QoS Policy Accounting

The QoS Policy Accounting feature helps you accurately account for traffic on your system. It also provides greater flexibility in assigning quality of service (QoS) configurations to subscribers. In addition, the QoS Accounting High Availability feature ensures that QoS accounting statistics persist, and that the RADIUS accounting billing server continues to report accounting counters during planned and unexpected Route Processor (RP) switchovers. This module describes how to configure QoS policy accounting, use subscriber templates, and activate subscriber accounting accuracy.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for QoS Policy Accounting

- PPP over Ethernet (PPPoE) or PPP over Ethernet over ATM (PPPoEoA) sessions are enabled.
- The RADIUS server is configured.

- Authentication, authorization, and accounting (AAA) is enabled.
- The subscriber's user profile on the RADIUS server has been created.
- A policy map is configured.
- A service template is configured.
- Traffic classes have been created.
- Stateful switchover (SSO) and In-service Software Upgrade (ISSU) prerequisites must be met. For more information, see the *Cisco IOS High Availability Configuration Guide*.

Restrictions for QoS Policy Accounting

- In system failover, the following occurs:
  - For QoS accounting configured statically at the policy map, QoS accounting statistics are reset to zero.
  - For QoS accounting configured dynamically using service templates, sessions no longer exist on the new active Route Processor (RP).

Note In Cisco IOS XE Release 3.5S and later releases, high availability (HA) support is available for accounting services enabled through a service template. Therefore, QoS accounting statistics and service sessions are preserved during a system failover and are available on the new active RP.

- Multicasting is not supported for QoS policy accounting services.
- The following QoS actions are not supported in service templates:
  - account
  - fair-queue
  - netflow-sampler
  - random-detect
- The following QoS filters are not supported in service templates:
  - atm
  - class-map
  - cos
  - destination-address
  - discard-class
  - fr-de
  - fr-dlci

  - input-interface
  - mpls
  - not
  - packet
  - source-address
  - vlan

- Service template definition lines may not exceed maximum configuration line length allowed by the Cisco IOS CLI. You may need to shorten shell variable names to stay within this limit.
- A template service activated on a session cannot be changed. Instead, you can deactivate it and activate a different template service.
- When a template service is active, a legacy complex parameterized string may not be used to change the QoS policy active on a session.
- IP address parameterization is supported only for IPv4 and only for named ACLs without remarks. IP addresses specified in the parameterized service activation are always added to the cloned ACL in this fixed pattern: "permit ip network mask any" and "permit ip any network mask".
- Service templates are supported only for PPP sessions and may not be activated on subinterfaces.
- Only one turbo button service can be active on a session at any given time. Turbo button service is any service that changes a QoS action other than “service-policy xxxx” (changing the child policy) in the class-default of the parent policy.
- Shell variables, QoS class map, and Access Control List (ACL) names may not have the following characters:
  - !
  - $
  - #
  - -
  - ,
  - >
  - <
- Service names are echoed back in the accounting records only for group accounting (when you use $_acctgrp in the service template).
- The IN/OUT QoS policy name active on a session is formed by concatenating the previously active QoS policy (or the static QoS policy specified in the last multiservice Change of Authorization (CoA) or Access-Accept).
- Two template services instantiated from the same service template may not be activated on the session at the same time. However, multiple template services instantiated from unrelated service templates can be active on a session at the same time.
- Template service support is available only for locally terminated PPP and PPP forwarded sessions on the Layer 2 Tunneling Protocol (L2TP) Access Concentrator (LAC).

- For PPP forwarded sessions on the LAC, to apply template services via Access-Accept, use the following configurations:
  - vpdn authen-before-forward.
  - Specify template services only in the user authorization profile (Access-Accept that is received after PPP authentication), not in the authentication profile.
- Only activate template services on the child policy under the parent class-default (only two levels) and on the parent policy (Turbo Button service).
- The default QoS policy can be only two levels deep (Parent + Child under class-default) and should not have a child policy configured under any class other than the class-default.
- A child policy should be configured under the default parent policy class-default in order for template services to be activated at the child level.
- Only rollback due to syntax error checking is supported.
- When multiple service activations or deactivations are included in a single CoA message, the failure of any operation (activation or deactivation) means that the CoA must roll back (undo) all previous operations to restore the session state to what it was before the CoA processing started. In other words, either all the operations must be processed successfully in a CoA or none at all. A CoA negative ACK (NACK) is sent to the RADIUS.
- For rollback to work during Access-Accept processing, subscriber service multiple-accept processing must be configured. The failure to process a service in an Access-Accept should roll back (undo) all previous services in the Access-Accept. The session will come up even if Access-Accept service processing fails.
- Errors originating in the platform or data plane will not trigger rollback which can result in an incomplete service.
- Do not modify a service template if its template services are in use or active on sessions. Use the **show subscriber policy ppm-shim-db** command to display which template services are in use.

Information About QoS Policy Accounting

RADIUS is a networking protocol that provides AAA management. Among other things, each RADIUS accounting message includes ingress and egress counters. The QoS Policy Accounting feature helps you resolve any inaccuracies between counters.

