting Interval

Service-Interim-Acct-Interval Value	Service Manager Action
0	Disables interim accounting for the service
1–599	Uses 600
600–86400	Uses the specified value
86401 or greater	Uses 86400
The tag for the service-interim-acct-interval attribute does not match the tag for any service-activate attribute (VSA 26-65)	Discards the service-interim-acct-interval attribute

Table 118 describes a sample Acct-Start message for a service session. In the table, the two fields used by Service Manager are highlighted. An Acct-Start message for a subscriber session without any active services does not include the Service-Session attribute.

Table 118: Sample Acct-Start Message for a Service Session

RADIUS Attribute	Sample Value
acct-status-type	1
username	client1@isp1.com
event-timestamp	1112191723
acct-delay-time	0
nas-identifier	ERX-01-00-06
acct-session-id	erx FastEthernet 12/0:0001048580:002478
nas-ip-address	10.6.128.45
class	(binary data)
framed-protocol	0
framed-compression	0
framed-ip-address	100.20.0.1
framed-ip-netmask	0.0.0.0
ingress-policy-name (vsa)	forwardAll
egress-policy-name (vsa)	forwardAll
calling-station-id	#ERX-01-00-06#E12#0
acct-input-gigawords	0
acct-input-octets	4032
acct-output-gigawords	0
acct-output-octets	2163
acct-input-gigapackets (vsa)	0

Table 118: Sample Acct-Start Message for a Service Session (continued)

RADIUS Attribute	Sample Value
acct-input-packets	7
acct-output-gigapackets (vsa)	0
acct-output-packets	7
nas-port-type	15
nas-port	3221225472
nas-port-id	FastEthernet 12/0
acct-authentic	1
acct-session-time	0
acct-service-session	tiered(1280000, 5120000)
service-interim-acct-interval	1200
1	

aaa service accounting interval

- Use to specify the default interval between service accounting updates. Service manager uses the default interval when no value is specified in the Service-Interim-Acct-Interval attribute (Juniper VSA 26-140).
- This command and the **aaa user accounting interval** command replace the **aaa accounting interval** command, which is deprecated and might be removed in a future release.
- The default interval is applied on a virtual router basis—this setting is used for services associated with all users who attach to the corresponding virtual router.
- Specify the service accounting interval, in the range 10–1440 minutes. The default setting is 0, which disables the feature.



NOTE: To enable interim service accounting, the service accounting interval must be set to a non-zero value and the service statistics type must *not* be set to *none*.

- Example
host1(config)#**aaa service accounting interval 60**
- Use the **no** version to reset the accounting interval to 0, which turns off interim service accounting when no value is specified in the Service-Interim-Acct-Interval attribute (Juniper VSA 26-140).

aaa user accounting interval

- Use to specify the default interval between user accounting updates. The router uses the default interval when no value is specified in the RADIUS Acct-Interim-Interval attribute (RADIUS attribute 85).
- This command and the **aaa service accounting interval** command replace the **aaa accounting interval** command, which is deprecated and might be removed in a future release.

- The default interval is applied on a virtual router basis—this setting is used for all users who attach to the corresponding virtual router.
- Specify the user accounting interval, in the range 10–1440 minutes. The default setting is 0, which disables the feature.
- Example

```
host1(config)#aaa user accounting interval 20
```
- Use the **no** version to reset the accounting interval to 0, which turns off interim user accounting when no value is specified in the RADIUS Acct-Interim-Interval attribute.

Using the CLI to Manage Subscriber Service Sessions

The CLI-based Service Manager creates static subscriber sessions and service sessions. You can also use CLI commands to immediately deactivate subscriber service sessions. The CLI-based support is particularly useful for:

- Testing your service definitions—for example, you might use the CLI commands to verify that a newly created service definition is correct. When you are satisfied with the service definition, you can then use RADIUS to activate the service for your subscribers.
- Preprovisioning Service Manager services—preprovisioning improves performance and efficiency by freeing Service Manager from having to repeatedly create and remove a service that you activate and deactivate for multiple subscribers. See *Preprovisioning Services* on page 567 for more information about service preprovisioning.

Using the CLI to Activate Subscriber Service Sessions

A subscriber session represents a specific subscriber—the session consists of the subscriber's name, the interface used for the session, and any active services for the subscriber. A subscriber can have one subscriber session active at any given time.

You create a subscriber's service session when you assign a service definition to a subscriber session. Like an AAA-created service, a single subscriber session can have multiple simultaneous service sessions. You can use one method to create the subscriber session, and then a different method to activate the subscriber's service session. For example, you might use RADIUS to create the AAA subscriber session, then use the CLI to activate the service session for the subscriber. You can optionally specify a service session profile that you want to attach to the service session.

