



JunosE™ Software for E Series™ Broadband Services Routers

Classifier Groups, Policy Rules Management, and
Merging Policies

Release

14.1.x



Published: 2012-12-20

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2012, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

JunosE™ Software for E Series™ Broadband Services Routers Classifier Groups, Policy Rules Management, and Merging Policies
Release 14.1.x
Copyright © 2012, Juniper Networks, Inc.
All rights reserved.

Revision History
December 2012—FRS JunosE 14.1.x

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	ix
	E Series and JunosE Documentation and Release Notes	ix
	Audience	ix
	E Series and JunosE Text and Syntax Conventions	ix
	Obtaining Documentation	xi
	Documentation Feedback	xi
	Requesting Technical Support	xi
	Self-Help Online Tools and Resources	xii
	Opening a Case with JTAC	xii
Part 1	Overview	
Chapter 1	Classifier Groups and Policy Rules	3
	Classifier Groups and Policy Rules Overview	3
	Policy Rule Precedence	4
	Packet Tagging Overview	6
	Applying Policy Lists to Interfaces and Profiles Overview	7
	Using RADIUS to Create and Apply Policies Overview	10
	Construction of IPv6 Classifiers from the Hexadecimal Ascend-Data-Filter Attribute	13
	Ascend-Data-Filter Attribute for IPv4/IPv6 Subscribers in a Dual Stack	13
Chapter 2	Merging Policies	17
	Merging Policies Overview	17
	Resolving Policy Merge Conflicts	19
	Merged Policy Naming Conventions	21
	Reference Counting for Merged Policies	22
	Persistent Configuration Differences for Merged Policies Through Service Manager	22
	Policy Attachment Sequence at Login Through Service Manager	22
	Policy Attachment Rules for Merged Policies	23
	Error Conditions for Merged Policies	24
	Parent Group Merge Algorithm	24
	Overlapping Classification for IP Input Policy	26
	Starting Policy Processing	28
	Processing the Classifier Result	28
	Processing the Auxiliary-Input Policy Attachment	29
	Policy Actions	29

Part 2	Configuration	
Chapter 3	Configuration Tasks for Managing Classifier Groups and Policy Rules	35
	Using Policy Rules to Provide Routing Solutions	35
	Configuring Policies to Provide Network Security	36
	Creating an Exception Rule within a Policy Classifier Group	37
	Defining Policy Rules for Forwarding	38
	Forwarding Based on Next-Hop Addresses for Input IPv4 and IPv6 Policies	39
	Assigning Values to the ATM CLP Bit	41
	Enabling ATM Cell Mode	42
	Enabling IP Options Filtering	42
	Creating Multiple Forwarding Solutions with IP Policy Lists	43
	Creating a Classifier Group for a Policy List	44
Chapter 4	Configuration Tasks for Merging Policies	47
	Merging Policies	47
Chapter 5	Examples	59
	Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers	59
	Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers	64
Part 3	Administration	
Chapter 6	Monitoring Tasks	71
	Monitoring Color-Mark Profiles	71
Part 4	Index	
	Index	75

List of Figures

Part 1	Overview	
Chapter 2	Merging Policies	17
	Figure 1: Input Policy with Primary Stage and Auxiliary Substage	28

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	x
	Table 2: Text and Syntax Conventions	x
Part 1	Overview	
Chapter 1	Classifier Groups and Policy Rules	3
	Table 3: Policy Rule Commands and Precedence	4
	Table 4: Ascend-Data-Filter Fields	11
Chapter 2	Merging Policies	17
	Table 5: Input Action and Secondary Input Actions	30
Part 2	Configuration	
Chapter 5	Examples	59
	Table 6: Ascend-Data-Filter Attribute for an Input Policy on an IPv4 Interface	59
	Table 7: Ascend-Data-Filter Attribute Values for a RADIUS Record	63
	Table 8: Ascend-Data-Filter Attribute for an Output Policy on an IPv6 Interface	65
	Table 9: Ascend-Data-Filter Attribute for an Input Policy on an IPv6 Interface	66

About the Documentation

- E Series and JunosE Documentation and Release Notes on page ix
- Audience on page ix
- E Series and JunosE Text and Syntax Conventions on page ix
- Obtaining Documentation on page xi
- Documentation Feedback on page xi
- Requesting Technical Support on page xi

E Series and JunosE Documentation and Release Notes

For a list of related JunosE documentation, see
<http://www.juniper.net/techpubs/software/index.html>.