QoS Policy Accounting Feature in Groups

The QoS Policy Accounting feature collects and reports the following information to the RADIUS server per-session:

- Acct-Session-Id
- Ingress and egress packets/bytes/gigawords, packets, and bytes of successfully transmitted packets
- Parent-Session-Id
- Policy name and class or group name (if the QoS Policy Accounting feature is enabled on the group)

- Service name
- Username

When you enable the QoS Policy Accounting feature on a group and assign it a group name, this feature aggregates packets that meet the following criteria:

- Classified by traffic classes in the same group
- Included in the ingress or egress QoS policy applied on the same target

Separate Accounting Streams

If you do not assign a traffic class to a group, but instead assign it to an AAA method list, separate QoS policy accounting streams are created for each traffic class. Separate accounting streams allow you to differentiate between traffic that matches more than one class. Each unique target, direction, policy name, and class name has a unique RADIUS Acct-Session-Id value.

Service Templates

Service templates allow you to dynamically change QoS parameters without defining a new QoS policy on the CLI. You can change QoS policy when a session begins or any time after the session is established. Before you dynamically modify an active QoS deactivate the current service.

To understand service templates, learn the following terms:

- Service templates:
  - Are Cisco IOS shell functions
  - Have IN QoS policy-map definitions
  - Have OUT QoS policy-map definitions
  - Are programmatically invoked
  - Specify default values for shell variables
- Template services:
  - Are QoS service names with a parenthesis in them
  - Have a matching shell-map template definition
  - Are created dynamically during service template shell function execution
- IN Net effect policy map
- OUT Net effect policy map

The QoS Policy Accounting feature, describes how the Cisco IOS shell overrides default values of variables used in service template shell functions. QoS policy definitions inside a shell map may have shell variables in place of QoS action parameter values.

Using Service Templates

To create a service template, you write the service template in a text editor and you then copy the template to the CLI. The contents of a shell map block are treated as text.

When you define the service-template policy maps (policy map $_outgoing/$_incoming), there is no CLI help or prompts available. For example you cannot access the following CLI aids:

- Parser auto completion
- Command options
- Range help
- Syntax checking

Note There is no editor available to you in the CLI, if you make a mistake you must delete the entire service template and then configure it again from the start.

Verifying Service Templates

When you write a service template in a text editor you do not have a syntax checking facility. Therefore, before you activate your service template, you must verify its syntax. The following code sample shows how to verify the *voice-service1* service template. To verify your own template, replace *voice-service1* with your service template name.

```
(shell map voice-service1 police_rate=100000 prec_value=4 queue_size=1)
configure terminal
no policy-map test-svc_IN <-------- Removes previous service template verifications.
no policy-map test-svc_OUT <-------- Removes previous service template verifications.
no aaa-accounting group test_svc_GRP <-------- Removes previous service template
verifications.
end
trigger voice-service1 _incoming=test-svc_IN _outgoing=test-svc_OUT _acctgrp=test-svc_GRP
show policy-map test-svc-IN <--------
Ensure that the output matches the expected service template template service with default
 values.
show policy-map test-svc-OUT <--------
Ensure that the output matches the expected service template template service with default
 values.
```

Removing Service Templates

To remove a service template, at the command line enter:

```
no shell map voice-service1 police_rate=100000 prec_value=4 queue_size=1 in_h=class-default
 out_h=class-default
```

Where voice-service1 is the name of your service template.

Sample Service Templates

Service Template

This example shows a sample service template:

```
{
    configure terminal
    accounting group $_acctgrp list default
    policy-map $_outgoing
      class voip
       police $police_rate 60625 0 conform-action transmit exceed-action drop violate-action
 drop
 exit
        priority level 1
        queue-limit 8 packets
        set precedence $prec_value
        set cos 6
        aaa-accounting group $_acctgrp
      class voip-control
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

        queue-limit $queue_size packets
        set precedence 6
        aaa-accounting group $_acctgrp
   policy-map $_incoming
      class voip
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 5
        aaa-accounting group $_acctgrp
      class voip-control
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
        aaa-accounting group $_acctgrp
}
```

Action Parameter Override

Action Parameter Override is a type of service template where shell variables are used in place of parameters for QoS actions such as police, shape, and bandwidth, configurations entered under a class in a QoS policy.

If you deactivate a template service, the system restores the previously active QoS policy. The QoS policy name may be different but is structurally and functionally identical to the QoS policy active before the template service was activated.

This example generates the service with the following parameters:

```
Reserved variable initialization before executing the service template shell function:
$_incoming = voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN
$_outgoing = voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT
$_acctgrp = aaa-accounting group
voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP list default
```

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
```

After you activate voice-service1(police_rate=200000,prec_value=5,queue_size=32) on the target session, this is the active OUT policy:

```
policy-map
output_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

 class class-default
   shape average 10000000
   service-policy
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
policy-map
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

      class voip
        police 200000 60625 0 conform-action transmit exceed-action drop violate-action
drop
        priority level 1
        queue-limit 8 packets
        set precedence 5
        set cos 6
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP

      class voip-control
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

        queue-limit 32 packets
        set precedence 6
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
class class-default
```

After you activate voice-service1(police_rate=200000,prec_value=5,queue_size=32) on the target session, this is the active IN policy:

```
policy-map
input_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default

 class class-default
   police cir 10000000 bc 312500 conform-action transmit exceed-action drop
   service-policy
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
policy-map
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
      class voip
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 5
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP

      class voip-control
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP

 class-default
```

Action Parameterization Default Parameters

Action Parameterization Default Paramters is a type of service template where shell variables are used in place of parameters for QoS actions such as police, shape, and bandwidth, configurations entered under a class in a QoS policy.

If you deactivate a template service, the system restores the previously active QoS policy. The QoS policy name maybe different but is structurally and functionally identical to the QoS policy active before the template service was activated.

OUT QoS policy active on the session:

```
policy-map output_parent
class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
ip access-list extended voip-acl
 permit ip 10.1.1.0 0.0.0.255 any
ip access-list extended voip-control-acl
 permit ip 10.2.2.0 0.0.0.255 any
class-map match-any voip
 match access-group name voip-acl
!
class-map match-any voip-control
 match access-group name voip-control-acl
!
shell map voice-service1 police_rate=100000 prec_value=4 queue_size=1 in_h=class-default
out_h=class-default
{
    configure terminal
    accounting group $_acctgrp list default
    policy-map $_outgoing
      class voip
       police $police_rate 60625 0 conform-action transmit exceed-action drop violate-action
 drop
 exit
        priority level 1
        queue-limit 8 packets
        set precedence $prec_value
        set cos 6
        aaa-accounting group $_acctgrp
      class voip-control
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

        queue-limit $queue_size packets
        set precedence 6
        aaa-accounting group $_acctgrp
   policy-map $_incoming
      class voip
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 5
        aaa-accounting group $_acctgrp
      class voip-control
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
        aaa-accounting group $_acctgrp
}
```

After you activate voice-service1 on the target session, this is the active OUT policy:

```
policy-map output_parent$class-default$voice-service1><_OUT$class-default class
  class-default
  shape average 10000000
  service-policy output_child$voice-service1><_OUT$class-default
policy-map output_child$voice-service1><_OUT$class-default
  class voip
   police 10000 60625 0 conform-action transmit exceed-action drop violate-action drop
   priority level 1
   queue-limit 8 packets
   set precedence 4
   set cos 6
   aaa-accounting group voice-service1><_GRP
  class voip-control
   police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
   queue-limit 16 packets
   set precedence 6
   aaa-accounting group voice-service1><GRP
 class class-default
```

After you activate voice-service1 on the target session, this is the active IN policy:

```
policy-map input_parent$class-default$voice-service1><_IN$class-default
 class class-default
 police cir 10000000 bc 312500 conform-action transmit exceed-action drop
 service-policy input_child$voice-service1><_IN$class-default
policy-map input_child$voice-service1><_IN$class-default
 class voip
  police 200000 9216 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 5
  aaa-accounting group voice-service1><_GRP
 class voip-control
  police 112000 21000 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 7
  aaa-accounting group voice-service1><_GRP
 class-default
```

Class Name Override

Class name override is a type of service template where shell variables are used in place of parameters for QoS actions such as police, shape, and bandwidth, configurations entered under a class in a QoS policy. Shell variables may also be used in place of class names in service template policy definitions. Shell variables may completely substitute a class name or may be configured as a variable suffix with a constant prefix.

If you deactivate a template service, the system restores the previously active QoS policy. The QoS policy name may be different but is structurally and functionally identical to the QoS policy active before the template service was activated.

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
! Pre-configured ACLs/class-maps
ip access-list extended aol_classifier_acl         ! Locally pre-configured
 permit ip host 10.1.30.194 any
class-map match-all voice-control-aol_classifier_reference      ! Locally pre-configured
  match access-group name aol_classifier_acl
```

```
! Other pre-configured ACLs/classes here (e.g., voice-aol_classifier_reference,
voice-t_online, etc.)
! Service template:
shell map voice-aol-service1 prec_value=3 police_rate=100000 class_ref=t_online
in_h=class-default out_h=class-default
{
    configure terminal
    accounting group $_acctgrp list default
    policy-map $_outgoing
      class voice-control-$class_ref
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

        queue-limit 16 packets
        set precedence 6
        aaa-accounting group $_acctgrp
      class voice-$class_ref
       police $poice_rate 60625 0 conform-action transmit exceed-action drop violate-action
 drop
        priority level 1
        queue-limit 8 packets
        set precedence $prec_value
        set cos 6
        aaa-accounting group $_acctgrp
    policy-map $_incoming
      class voice-control-$class_ref
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
        aaa-accounting group $_acctgrp
      class voice-$class_ref
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence $prec_value
        aaa-accounting group $_acctgrp
}
```