You can use the CLI to activate a service session based on subscriber information or owner information:

- Subscriber name and interface method—Activates the service session based on the subscriber name and the interface that the subscriber is using for this subscriber session.

```
host1(config)#service-management subscriber-session client1@isp1.com  
interface atm 4/0.1 service-session "tiered(1280000, 5120000)"
```

- Owner name and ID method—Activates the service session based on the owner that created the subscriber session and the ID that was generated by the owner. For example, if RADIUS is used to create the subscriber session, the owner name is AAA and the owner ID is the Acct-Session-ID that was generated by RADIUS during subscriber creation.

```
host1(config)#service-management owner-session AAA 537446 service-session  
"tiered(1280000, 5120000)"
```



NOTE: You must specify the parameter values in the order in which the parameters appear in the template name of the service definition file. Enclose the service definition name in double quotation marks, with the service's parameter values in parentheses. For example, for the tiered service that is defined in Figure 27 on page 539, the template name is:

```
<# tiered(inputBW, outputBW) #>
```

Use the following format with the **service-session** keyword:

```
"tiered(1280000, 5120000)"
```

service-management owner-session

- Use to activate a service for an existing subscriber by identifying the owner used to create the subscriber session and specifying the service session to use.
- The subscriber session must exist before you use this command.
- Use this command in Privileged Exec mode to create a dynamic subscriber session—dynamic sessions are deleted after a router reboot.
- Use this command in Global Configuration mode to create persistent subscriber sessions that are retained across reboots.
- Specify the name of the owner (the method originally used to create the subscriber session), and the ID generated by the of the owner. For example, if RADIUS was used to create the subscriber session, the owner name is AAA and the owner ID is the Acct-Session-ID generated by RADIUS when the subscriber session was created.
- Include the optional **service-session-profile** keyword to assign a profile to the service session. The service session profile includes additional attributes, such as the type of statistics to be captured for the service session.

- You can activate one subscriber session for a subscriber—and multiple service sessions for a particular subscriber session. If you create a second subscriber session for the same subscriber, only the newest subscriber session, with its services, is used.
- Example 1—Activate a service session for an existing subscriber
host1(config)#**service-management owner-session aaa 573498 service-session “video(4500000, 192.168.10.3)”**

- Example 2—Activate multiple service sessions for an existing subscriber

```
host1(config)#service-management owner-session aaa 573498 service-session
"video(4500000, 192.168.10.3)"
host1(config)#service-management owner-session aaa 573498 service-session
"tiered(1000000, 2000000)"
host1(config)#service-management owner-session aaa 573498 service-session
"voice(1000000, 10.10.10.1)"
```

- Example 3—Include a service session profile when you activate a subscriber's service session

```
host1(config)#service-management owner-session aaa 426777 service-session
"video(4500000, 192.168.10.3)" service-session-profile vodISP1
```

- Use the **no** version to deactivate service sessions based on owner information. See *Using the CLI to Deactivate Subscriber Service Sessions* on page 570 for more information about deactivating subscriber service sessions.

service-management subscriber-session service-session

- Use to activate a service for a subscriber by creating a subscriber session and a service session.



NOTE: Always activate at least one service session for a subscriber session. The ability to create a subscriber session without a service session (by omitting the **service-session** keyword) is not currently supported.

- Use this command in Privileged Exec mode to create a dynamic subscriber session—dynamic sessions are deleted after a router reboot.
- Use this command in Global Configuration mode to create persistent subscriber sessions that are retained across reboots.
- Include the optional **service-session-profile** keyword to assign a profile to the service session. The service session profile includes additional attributes, such as the type of statistics to be captured for the service session.
- You can create one subscriber session for a subscriber—and multiple service sessions for a particular subscriber session. If you create a second subscriber session for the same subscriber, only the newest subscriber session, with its services, is used.
- Example 1—Activate a subscriber session with a single service session

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "video(4500000, 192.168.10.3)"
```

- Example 2—Activate a single subscriber session with multiple service sessions

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "video(4500000, 192.168.10.3)"
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "tiered(1000000, 2000000)"
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "voice(1000000, 10.10.10.1)"
```

- Example 3—Include a service session profile when you activate a subscriber's service session

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "video(4500000, 192.168.10.3)"
service-session-profile vodISP1
```

- Use the **no** version to deactivate service sessions. See *Using the CLI to Deactivate Subscriber Service Sessions* on page 570 for more information about deactivating subscriber service sessions.

Preprovisioning Services

Preprovisioning service sessions is a technique you can use to improve Service Manager's performance. Typically, when you use a service definition to activate a subscriber's service session, Service Manager uses resources to build that service. However, if you later use the same service definition to activate a service session for a second subscriber, Service Manager does not have to rebuild the service—it bases the new service on the service that it built for the first service session. After you deactivate the first session, Service Manager must build a new service for any subsequent subscribers.

Preprovisioning entails activating a service for a dummy user on the null interface. You can then use the preprovisioned service session to activate service sessions for actual subscribers. This technique improves performance because you only require Service Manager to build the service one time, then reuse the original service when you activate future subscriber service sessions.

To preprovision a service you use a command similar to the following example:

```
host1(config)#service-management subscriber-session dummy interface null
service-session "tiered(1000000, 2000000)"
```

Using Service Session Profiles

Service session profiles provide additional flexibility to the Service Manager application by enabling you to assign one or more supported attributes to a particular activation of a service.

For example, you might assign the same video service to two subscribers, but use different service session profiles to set different time limits for each subscriber's service. One subscriber uses the video service for 5 hours (18000 seconds) while the other subscriber's video service is for 10 hours (36000 seconds). Or, you might enable statistics on a subscriber's voice service and disable statistics on the same subscriber's video service.

You can create multiple service session profiles independent of the service activation process. Then, when you activate a service session, you specify the profile that you want to use with that particular service session—you can apply one profile to a service session.

You can configure the following attributes in service session profiles:

- **statistics**—Enables statistics and specifies the type of statistics you want to capture for the service. See *Configuring Service Manager Statistics* on page 572 for additional information about capturing Service Manager statistics. You can specify the following types of statistics:
 - **time**—The service’s duration
 - **volume-time**—The service’s duration and traffic volume
- **volume**—Specifies that the service is automatically deactivated when the indicated traffic volume is exceeded.
- **time**—Specifies that the service is automatically deactivated when the indicated time period is exceeded.



NOTE: The **volume** and **time** attributes use values captured by the Service Manager statistics feature to determine when the threshold is exceeded. Service Manager collects time statistics by default—you must configure and enable volume statistics collection. See *Configuring Service Manager Statistics* on page 572.

To create or modify a service session profile:

1. Specify the name of the service session profile; doing this enters Service Session Profile Configuration mode.

```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#
```

2. Specify the attributes you want to include in the profile.

```
host1(config-service-session-profile)#statistics volume-time
host1(config-service-session-profile)#time 6000
```

3. (Optional) To modify an existing profile, you can add new attributes or use the **no** version of a command to remove an attribute.

```
host1(config-service-session-profile)#no time
```

service-management service-session-profile

- Use to create a new service session profile or to specify the name of an existing profile you want to modify, and to enter Service Session Profile Configuration mode.
- In Service Session Profile Configuration mode, you specify the attributes used in the service session profile, such as the maximum volume limit for the session and the maximum time the session can be used. You can also specify that Service Manager collect statistics for time, or volume, or both.

- Example

```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#
```

- Use the **no** version to delete the service session profile.

statistics

- Use to enable statistics collection and to specify the type of statistics to collect.
 - Use the **time** keyword to collect statistics about the duration of the service session.
 - Use the **volume-time** keyword to collect statistics about both the volume of traffic and the duration of the service session.

- Example

```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#statistics volume-time
```

- Use the **no** version to disable statistics collection.



NOTE: Service Manager statistics collection is a two-part procedure. You must configure statistics information in the service definition macro file and also enable statistics collection. See *Configuring Service Manager Statistics* on page 572.

time

- Use to specify the maximum amount of time that the service session can be active for the subscriber.
- The router immediately terminates the subscriber's service session when the specified time is exceeded.
- The range is 0–16777251 seconds.
- Example


```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#time 6000
```
- Use the **no** version to delete the time attribute from the service session profile.

volume

- Use to specify the maximum amount of bandwidth that can use the service.
- The router immediately terminates the subscriber's service session when the specified traffic volume is exceeded.



NOTE: The **volume** attribute uses values captured by the Service Manager statistics feature to determine when the threshold is exceeded. Therefore, you must configure and enable statistics collection to use this attribute. See *Configuring Service Manager Statistics* on page 572.

- The range is 0–16777251 MB.

- Example

```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#volume 1000000
```
- Use the **no** version to delete the volume attribute from the service session profile.

Using the CLI to Deactivate Subscriber Service Sessions

The CLI supports several methods that enable you to manually deactivate service sessions. You can:

- Gracefully terminate all services or a specific service for a particular subscriber
- Gracefully terminate all service or a specific service associated with a particular owner
- Force the immediate termination of all of a subscriber's sessions
- Use service session profiles to create time or volume thresholds for the service and deactivate the service when the threshold is reached. See *Using Service Session Profiles* on page 567.



NOTE: You can use the CLI commands described in this section to delete subscriber and service sessions that are created by either RADIUS or the CLI.

The Service Manager CLI commands enable you to use variations of the **no service-management subscriber-session** command to terminate service sessions.

Gracefully Deactivating Subscriber Service Sessions

Use the following commands to gracefully deactivate subscriber's services—you can deactivate a specific service for a subscriber, or you can delete a subscriber session, which deactivates all of the subscriber's service sessions. We recommend you use this command to deactivate subscriber service sessions.

no service-management owner-session

- Use to gracefully deactivate service sessions for a subscriber based on owner information.
- Specify the owner name and owner ID of the service session you want to deactivate.
- Use the **no** version with the **service-session** keyword to deactivate the specified service session.
- Use the **no** version *without* the **service-session** keyword to delete the subscriber's session and deactivate all of the subscriber's service sessions.
- Example