If the information in the latest release notes differs from the information in the documentation, follow the *JunosE Release Notes*.

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at
<http://www.juniper.net/techpubs/>.

Audience

This guide is intended for experienced system and network specialists working with Juniper Networks E Series Broadband Services Routers in an Internet access environment.

E Series and JunosE Text and Syntax Conventions

Table 1 on page x defines notice icons used in this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page x defines text and syntax conventions that we use throughout the E Series and JunosE documentation.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents commands and keywords in text.	<ul style="list-style-type: none"> Issue the clock source command. Specify the keyword exp-msg.
Bold text like this	Represents text that the user must type.	host1(config)#traffic class low-loss1
Fixed-width text like this	Represents information as displayed on your terminal's screen.	host1#show ip ospf 2 Routing Process OSPF 2 with Router ID 5.5.0.250 Router is an Area Border Router (ABR)
<i>Italic text like this</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies variables. Identifies chapter, appendix, and book names. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>clusterId</i>, <i>ipAddress</i>. <i>Appendix A, System Specifications</i>
Plus sign (+) linking key names	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Syntax Conventions in the Command Reference Guide		
Plain text like this	Represents keywords.	terminal length
<i>Italic text like this</i>	Represents variables.	<i>mask</i> , <i>accessListName</i>

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
(pipe symbol)	Represents a choice to select one keyword or variable to the left or to the right of this symbol. (The keyword or variable can be either optional or required.)	diagnostic line
[] (brackets)	Represent optional keywords or variables.	[internal external]
[]* (brackets and asterisk)	Represent optional keywords or variables that can be entered more than once.	[level1 level2 l1]*
{ } (braces)	Represent required keywords or variables.	{ permit deny } { in out } { clusterId ipAddress }

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the Technical Documentation page on the Juniper Networks Web site at <http://www.juniper.net/>.

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the Portable Libraries page at

<http://www.juniper.net/techpubs/resources/index.html>

Copies of the Management Information Bases (MIBs) for a particular software release are available for download in the software image bundle from the Juniper Networks Web site at <http://www.juniper.net/>.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract,

or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Overview

- [Classifier Groups and Policy Rules on page 3](#)
- [Merging Policies on page 17](#)

CHAPTER 1

Classifier Groups and Policy Rules

- [Classifier Groups and Policy Rules Overview on page 3](#)
- [Policy Rule Precedence on page 4](#)
- [Packet Tagging Overview on page 6](#)
- [Applying Policy Lists to Interfaces and Profiles Overview on page 7](#)
- [Using RADIUS to Create and Apply Policies Overview on page 10](#)

Classifier Groups and Policy Rules Overview

Classifier groups contain the policy rules that make up a policy list. A policy rule is an association between a policy action and an optional CLACL. The CLACL defines the packet flow on which the policy action is taken.

A policy list might contain multiple classifier groups—you can specify the precedence in which classifier groups are evaluated. Classifier groups are evaluated starting with the lowest precedence value. Classifier groups with equal precedence are evaluated in the order of creation.



NOTE: For IP policies, the **forward** command supports the **order** keyword, which enables you to order multiple forward rules within a single classifier group. (See [“Using Policy Rules to Provide Routing Solutions” on page 35.](#))

From Policy Configuration mode, you can assign a precedence value to a CLACL by using the **precedence** keyword when you create a classifier group. The default precedence value is 100. For example:

```
host1(config-policy-list)#classifier-group ipCLACL25 precedence 21
host1(config-policy-list-classifier-group)#
```

The **classifier-group** command puts you in Classifier Group Configuration mode. In this mode you configure the policy rules that make up the policy list. For example:

```
host1(config-policy-list-classifier-group)#forward next-hop 172.18.20.54
```

To stop and start a policy rule without losing statistics, you can suspend the rule. Suspending a rule maintains the policy rule with its current statistics, but the rule no longer affects packets in the forwarding path.