After you activate voice-aol-service1(class_ref=aol_classifier_reference) on the target session, this is the active OUT policy:

```
policy-map
output_parent$class-default$voice-aol-service1<class_ref=aol_classifier_reference>_OUT$class-default

 class class-default
    shape average 10000000
   service-policy
output_child$voice-aol-service1<class_ref=aol_classifier_reference>_OUT$class-default
policy-map
output_child$voice-aol-service1<class_ref=aol_classifier_reference>_OUT$class-default
      class voice-control-aol_classifier_reference     ! Reference to pre-configured class
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

        queue-limit 16 packets
        set precedence 6
        aaa-accounting group voice-aol-service1<class_ref=aol_classifier_reference>_GRP
      class voice-aol_classifier_reference     ! reference to pre-configured class
        police 100000 60625 0 conform-action transmit exceed-action drop violate-action
drop
        priority level 1
        queue-limit 8 packets
        set precedence 3
        set cos 6
        aaa-accounting group voice-aol-service1<class_ref=aol_classifier_reference>_GRP
class class-default
```

After you activate voice-aol-service1(class_ref=aol_classifier_reference) on the target session, this is the active IN policy:

```
policy-map
input_parent$class-default$voice-aol-service1<class_ref=aol_classifier_reference>_IN$class-default

 class class-default
    police cir 10000000 bc 312500 conform-action transmit exceed-action drop
```

```
   service-policy
input_child$voice-aol-service1<class_ref=aol_classifier_reference>_IN$class-default
policy-map input_child$voice-aol-service1<class_ref=aol_classifier_reference>_IN$class-default

      class voice-control-aol_classifier_reference     ! reference to pre-configured class
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
        aaa-accounting group voice-aol-service1<class_ref=aol_classifier_reference>_GRP
      class voice-aol_classifier_reference     ! reference to pre-configured class
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 3
        aaa-accounting group voice-aol-service1<class_ref=aol_classifier_reference>_GRP
       class-default
```

IP Address Parameterization

IP Address Parameterization is a type of Action Parameterization service template in which classifiers may be dynamically modified by adding more entries to ACLs. The entries to be added in an ACL are a list of IP addresses in a shell variable.

If you deactivate a template service, the system restores the previously active QoS policy. The QoS policy name may be different but is structurally and functionally identical to the QoS policy active before the template service was activated.

Note Classes must be predefined; they are not dynamically created.

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
! Base ACLs:
ip access-list extended IPOne-control-acl     ! Base ACL locally pre-configured
  permit ip any host 10.0.132.118
  permit ip host 10.0.132.118 any
  permit ip any host 10.1.245.122
  permit ip host 10.1.245.122 any
ip access-list extended IPOne-combined-acl      ! Base ACL pre-configured
 permit ip any 10.0.132.0 0.0.0.127
 permit ip 10.0.132.0 0.0.0.127 any
 permit ip any 10.1.245.64 0.0.0.63
 permit ip 10.1.245.64 0.0.0.63 any
! Base class-maps:
class-map match-any voice-control      ! Base class map pre-configured
  match access-list name IPOne-control-acl      !   Match on the base ACL
class-map match-any voice      ! base class-map pre-configured
  match access-list name IPOne-combined-acl      !   Match on the base ACL
! Service template:
shell map voice-toi  prec_value=3 police_rate=100000 ip_list=10.2.1.0/28,10.2.1.0/29
in_h=class-default out_h=class-default
{
    configure terminal
    ! Class-map templates:
```

```
    classmap-template voice-control  $ip_list
    classmap-template voice  $ip_list
    ! Service parameter templates:
    policy-map $_outgoing
      class voice-control-$ip_list     ! class names MUST end with -$ip_list
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

       queue-limit 16 packets
       set precedence 6
       aaa-accounting group IPOne-aol
      class voice-$ip_list
       police $police_rate 60625 0 conform-action transmit exceed-action drop violate-action
 drop
       priority level 1
       queue-limit 8 packets
       set precedence $prec_value
       aaa-accounting group IPOne-aol
   policy-map $_incoming
      class voice-control-$ip_list
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
       set precedence 7
       aaa-accounting group IPOne-aol
      class voice-$ip_list
       police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
       set precedence $prec_value
       aaa-accounting group IPOne-aol
```

After you activate voice-toi(ip_list=10.1.30.0/28,10.1.40.0/29) on the target session, this is the active OUT QoS policy :

```
policy-map output_parent$class-default$
voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_OUT$class-default
 class class-default
    shape average 10000000
   service-policy output_child$voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_OUT$class-default
policy-map output_child$voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_OUT$class-default
      class voice-control-10.1.30.0/28,10.1.40.0/29
       police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop

       queue-limit 16 packets
       set precedence 6
       aaa-accounting group IPOne-aol
      class voice-10.1.30.0/28,10.1.40.0/29
       police 100000 60625 0 conform-action transmit exceed-action drop violate-action
drop
       priority level 1
       queue-limit 8 packets
       set precedence 3
       aaa-accounting group IPOne-aol
class class-default
```

After you activate voice-toi(ip_list=10.1.30.0/28,10.1.40.0/29) on the target session, this is the active IN QoS policy :

```
policy-map
input_parent$class-default$voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_IN$class-default
 class class-default
    police cir 10000000 bc 312500 conform-action transmit exceed-action drop
   service-policy input_child$voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_IN$class-default
policy-map input_child$voice-toi>ip_list=10.1.30.0/28,10.1.40.0/29<_IN$class-default
      class voice-control-10.1.30.0/28,10.1.40.0/29
       police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
       set precedence 7
       aaa-accounting group IPOne-aol
      class voice-10.1.30.0/28,10.1.40.0/29
       police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
       set precedence 3
       aaa-accounting group IPOne-aol
```

```
 class-default
```

Note The following configurations are dynamically created.