```
host1(config)#no service-management owner-session aaa 426777
service-session "video(4500000, 192.168.10.3)"
```
- This is the **no** version of the **service-management owner-session** command.

no service-management subscriber-session service-session

- Use to gracefully deactivate service sessions for a subscriber.
- Use the subscriber's username and interface, not the subscriber session ID, for graceful deactivation.
- Use the **no** version without the **service-session** keyword to delete the subscriber's session and deactivate all of the subscriber's service sessions.
- Use the **no** version with the **service-session** keyword to deactivate the specified service session.
- Example


```
host1(config)#no service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "tiered(1000000, 2000000)"
```
- This is the **no** version of the **service-management subscriber-session** command.

Forcing Immediate Deactivation of Subscriber Service Sessions

Use the following command to force the immediate deactivation of the specified subscriber session—doing this deletes all active service sessions for the subscriber. We recommend this method if you encounter difficulty when you used the graceful deactivation method. Always use the graceful method first.

no service-management subscriber-session force

- Use to force the immediate termination of a subscriber session and to deactivate all services for the specified subscriber session.
- You must specify the subscriber session ID to use the **force** keyword to terminate the subscriber session.



NOTE: To determine the subscriber session ID of a session you want to deactivate, use the **show service-management subscriber-session brief** command. The display lists the IDs of all active subscriber sessions and the owner that created the session, such as AAA (RADIUS) or CLI.

- Example


```
host1(config)#no service-management subscriber-session 8 force
```
- There is no affirmative version of this command; there is only a **no** version.

Using Service Session Profiles to Deactivate Service Sessions

To terminate a subscriber service session when a threshold is reached, you create a service session profile that includes a time threshold, or a volume threshold, or both. Then, you attach the service session profile when you activate the service session. When the specified threshold is reached, the service session is terminated.



NOTE: This feature is not supported by the **service-management owner-session** command. The **service-management owner-session** command only supports service session profiles when activating service sessions.

The following example shows the commands you might use to create a time threshold for deactivating a service session. See *Using Service Session Profiles* on page 567 for information about using the **time** and **volume** keywords in service session profiles.

To create or modify a service session profile:

1. Specify the name of the service session profile and configure the threshold:

```
host1(config)#service-management service-session-profile vodISP1
host1(config-service-session-profile)#time 6000
host1(config-service-session-profile)#exit
```

2. Include the service session profile when you activate the subscriber service session:

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "video(4500000, 192.168.10.3)"
service-session-profile vodISP1
```

Configuring Service Manager Statistics

The Service Manager application provides a flexible and efficient process for identifying and capturing statistics related to subscriber service sessions. Configuring Service Manager to collect statistics is a two-part process. First, you design the service definition macro file to identify the statistics that you want to collect. Second, you configure Service Manager to enable statistics collection when a service session is activated by either RADIUS or the CLI.

The following section describes how to configure the service definition macro file. For information about configuring Service Manager to enable statistics, see *Enabling Statistics Collection with RADIUS* on page 574 if you are using RADIUS to activate services, or see *Enabling Statistics Collection with the CLI* on page 574 if you are using the CLI.

Setting Up the Service Definition File for Statistics Collection

Service Manager statistics are based on classifier lists—the classifier lists are referenced by policy lists that you define in your service definition macro file.

When you configure your service definition for statistics, you include the macro environment command **env.setResult** to indicate the type of statistics to track and to identify the classifier lists to use when generating statistics. The format of the environment command is:

```
<# env.setResult("string", "classifier-list-name") #>
```

The *string* variable specifies the type of statistics to track—Service Manager supports the following strings:

- **input-stat-clacl**—track input statistics
- **output-stat-clacl**—track output statistics
- **secondary-input-stat-clacl**—track input statistics for a policy attached at the secondary input stage

The *classifier-list-name* variable is the name of the classifier list that is associated with the policy list that is defined in the service definition. You can specify multiple classifier lists in the command.

Example 1 This example is a portion of the service definition macro file in Figure 27 on page 539. The two highlighted commands specify the statistics used by the Service Manager application.

```
profile <# name; '\n' #>
  ip policy secondary-input <# name #> statistics enabled merge
  ip policy output <# oname #> statistics enabled merge
  qos-profile triplePlayIP
  qos-parameter maxSubscBW <# outputBW; '\n' #>

<# env.setResult("activate-profile", name) #>
<# env.setResult("secondary-input-stat-clacl", "matchAll") #>
<# env.setResult("output-stat-clacl", "matchAll") #>

<# endtmpl #>
```

The `<# env.setResult("secondary-input-stat-clacl", "matchAll") #>` command specifies that Service Manager track statistics associated with the classifier list named `matchAll`, and that this classifier list is associated with the policy that is attached at the secondary input stage.