From Classifier Group Configuration mode, you can suspend a rule by using the **suspend** version of that policy rule command. The **no suspend** version reactivates a suspended rule. For example:

```
host1(config-policy-list-classifier-group)#suspend forward next-hop 172.18.20.54
host1(config-policy-list-classifier-group)#no suspend forward next-hop 172.18.20.54
```

You can add, remove, or suspend policy rules while the policy is attached to one or more interfaces. The modified policy takes effect once you exit Policy Configuration mode.

Related Documentation

- [Policy Rule Precedence on page 4](#)

Policy Rule Precedence

Because of the flexibility in creating policy lists and classifier groups, you can configure a classifier group that has multiple policy rules.

If a classifier group has multiple rules, the router uses the rules according to their precedence—not in the order in which you created the rules. The first rule listed (the forward rule) for a policy list type has the highest precedence and the last rule has the lowest. The precedence is based on the order in which the router performs rules. Rules are performed in order from lower to higher precedence. In the event of a conflict, a higher precedence rule overrides the lower precedent rule.

The precedence of rules is important if you want a specific rule to be applied. For example, if an IP policy list has both a rate-limit-profile rule (which specifies a color) and a color rule in the same classifier-group, the color specified by the color rule is always used rather than the color implied in the rate-limit-profile rule (the color rule has a higher precedence).

[Table 3 on page 4](#) lists the policy rule commands that you can use for each type of policy list. The table lists the rules in their order of precedence.



NOTE: The ES2 10G Uplink LM and the ES2 10G LM support only IP, MPLS, and VLAN interfaces.

Table 3: Policy Rule Commands and Precedence

ATM	Frame Relay	GRE	IP	IPv6	L2TP	MPLS	VLAN
forward	forward	forward	forward	forward	forward	forward	forward
color	color	color	forward interface (input, secondary input, and output policies only)	forward next-hop (for input policies only)	color	color	color

Table 3: Policy Rule Commands and Precedence (*continued*)

ATM	Frame Relay	GRE	IP	IPv6	L2TP	MPLS	VLAN
–	–	–	exception for input and secondary input policies only (not supported on ES2 10G Uplink LM)	–	–	–	–
mark-clp (See mark-clp in the <i>JunosE Command Reference Guide</i> for platform support information.)	mark-de	mark	forward next-hop (for input policies only)	color	rate-limit-profile	rate-limit-profile	mark-user-priority
filter	filter	filter	color	rate-limit-profile	filter	mark-exp	filter
user-packet-class	user-packet-class	user-packet-class	rate-limit-profile	user-packet-class	user-packet-class	filter	user-packet-class
traffic-class	traffic-class	traffic-class	user-packet-class	traffic-class	traffic-class	user-packet-class	traffic-class
–	–	–	traffic-class	mark	–	traffic-class	–
–	–	–	mark	filter	–	–	–
–	–	–	filter	–	–	–	–
–	–	–	log (not supported on ES2 10G Uplink LM or ES2 10G LM)	–	–	–	–



NOTE: The commands listed in this section replace the Policy List Configuration mode versions of the commands. For example, the `color` command replaces the Policy List Configuration mode version of the `color` command. The original command may be removed completely in a future release.

- Related Documentation**
- [Classifier Groups and Policy Rules Overview on page 3](#)
 - Monitoring Policy Management Overview
 - color
 - color-mark-profile
 - filter
 - forward
 - forward interface
 - forward next-hop
 - green-mark
 - log
 - mark
 - mark-clp
 - mark-de
 - mark-exp
 - mark-user-priority
 - next-hop
 - next-interface
 - rate-limit-profile
 - red-mark
 - reference-rate
 - traffic-class
 - user-packet-class
 - yellow-mark

Packet Tagging Overview

You can use the traffic-class rule in policies to tag a packet flow so that the QoS application can provide traffic-class queuing. Policies can perform both in-band and out-of-band packet tagging:

- Policies perform in-band tagging by using their respective mark rule to modify a packet header field. For example, IP policies use the **mark** rule to modify an IP packet header ToS field, and Frame Relay policies use the **mark-de** rule to modify the DE bit.
- Policies perform out-of-band tagging by using the traffic class or color rule. Explicit packet coloring lets you configure prioritized packet flows without having to configure a rate-limit profile. The router uses the color to queue packets for egress queue threshold dropping as described in *Creating Rate-Limit Profiles*.