```
! Internally created ACLs:
ip access-list extended IPOne-control-acl-10.1.30.0/28,10.1.40.0/29
  permit ip any host 10.0.132.118
  permit ip host 10.0.132.118 any
  permit ip any host 10.1.245.122
  permit ip host 10.1.245.122 any
  permit ip 10.1.30.0 0.0.0.15 any  ! ACEs derived from $ip_list
  permit ip any  10.1.30.0 0.0.0.15
  permit ip 10.1.40.0 0.0.0.7 any
  permit ip any 10.1.40.0 0.0.0.7
ip access-list extended IPOne-combined-acl-10.1.30.0/28,10.1.40.0/29
  permit ip any 10.0.132.0 0.0.0.127
  permit ip 10.0.132.0 0.0.0.127 any
  permit ip any 10.1.245.64 0.0.0.63
  permit ip 10.1.245.64 0.0.0.63 any
  permit ip 10.1.30.0 0.0.0.15 any  ! ACEs derived from $ip_list
  permit ip any 10.1.30.0 0.0.0.15
  permit ip 10.1.40.0 0.0.0.7 any
  permit ip any 0.0.0.7 10.1.40.0
! internally created class-maps:
class-map match-any voice-control-10.1.30.0/28,10.1.40.0/29
   match access-group name IPOne-control-acl-10.1.30.0/28,10.1.40.0/29
class-map match-any voice-10.1.30.0/28,10.1.40.0/29
   match access-group name IPOne-combined-acl-10.1.30.0/28,10.1.40.0/29
```

Turbo Button Service

Turbo Button service is a type of Action Parameterization service template in which only policy parameters in the INPUT parent class-default and shape parameters in the OUT parent class-default can be dynamically modified.

This example shows how to create a service template for the Turbo Button service:

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
shell map turbo-button in_police_val=20000000 $out_shape=20000000
configure terminal
accounting group $_acctgrp list default
policy-map $_outgoing
class class-default
shape average $out_shape
aaa-accounting group $_acctgrp
policy-map $_incoming
class class-default
police $in_police_val
aaa-accounting group $_acctgrp
```

Turbo Button Activation

This example shows how to activate the Turbo Button service using the default values.

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
accounting group turbo-button>< list default

accounting group turbo-button>< list default
! Service outgoing:
 policy-map  turbo-button><_OUT
 class class-default
 shape average 20000000
 aaa-accounting group turbo-button>< list default
! Service incoming:
 policy-map  turbo-button><_IN
 class class-default
 police 20000000
aaa-accounting group turbo-button>< list default
```

After you activate the service on the target session, this is the active OUT policy:

```
policy-map output_parent$turbo-button><_OUT$
class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

 class class-default
 shape average 20000000
aaa-accounting group turbo-button>< list default
service-policy
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
policy-map
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
class voip
police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
 priority level 1
 queue-limit 8 packets
 set precedence 5
 set cos 6

aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
 class voip-control
 police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
 queue-limit 32 packets
 set precedence 6
aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
class class-default
```

After you activate the service on the target session, this is the active IN policy:

```
policy-map input_parent$turbo-button>
<_IN$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
class class-default
   police cir 20000000 bc 312500 conform-action transmit exceed-action drop
   aaa-accounting group turbo-button>< list default

service-policy
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
```

```
policy-map
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default

   class voip
        police 200000 9216 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 5
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP

   class voip-control
        police 112000 21000 0 conform-action transmit exceed-action transmit violate-action
 drop
        set precedence 7
       aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
class-default
```

Turbo Button Deactivation

This example shows how to deactivate the Turbo Button service using the default values of VSA 252 0c turbo-button().

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
```

After you activate the service on the target session, this is the active OUT policy:

```
policy-map
output_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
class class-default
  shape average 10000000
  service-policy
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
policy-map
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

  class voip
   police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
   priority level 1
   queue-limit 8 packets
   set precedence 5
   set cos 6
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class voip-control
   police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
   queue-limit 32 packets
   set precedence 6
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP

class class-default
```

After you activate the service on the target session, this is the active IN policy:

```
policy-map
input_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default

   class class-default
   police cir 10000000 bc 312500 conform-action transmit exceed-action drop
   service-policy
```

```
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default

policy-map
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
  class voip
   police 200000 9216 0 conform-action transmit exceed-action transmit violate-action drop
   set precedence 5
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class voip-control
   police 112000 21000 0 conform-action transmit exceed-action transmit violate-action drop

   set precedence 7
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class-default
```

Turbo Button Override

This example shows how to activate the Turbo Button service using the default values of VSA 250 Aturbo-button(in_police_val=30000000, out_shape_val=30000000) (Activation from Access-Accept) or VSA 252 0b turbo-button(in_police_val=30000000, out_shape_val=30000000) (Activation from CoA).