The `<# env.setResult("output-stat-clacl", "matchAll") #>` command specifies that Service Manager track the output statistics associated with the `matchAll` classifier list, which is associated with the policy attached at the output stage.

Example 2 This example shows how you can also configure your service definition to collect total statistics from multiple classifier lists. The following command specifies that three classifier lists are used to generate output statistics for a service created by the service definition. Each time statistics are reported for this service, Service Manager uses the total of the statistics for `clacl1`, `clacl2`, and `clacl3`.

```
<# env.setResult("output-stat-clacl", "clacl1 clacl2 clacl3") #>
```

Enabling Statistics Collection with RADIUS

You use the Service-Statistics RADIUS VSA [26-69] with either the RADIUS login or CoA-Request method to enable statistics for RADIUS-activated service sessions. To enable statistics, configure the Service-Statistics VSA with a value of either 1 (timestamp only) or 2 (volume and timestamp).

Table 119 describes a partial RADIUS message in which the Service-Statistics attribute has a value of 2—this enables volume and timestamp statistics for the tiered service assigned to subscriber client1@isp1.com.

Table 119: RADIUS-Enabled Statistics

RADIUS Attribute	Tag	Value
username	none	client1@isp1.com
activate-service	6	tiered(1280000, 5120000)
service-statistics	6	2

When you enable statistics for a RADIUS-activated service, RADIUS accounting reports can use the statistics.

Enabling Statistics Collection with the CLI

You use service session profiles to enable statistics when you activate a service session with the CLI. See *Using Service Session Profiles* on page 567 for detailed information about creating and using service session profiles.

For example, you can use the following procedure to capture statistics that are defined in the service definition macro file for the tiered service:

1. Configure the service session profile to enable statistics. Specify the type of statistics you want to capture (either time or both volume and time).

```
host1(config)#service-management service-session-profile isp1_tiered3
host1(config-service-session-profile)#statistics volume-time
host1(config-service-session-profile)#
```

2. Apply the service session when you activate the subscriber service session.

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "tiered(1000000, 2000000)"
service-session-profile isp1_tiered3
```

The captured statistics are now used when you use the Service Manager **show service-management** commands. For example:

```
host1#show service-management subscriber-session client1@isp1.com interface atm
4/0.1 service-session
User Name: client1@isp1.com, Interface: atm 4/0.1
Service : tiered(1000000,2000000)
Non-volatile : False
Owner : CLI
State : Config ApplySuccess
Activate : True
Statistics Type : time-based and volume-based
```

```

Statistics Complete : False
Poll Interval : 0
Poll Expire : 0
Activate Time : THU MAR 01 21:09:12 2006
Time : 0
Time Expire : 0
Volume MBytes: 2
Volume Expire MBytes: 1
Input Bytes : 594
Output Bytes : 1196
Input Packets : 1
Output Packets : 2

```

Service Manager Performance Considerations

Like any application, Service Manager requires a certain amount of system resources. Consider the following guidelines to maximize the performance of Service Manager when delivering subscriber services:

- Minimize service definitions—Use the minimum number of JUNOS commands in a service definition to specify a service.
- Reference objects in service definitions—Referencing commonly used objects is more resource-efficient than using unique objects for each subscriber (for example, using a subscriber's IP address as a match criteria in a classifier list).
- Preprovision frequently used services— Preprovisioning saves resources by requiring Service Manager to build a popular service only once. You then reuse the original service when you activate future subscriber service sessions. See *Preprovisioning Services* on page 567 for details.
- Capture volume statistics when needed—Repeatedly capturing volume statistics can waste resources.

Service Definition Examples

This section provides examples of service definition macro files. Commented text explains the parameterized values in the examples. Each example is followed by examples of RADIUS information and the CLI command that you might use to activate a subscriber service session.

Tiered Service Example

This example creates a tiered service. A tiered service typically provides set bandwidths for both inbound and outbound traffic for a subscriber. In this example, the bandwidth values are parameterized. Also, this example assumes that QoS profile triplePlayIP and QoS parameter maxSubscBW are configured.

```

!parameterizes input and output bandwidth
<# tiered(inputBW, outputBW) #>

<# uid := app.servicemanager.getUniqueId #>
<# name := "SM-tiered-" $ uid #>
<# oname := "SM-O-tiered-" $ uid #>

```

```

classifier-list matchAll ip any any
rate-limit-profile <# name #> one-rate
    committed-rate <# inputBW; '\n' #>

policy-list <# name; '\n' #>
    classifier-group matchAll precedence 10000
    rate-limit-profile <# name; '\n' #>
    traffic-class best-effort

policy-list <# oname; '\n' #>
    classifier-group matchAll precedence 10000
    traffic-class best-effort

profile <# name; '\n' #>
    ip policy secondary-input <# name #> statistics enabled merge
    ip policy output <# oname #> statistics enabled merge
    qos-profile triplePlayIP
    qos-parameter maxSubscBW <# outputBW; '\n' #>