For example, an Internet service provider (ISP) provides a Broadband Remote Access Server (B-RAS) service that has both video and data components, and the ISP wants to guarantee that the video traffic gets priority treatment relative to the data traffic. The ISP's users have a 1.5 Mbps virtual circuit (VC) terminating on a digital subscriber line access multiplexer (DSLAM). The ISP wants to allocate 800 Kbps of this link for video, if there is a video stream.

The ISP creates a classifier list to define a video packet flow, creates a policy to color the packets, and applies the policy to the interface:

```
host1(config)#ip classifier-list video ip any any dsfield 16
host1(config)#ip classifier-list data ip any any dsfield 32
host1(config)#ip policy-list colorVideoGreen
host1(config-policy-list)#classifier-group video
host1(config-policy-list-classifier-group)#color green
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group data
host1(config-policy-list-classifier-group)#color yellow
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 12/1.1
host1(config-if)#ip policy input colorVideoGreen statistics enabled
```

**Related
Documentation**

- classifier-group
- color
- ip classifier-list
- ip policy-list

Applying Policy Lists to Interfaces and Profiles Overview

You can assign a policy list to supported interfaces and profiles. Policy lists are supported on Frame Relay, IP, IPv6, GRE tunnel, MPLS layer 2, and VLAN interfaces. You can also specify IP, IPv6, and L2TP policies in profiles to assign a policy list to an interface. In either case, you can enable or disable the recording of statistics for bytes and packets affected by the assigned policy.

You can also preserve statistics when you attach a new policy that has a classifier list that is the same for both the original and the new policy attachments.

You can use policy commands to assign an ATM, Frame Relay, GRE tunnel, IP, IPv6, MPLS, or VLAN policy list to an interface. Also, you can use them to specify an IP, IPv6, or L2TP policy list to a profile, which then assigns the policy to the interfaces to which the profile is attached

**NOTE:**

- The `mpls policy` command is used to attach policies to MPLS Layer 2 circuits only.
- The SRP module Fast Ethernet port does not support policy attachments, nor can the module be the destination for the forward next-hop, forward next-interface, next-hop, and next-interface commands



NOTE: Some of the VLAN subinterfaces on a line module that are in the dormant state are deleted even before the maximum number of VLAN subinterfaces supported on the line module is reached. Such a deletion of VLAN subinterfaces in the dormant state enables input and output policy attachments to the other VLAN subinterfaces that are in the active state to occur successfully. For example, a number of subscribers might be disconnected from VLAN subinterfaces and after the maximum number of supported VLAN subinterfaces is exceeded on a line module, a certain number of clients might be logged in again. In such cases, the deletion of some of the dormant VLAN subinterfaces enables successful attachment of input and output policies to the VLAN subinterfaces for the subscribers that newly logged in.

The Ethernet application on the interface controller starts a timer for 8 milliseconds and deletes the dormant VLAN subinterfaces within this period. The number of dormant Ethernet VLAN subinterfaces that are deleted varies depending on the processor load of the line module.

Use the input or output keyword to assign the policy list to the ingress or egress of the interface. For ATM, IP, and IPv6 policy lists, use the secondary-input keyword to assign the policy list, after route lookup, to data destined for local or remote destinations. For IP and IPv6 policy lists, use the secondary-input keyword to assign the policy list, after route lookup, to data destined to local or remote destinations. The router supports secondary input policies whose principal applications are:

- To defeat denial-of-service attacks directed at a router's local IP or IPv6 stack
- To protect a router from being overwhelmed by legitimate local traffic
- To apply policies on packets associated with the route class



NOTE: The local-input keyword for the ip policy and ipv6 policy commands is deprecated, and may be completely removed in a future release. We recommend you remove the keyword from scripts. Re-create any local input policies using the ip classifier-list local true command and attaching the policies using the ip policy secondary-input command.

You can enable or disable the recording of routing statistics for bytes and packets affected by the policy. If you enable statistics, you can enable or disable baselining of the statistics. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved. You must also enable baselining on the interface with the appropriate baseline command.



NOTE: The `gre-tunnel policy` command does not support the `baseline` keyword.