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
accounting group turbo-button>in_police_val=30000000#out_shape_val=30000000 list default

! Service outgoing:
 policy-map turbo-button>in_police_val=30000000#out_shape_val=30000000<_OUT
 class class-default
  shape average 30000000
  accounting group turbo-button>in_police_val=30000000#out_shape_val=30000000
! Service incoming:
  policy-map turbo-button>in_police_val=30000000#out_shape_val=30000000<_IN
  class class-default
   police 30000000
   accounting group turbo-button>in_police_val=30000000#out_shape_val=30000000
```

After you activate the service on the target session, this is the active OUT policy:

```
policy-map output_parent$turbo-button>
in police_val=30000000#out_shape_val=30000000<_OUT$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
class class-default
 shape average 20000000
 accounting group turbo-button>in_police_val=30000000#out_shape_val=30000000
 service-policy
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

 policy-map
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
class voip
 police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
 priority level 1
 queue-limit 8 packets
 set precedence 5
 set cos 6
 aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
```

```
class voip-control
 police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
 queue-limit 32 packets
 aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
class class-default
```

After you activate the service on the target session, this is the active IN policy:

```
policy-map
input_parent$turbo-button>in_police_val=30000000#out_shape_val=30000000<_IN$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
class class-default
 police cir 20000000 bc 312500 conform-action transmit exceed-action drop
 accounting group turbo-button>in_police_val=30000000#out_shape_val=30000000
 service-policy
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
policy-map
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
 class voip
  police 200000 9216 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 5
  aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
 class voip-control
  police 112000 21000 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 7
  aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
 class-default
```

Example Turbo Button Override Deactivation

This example shows how to deactivate the Turbo Button override using the default values of VSA 252 0c turbo-button (in_police_val=30000000, out_shape_val=30000000).

OUT QoS policy active on the session:

```
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
```

IN QoS policy active on the session:

```
policy-map input_parent
 class class-default
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
```

After you activate the service on the target session, this is the active OUT policy:

```
policy-map
output_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default


 class class-default
  shape average 10000000
  service-policy
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default
policy-map
output_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_OUT$class-default

  class voip
   police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
   priority level 1
   queue-limit 8 packets
   set precedence 5
   set cos 6
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class voip-control
   police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
   queue-limit 32 packets
```

```
   set precedence 6
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class class-default
```

After you activate the service on the target session, this is the active IN policy:

```
policy-map
input_parent$class-default$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default

  class class-default
  police cir 10000000 bc 312500 conform-action transmit exceed-action drop
  service-policy
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
  policy-map
input_child$voice-service1>police_rate=200000#prec_value=5#queue_size=32<_IN$class-default
  class voip
   police 200000 9216 0 conform-action transmit exceed-action transmit violate-action drop
   set precedence 5
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class voip-control
   police 112000 21000 0 conform-action transmit exceed-action transmit violate-action drop

   set precedence 7
   aaa-accounting group voice-service1>police_rate=200000#prec_value=5#queue_size=32<_GRP
  class-default
```

Example Overriding Interim Accounting Interval

Overriding Interim Accounting Interval is a type of Action Parameterization service template in which you can use the shell variables in place of interim interval values in the accounting method list definition, allowing the account interim value to be dynamically modified.

This example shows how to do an accounting group override using the default values of: VSA 252 0b voice-service1(police_rate=200000,prec_value=5,acct_interval=600).

This example generates a service with the following parameters:

```
! Global AAA method list and accounting group parameters
aaa accounting network list-600
 action-type start-stop periodic interval 600
  accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600 <_GRP
list list-600
! OUT policy-map:
 policy-map voice-service1>policy_rate=200000#prec_value-5#acct_interval=600 <_OUT
  class voip
   police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
   priority level 1
   queue-limit 8 packets
   set precedence 5
   set cos 6
   aaa-accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_GRP
  class voip-control
   police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
   queue-limit 32 packets
   set precedence 6
   aaa-accounting group


OUT:
policy-map output_parent
 class class-default
   shape average 10000000
   service-policy output_child
policy-map output_child
class class-default
IN:
policy-map input_parent
 class class-default
```

```
   police 10000000
   service-policy input_child
policy-map input_child
 class-default
```

After you activate the service on the target session, this is the active OUT policy:

```
policy-map
output_parent$class-default$voice-service1>policy_rate=200000#prev_value=5#acct_interval=600
 <_OUT$class-default
 class class-default
 shape average 10000000
 service-policy output_child$voice-service1>policy_rate=200000#prev_value=5#acct_interval=600
 <_OUT$class-default
 policy-map output_child$voice-service1>policy_rate=200000#prev_value=5#acct_interval=600
<_OUT$class-default
 class voip
  police 200000 60625 0 conform-action transmit exceed-action drop violate-action drop
  priority level 1
  queue-limit 8 packets
  set precedence 5
  set cos 6
  aaa-accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_GRP
  class voip-control
  police 112000 1000 0 conform-action transmit exceed-action drop violate-action drop
  queue-limit 32 packets
  set precedence 6
  aaa-accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_GRP
 class class-default
```

After you activate the service on the target session, this is the active IN policy:

```
policy-map
input_parent$class-default$voice-service1>policy_rate=200000#prec_value-5#acct_interval=600
 <_IN$class-default
class class-default
 police cir 10000000 bc 312500 conform-action transmit exceed-action drop
 service-policy input_child$voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
 <_IN$class-default
policy-map input_child$voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_IN$class-default
 class voip
  police 200000 9216 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 5
  aaa-accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_GRP
 class voip-control
  police 112000 21000 0 conform-action transmit exceed-action transmit violate-action drop
  set precedence 7
  aaa-accounting group voice-service1>policy_rate=200000#prec_value=5#acct_interval=600
<_GRP
 class class-default
```

Subscriber Accounting Accuracy

The Subscriber Accounting Accuracy feature guarantees that the I/O packet/byte statistics in the Accounting-Stop record are accurate to within one second.

Subscriber accounting data is sent to authentication, authorization, and accounting (AAA) servers during the following events:

- Configured intervals during the lifetime of the session or service
- Service logoff
- Session tear down

Use the **subscriber accounting accuracy** *milliseconds* command to set the value for the Subscriber Accounting Accuracy feature.

Change of Authorization (CoA) ACK Ordering

CoA ACK ordering sends a CoA-ACK for each CoA event before a QoS accounting record is sent for that CoA. A CoA may contain activation or deactivation of single or multiple services.

If a service fails to install on a session the following happens:

- The entire CoA fails.
- The Policy Manager sends a CoA-NAK to the RADIUS server.
- The previous service configuration is restored

If one or more services install before a failure is detected the following happens:

- The entire CoA fails.
- Services are backed out.
- The Policy Manager sends a CoA-NAK to the RADIUS server.
- The previous service configuration is restored.

Multiservice CoAs can compose up of either of the following:

- QoS services—The Policy Manager combines the services into one net-effect policy map. Only one QoS policy is applied to the session for all services. If the policy fails to install, the system restores the session to use the previous policy map. In effect the session is restored to the state prior to the CoA.
- QoS and Intelligent Services Gateway (ISG) services—The Policy Manager applies the ISG service first, then the QoS service. If the QoS policy fails to install, the system restores the session to the previous policy map. Both the ISG and QoS service are rolled back to the previous state.

For multiservice CoA only one CoA-ACK is sent when all services successfully install.

Change of Authorization Rollback

The CoA Rollback feature restores QoS policy accounting to its state before the CoAs were issued. CoA Rollback also properly acknowledges the RADIUS server using a CoA-NAK.

The CoA Rollback feature applies to syntax mistakes and policy install failures such as admission control and resource allocation failure.

If CoA fails, the system sends a CoA-NAK and does not send QoS accounting records. The accounting record for existing services keeps previous counters and continues to count new packets.

QoS Accounting High Availability

When QoS accounting is enabled in a class the policy accounting feature supports three types of events:

- Start—Indicates a new accounting flow. The start record contains statistics and attributes specific to this flow.
- Interim—Indicates how often flow statistics are reported.
- Stop—Indicates the end of an accounting flow. The stop record also contains statistics and attributes specific to this flow.

The policy accounting feature collects the statistics for the accounting flows and sends the information to the RADIUS accounting billing server.

The QoS accounting high availability feature ensures that the start, interim, and stop accounting records are not affected if a planned or unexpected failover occurs. When a planned or unexpected failover occurs the QoS accounting HA feature ensures that the RP switchover occurs without interrupting the flow of information to the RADIUS accounting billing server. The feature also ensures that all QoS services on all active sessions continue without any interruption and that the service accounting counters persist across the RP switchover.

Persistence of Policy Accounting States

To ensure that start, stop, and interim accounting is not affected by a stateful switchover (SSO) or an in-service software upgrade (ISSU), the Policy Manager synchronizes all QoS services and parameterized CoA functionality with the standby RP at the time of the failover. In addition, the dynamic QoS configurations and the polling interval are synchronized between the active and standby RPs.

To synchronize a parameterized CoA event to a standby RP, the Policy Manager performs the following functions:

- Manages the CoA replay to synchronize provisioning events on the standby RP.
- Uses the same service template on both the active and standby RP.
- Creates the same policy map and class map names to apply to the session on both the active ad standby RP.
- Uses predefined QoS policy maps and class maps during service template activation.

Persistence of Policy Accounting Counters

The QoS Accounting HA feature ensures that the policy accounting counters persist across an SSO or failover. After a switchover occurs, the standby RP becomes the active RP and accumulates the statistics from the previously active RP. If the newly active RP receives a periodic update after the switchover it generates an interim record using the statistics it accumulated plus the values from the periodic update. If the newly active RP does not receive a periodic update after the switchover, it generates the interim record using only the statistics it accumulated from the previously active RP.

For more information on SSOs and ISSUs, see the *Cisco IOS High Availability Configuration Guide*.

How to Use QoS Policy Accounting

To use QoS Policy Accounting you must assign a group or AAA method list to a traffic class, then you configure the service template for policy accounting, and finally you activate the subscriber accounting accuracy functionality.

Note By default, QoS Policy Accounting is not assigned to traffic classes.

Assigning a Group or AAA Method List to a Traffic Class

Before You Begin

Ensure the group or AAA method list already exists. If you try to add an undefined group or AAA method list to a traffic class, you will receive an error message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** *list-name method1*
4. **aaa accounting network** *methodlist-name*
5. **action-type start-stop**
6. **periodic interval** *minutes*
7. **accounting group group_name** *list list-name*
8. **policy-map** *policy-map-name*
9. **class class-default**
10. **accounting aaa list** *list-name* [*group-name*]
11. **end**
12. **show policy-map session**
13. **show accounting group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp** *list-name method1*	Specifies a valid AAA authentication method.

	Command or Action	Purpose
	Example: `Router(config)# aaa authentication ppp group radius`	• Group RADIUS enables global RADIUS authentication.
Step 4	**aaa accounting network** *methodlist-name* **Example:** `Router(config)# aaa accounting network list1`	Enables AAA of services when you use RADIUS. • The algorithm determining the interim interval for a class or group uses the method list specified here.
Step 5	**action-type start-stop** **Example:** `Router(config)# action-type start-stop`	Sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process.
Step 6	**periodic interval** *minutes* **Example:** `Router(config)# periodic interval 1`	Adds the interim interval value (1 to 71,582 minutes) in the method list, if specified. • If you do not define an interim interval, the global value defined by AAA is used. • If the method list disables interim updates, the accounting flows using the method list do not generate an interim update.
Step 7	**accounting group group_name** *list list-name* **Example:** `Router(config)# accounting group group_name AAAmethodlist AAAmethodlist1`	Sets properties in the AAA method list. • You can make per-session changes to existing traffic classes by temporarily overwriting properties in the groups or AAA method lists to which they are assigned. This allows you to provide dynamic customized QoS configuration to each subscriber.
Step 8	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map p1`	Creates a policy map.
Step 9	**class class-default** **Example:** `Router(config)# class class-default`	Creates a traffic class.
Step 10	**accounting aaa list** *list-name* [*group-name*]	Assigns the traffic class to a group or an AAA method list.

	Command or Action	Purpose
	Example: `Router(config)# accounting aaa list AAAmethodlist1`	• This example shows the QoS Policy Accounting feature enabled for instances of a traffic class using list AAAmethodlist1 with no group.
Step 11	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 12	**show policy-map session** **Example:** `Router# show policy-map session`	(Optional) Displays QoS Policy Accounting feature information for traffic classes with a group or an AAA method list.
Step 13	**show accounting group** *group-name* **Example:** `Router# show accounting group acc-group1`	(Optional) Displays all group-to-method list associations. • Enter a group name to view information specific to that group.

Activating Subscriber Accounting Accuracy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **subscriber accounting accuracy** *milliseconds*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**subscriber accounting accuracy** *milliseconds* **Example:** `Device(config)# subscriber accounting accuracy 1000`	Sets the value for the Subscriber Accounting Accuracy feature.
Step 4	**end** **Example:** `Device(config)# end`	Enters privileged EXEC mode.

Troubleshooting Service Templates

To troubleshoot any service template issues, you can display usage information for all template service policy maps on your router.

SUMMARY STEPS

1. **enable**
2. **show subscriber policy ppm-shim-db**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show subscriber policy ppm-shim-db** **Example:** `Router(config)# show subscriber policy ppm-shim-db`	Displays reference counts (usage) of all template service policy-maps and Net Effect policy-maps on the router.

Configuration Examples for QoS Policy Accounting

Example: Using the QoS Policy Accounting Feature in Groups

The following example shows grouping:

```
policy-map my-policy
class voip
police
aaa-accounting group premium-services
class voip-control
police
aaa-accounting group premium-services
```

Example: Generating Separate Accounting Streams

The following example shows two classifiers called class voip and class voip-control. The classifiers are assigned to one policy associated with one target. This configuration generates two separate QoS policy accounting streams.

```
policy-map my-policy
class voip
police 200000
accounting aaa list AAA-LIST
class voip-control
police 100000
accounting aaa list AAA-LIST
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands	*Cisco IOS QoS Command Reference*
Cisco IOS High Availability	*Cisco IOS High Availability Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2866	RADIUS Accounting

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the QoS Policy Accounting Feature

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for the QoS Policy Accounting Feature

Feature Name	Releases	Feature Information
QoS Accounting HA	Cisco IOS XE Release 3.5S	The QoS Accounting High Availability (HA) feature ensures that QoS accounting statistics persist, and that the RADIUS accounting billing server continues to report accounting counters during planned and unexpected Route Processor (RP) switchovers. In Cisco IOS XE Release 3.5S, this service was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was modified: **debug qos accounting**
QoS Policy Accounting	Cisco IOS XE Release 2.6 Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.8S	The QoS Policy Accounting feature helps you accurately account for traffic on your system. It also provides greater flexibility in assigning QoS configurations to subscribers. Static CLI-driven accounting is supported. In Cisco IOS XE Release 2.6, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.2S, the service template, subscriber subsecond accuracy, dynamic CoAs, and uninterrupted accounting in case of services untouched by the dynamic activation are supported. The following commands were added: **show subscriber policy ppm-shim-db** and **subscriber accounting accuracy**.