<# env.setResult("activate-profile", name) #>
<# env.setResult("secondary-input-stat-clac1", "matchAll") #>
<# env.setResult("output-stat-clac1", "matchAll") #>

<# endtmpl #>

```

Sample RADIUS Attributes

RADIUS Attribute	Tag	Value
username	none	client1@isp1.com
activate-service	1	tiered(1280000, 5120000)

Sample CLI Command

```

host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "tiered(1280000, 5120000)"

```

Video-on-Demand Service Definition Example

The following example shows a sample service definition macro file that creates a video-on-demand service—the service provides bandwidth that meets the needs of video streams. The definition creates the bandwidth towards the subscriber and parameterizes the source of the video feed.

The sample CLI command shows an example of the **service-management owner-session** command that you can use to activate the service session.

```

!parameterizes download bandwidth and server address
<# videoMin(downloadBW, serverAddress) #>

<# uid := app.servicemanager.getUniqueId #>
<# name := "SM-video-" $ uid #>

classifier-list <# name #> ip any <# serverAddress #> 0.0.0.0

policy-list <# name; '\n' #>
    classifier-group <# name #> precedence 5000
    traffic-class video

```

```

profile <# name; '\n' #>
  ip policy output <# name #> statistics enabled merge
  qos-parameter maxVideoBW add <# downloadBW; '\n' #>
  exit

<# env.setResult("activate-profile", name) #>
<# env.setResult("output-stat-clacl", name) #>

<# endtmp1 #>

```

Sample Owner ID

Owner	Owner ID	Value
AAA (RADIUS)	Acct-Session-ID (RADIUS attribute 44)	573498

Sample CLI Command

```

host1(config)#service-management owner-session aaa 573498 service-session
"videoMin(4500000, 192.168.23.58)"

```

Voice-over-IP Service Definition Example

This example provides a voice-over-IP service. The service is a session border controller (SBC) media gateway (MG)-based service that has upstream and downstream components.

The IP address and port for both the subscriber and the opposite end of the phone call were originally negotiated with the SBC. The VoIP service learns the IP addresses and ports for both ends of the call, and then specifies that any traffic to either end is put in the voice traffic class.

```

!parameterizes source address and port, destination address and port, and protocol type
<# mgFlow(upDA, upDPort, downDA, downDPort, protType) #>

<# uid := app.servicemanager.getUniqueId #>
<# name := "SM-mgFlow-" $ uid #>
<# oname := "SM-O-mgFlow-" $ uid #>

classifier-list <# name #> <# protType #> any <#upDA #> 0.0.0.0 eq <# upDPort; '\n' #>
policy-list <# name; '\n' #>
  classifier-group <# name #> precedence 2000
    traffic-class voice
    forward

classifier-list <# oname #> <# protType #> any <#downDA #> 0.0.0.0 eq <# downDPort; '\n' #>
policy-list <# oname; '\n' #>
  classifier-group <# oname #> precedence 2000
    traffic-class voice
    forward

profile <# name; '\n' #>
  ip policy input <# name #> statistics enabled merge
  ip policy output <# oname #> statistics enabled merge

<# env.setResult("activate-profile", name) #>

<# endtmp1 #>

```

**Sample RADIUS
Attributes**

RADIUS Attribute	Tag	Value
username	none	client1@isp1.com
activate-service	1	mgFlow(10.10.10.10, 1234, 192.168.45.54, 1234, udp)

Sample CLI Command

```
host1(config)#service-management subscriber-session client1@isp1.com
interface atm 4/0.1 service-session "mgFlow(10.10.10.10, 1234,
192.168.45.54, 1234, udp)"
```

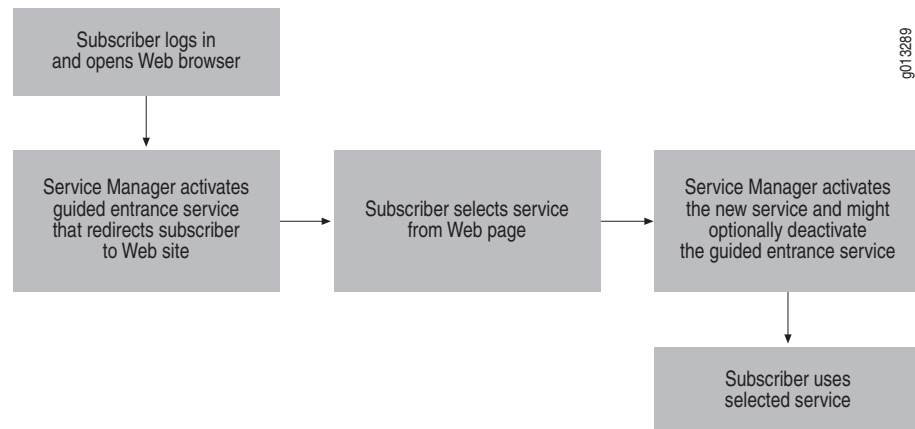
Guided Entrance Service Example

The guided entrance service enables you to create a controlled Internet browsing environment. Guided entrance-based services, which are sometimes called *walled gardens* or *captive portals*, are becoming increasingly important offerings for service providers. When a subscriber logs in and opens a Web browser, the Service Manager guided entrance service transparently directs the subscriber to a specific Web site—at that Web site, the subscriber is presented with a selection of possible services to use. For example, a subscriber might be shown a Web site that offers services such as:

- **Predefined services**—A group of user-selectable services that meets a variety of needs of a single subscriber. The subscriber might select the high-priced highest access speed to perform critical financial transactions but select a lower speed (and lower cost) service for e-mail. For viewing a real-time sports event, the subscriber can select the video-on-demand service. The subscribers have control over the choice and cost of the services they need and use.
- **Prepaid services**—A group of specific services that have been prepaid by the subscriber. For example, a subscriber who has purchased the sports package service is presented with a Web page that lists the currently available sporting events. Or, a subscriber might prepay a VoIP service for a set amount of time.
- **Controlled-service**— An educational service that enables students at a school to access authorized research sites. Or, a limited service for young children that restricts access to safe, closely monitored, age-appropriate Web sites.

Figure 30 shows the sequence of actions that take place during a guided entrance service.

Figure 30: Guided Entrance



Service Manager requires additional configuration considerations for the guided entrance service.

- The `<# redirectUrlName := "http://" $ serverIp $ ":" $ serverPort #>` command in the service definition—Specifies the HTTP local service to which the subscriber is redirected after login. See *Guided Entrance Service Definition Example* on page 579 for a sample guided entrance service definition.



NOTE: You must also configure a policy that redirects packets. See *Creating an Exception Rule within a Policy Classifier Group* in *JUNOS Policy Management Configuration Guide, Chapter 4, Creating Classifier Groups and Policy Rules* for information on creating redirect policies.

- HTTP local server application—Used by the policy in the activated service to direct a subscriber to a specific Web site when the subscriber logs in. See *Configuring the HTTP Local Server to Support Guided Entrance* on page 581 for information about the HTTP local server.
- RADIUS Dynamic Request Server and CoA messages—Enables RADIUS to dynamically activate the new service that the subscriber selects at the Web site. Can also optionally deactivate the original guided entrance service session that is used when the subscriber logs in. See *Chapter 4, Configuring RADIUS Dynamic-Request Server*.

Guided Entrance Service Definition Example

This example shows a guided entrance service. Upon login, the subscriber is redirected to a specific uniform resource locator (URL) at which the subscriber can choose from a list of available services.

```

!parameterizes server address and port
<# http(serverIp, serverPort) #>

```



```

<# serviceTag := "http-" #>
<# uid := app.servicemanager.getUniqueId #>
<# genericName := "SM-X-" $ serviceTag $ uid #>
<# genericInputName := "SM-I-" $ serviceTag $ uid #>
<# genericOutputName := "SM-O-" $ serviceTag $ uid #>
<# clacName := genericName #>

<# profileName := genericName #>
<# inputPolicyName := genericInputName #>
<# inputRateLimitName := genericInputName #>
<# outputPolicyName := genericOutputName #>
<# outputRateLimitName := genericOutputName #>

<# exceptionClacName := "exceptionClacPort" $ serverPort #>
<# serverClacName := "serverClacIp" $ serverIp #>
<# redirectUrlName := "http://" $ serverIp $ ":" $ serverPort #>

configure terminal

classifier-list <# serverClacName #> ip any host <# serverIp; '\n' #>

classifier-list <# exceptionClacName #> tcp any any eq <# serverPort; '\n' #>

ip policy-list <# inputPolicyName; '\n' #>
    classifier-group <# serverClacName; '\n' #>
        forward
    classifier-group <# exceptionClacName; '\n' #>
        exception http-redirect
    classifier-group *
        filter

profile <# profileName #>
    ip http redirectUrl <# redirectUrlName; '\n' #>
    ip policy input <# inputPolicyName #> statistics enabled merge

<# env.setResult("activate-profile", "" $ profileName) #>

<# endtmpl #>

```

Sample RADIUS Attributes

RADIUS Attribute	Tag	Value
username	none	client5@isp1.com
activate-service	1	http(192.168.25.2, 80)

Sample CLI Command

```

host1(config)#service-management subscriber-session client5@isp1.com
interface atm 5/0.1 service-session "http(192.168.25.2, 80)"

```

Using CoA Messages with Guided Entrance Services

Typically, a guided entrance service directs a subscriber to a Web site, where the subscriber can select from a group of available services. When the subscriber selects a new service to use, Service Manager uses a RADIUS CoA message to activate the new service—you can also configure RADIUS to deactivate the original guided entrance service. To inform Service Manager to deactivate the original guided entry service, you must include the Deactivate-Service attribute in the RADIUS records of the services that can be selected from the Web site.

If you configure a guided entrance service, you must also ensure that the router's RADIUS dynamic-request server is enabled and supports CoA messages. See *Chapter 4, Configuring RADIUS Dynamic-Request Server*, for information about the RADIUS dynamic-request server and CoA messages.

Table 120 describes a partial RADIUS Access-Accept message for a guided entrance service and the CoA-Request message for the tiered service that the subscriber subsequently selects from the Web site. The CoA message for the tiered service includes the Deactivate-Service attribute that deactivates the guided entrance service.

Table 120: Deactivating a Guided Entrance Service

Guided Entrance Service Activated at Login

RADIUS Attribute	Tag	Value
username	none	client5@isp1.com
activate-service	1	http(192.168.25.2, 80)

Tiered Service Selected at Web Site

RADIUS Attribute	Tag	Value
username	none	client5@isp1.com
activate-service	2	tiered(1280000, 5120000)
deactivate-service		http(192.168.25.2, 80)
service-timeout	2	720
service-statistics	2	2

Configuring the HTTP Local Server to Support Guided Entrance

JUNOS software supports an embedded Web server, known as the HTTP local server, which is used to support the Service Manager application's guided entrance service. With guided entrance, subscribers are directed to a specific Web site when they log in. At the Web site, the subscribers can then select the service they want to use.

You can configure one HTTP local server per virtual router. The HTTP local server is disabled by default. To configure the HTTP local server:

1. Access the virtual router context.

```
host1(config)#virtual-router west400
host1:west400(config)#
```

2. Create the HTTP local server.

```
host1:west400(config)#ip http
```

3. (Optional) Specify a standard IP access list that defines which subscribers can connect to the HTTP local server.

```
host1:west400(config)#ip http access-class chicagoList
```

4. (Optional) Specify the port on which the HTTP local server receives connection attempts.

```
host1:west400(config)#ip http port 8080
```

5. (Optional) Specify the maximum number of connections that can exist between one IP address and the HTTP local server.

```
host1:west400(config)#ip http same-host-limit 20
```

6. Specify the maximum time that HTTP local servers maintain connections.

```
host1:west400(config)#ip http max-connection-time 1000
```

7. Enable the HTTP local server.

```
host1:west400(config)#ip http server
```

8. Configure the HTTP redirect feature for the profile, interface, or subinterface that will be referenced in the guided entrance service definition.

```
host1:west400(config)#profile guidEnt6
host1:west400(config-profile)#ip http redirectUrl http://ispsite.redirect.com
```

HTTP Local Server Commands

This section describes the commands used to configure the HTTP local server application on the E-series router.

ip http

- Use to create the HTTP local server.

- Example

```
host1(config)#ip http
```

- Use the **no** version to delete the HTTP local server.

ip http access-class

- Use to allow only subscribers on the specified standard IP access list to connect to the HTTP local server.
- Example
host1(config)#**ip http access-class chicagoList**
- Use the **no** version to remove the association between the access list and the HTTP local server.

ip http max-connection-time

- Use to specify the maximum time that the HTTP local server maintains an inactive connection.
- Specify a time in the range 3–7200 seconds, or 0. A value of 0 causes the server to maintain an inactive connection indefinitely.
- Example
host1(config)#**ip http max-connection-time 1000**
- Use the **no** version to restore the default time, 30 seconds.

ip http port

- Use to specify the port on which the HTTP local server receives connection attempts.
- Specify a port number in the range 1–65535.
- Example
host1(config)#**ip http port 8080**
- Use the **no** version to restore the default port number, 80.

ip http redirectUrl

- Use to specify the URL to which a subscriber's HTTP access session is redirected.
- The first access session is typically used by the Service Manager application to provide initial provisioning and service selection for the subscriber.
- HTTP redirect is per-interface; use the command in Profile Configuration mode for dynamic interfaces; use the command in Interface Configuration mode or Subinterface Configuration mode for static interfaces.
- The redirect URL can be a maximum of 64 characters.



NOTE: The HTTP local server must be configured and enabled in the virtual router for the interface on which you use the **ip http redirectUrl** command. Otherwise, the URL redirect operation will fail.

- Example

host1(config-if)#**ip http redirectUrl http://ispsite.redirect.com**

- Use the **no** version to restore the default, which disables the HTTP redirect feature.

ip http same-host-limit

- Use to specify the maximum number of connections that can exist between one IP address and the HTTP local server.
- Specify a number in the range 0–1000.
- Example
host1(config)#**ip http same-host-limit 20**
- Use the **no** version to restore the default number of allowed connections, 3.

ip http server

- Use to enable the HTTP local server.
- Example
host1(config)#**ip http server**
- Use the **no** version to disable the HTTP local server.