You can use the `preserve` keyword to save the existing statistics when you attach a policy to an interface that already has a policy attached. This keyword saves the statistics for any classifier-list that is the same for both the new and old policy attachments. Without the `preserve` keyword, all statistics are deleted when you attach the new policy.

For example, when you replace a policy attachment that references the original policy-list `plOne` with a new attachment referencing policy-list `plTwo`, the existing statistics for the classifier group referencing `clOne` and the default classifier group are saved.

Original Policy Attachment	New Policy Attachment	Comment
<code>ip policy-list plOne</code>	<code>ip policy-list plTwo</code>	-
<code>ip classifier-list clOne</code>	<code>ip classifier-list clOne</code>	statistics from <code>plOne</code> are saved
<code>Forward</code>	<code>Forward</code>	-
<code>ip classifier-list clTwo</code>	<code>ip classifier-list clFour</code>	-
<code>Forward</code>	<code>Forward</code>	-
<code>ip classifier-list clThree</code>	<code>ip classifier-list clFive</code>	-
<code>Forward</code>	<code>Forward</code>	-
<code>classifier-list *</code>	<code>classifier-list *</code>	statistics from <code>plOne</code> are saved
<code>Filter</code>	<code>Filter</code>	-

You can use the `merge` keyword to enable merging of multiple policies to form a single policy.

```
host1(config)#vlan policy input VlanPolicy33 statistics enabled preserve
```

```
host1(config)#ipv6 policy secondary-input my-policy
```

To assign the policy list named `routeForXYZCorp` with statistics enabled to the ingress IP interface over an ATM subinterface:

```
host1(config)#interface atm 12/0.1
```

```
host1(config)#ip policy input routeForXYZCorp statistics enabled
```

To create an L2TP profile that applies the policy list routeForABCCorp to the egress of an interface:

```
host1(config)#profile bostonProfile
host1(config)#l2tp policy output routeForABCCorp
```

**Related
Documentation**

- atm policy
- frame-relay policy
- gre-tunnel policy
- interface atm
- ip policy
- ipv6 policy
- l2tp policy
- mpls policy
- profile
- vlan policy

Using RADIUS to Create and Apply Policies Overview

E Series routers enable you to use RADIUS to create and apply policies on IPv4 and IPv6 interfaces. This feature supports the Ascend-Data-Filter attribute [242] through a RADIUS vendor-specific attribute (VSA) that specifies a hexadecimal field. The hexadecimal field is encoded with policy attachment, classification, and policy action information

The policy defined in the Ascend-Data-Filter attribute is applied when RADIUS receives a client authorization request and replies with an Access-Accept message.

When you use RADIUS to apply policies, a subset of the router's classification fields and actions is supported. The supported actions and classification fields are:

- Actions
 - Filter
 - Forward
 - Packet marking
 - Rate limit
 - Traffic class
- Classifiers
 - Destination address
 - Destination port

- Protocol
- Source address
- Source port



NOTE: An E Series router dynamically assigns names to the new classifier list and policy list as described in [“Ascend-Data-Filter Attribute for IPv4/IPv6 Subscribers in a Dual Stack” on page 13](#).

To create a policy, you use hexadecimal format to configure the Ascend-Data-Filter attribute on the RADIUS server. For example:

```
Ascend-Data-Filter="01000100 0A020100 00000000 18000000 00000000
00000000"
```

[Table 4 on page 11](#) lists the fields in the order in which they are specified in the hexadecimal Ascend-Data-Filter attribute.

Table 4: Ascend-Data-Filter Fields

Action or Classifier	Format	Comments
Type	1 byte	1=IPv4 3=IPv6
Filter or forward	1 byte	0=filter 1=forward
Indirection	1 byte	0=egress 1=ingress
Spare	1 byte	-
Source IP address	4 bytes for IPv4 16 bytes for IPv6	-
Destination IP address	4 bytes for IPv4 16 bytes for IPv6	-
Source IP prefix	1 byte	Type 1 = Number of leading zeros in the wildcard mask Type 3 = Higher-order contiguous bits of the address that comprise the network portion of the address
Destination IP prefix	1 byte	Type 1 = Number of leading zeros in the wildcard mask Type 3 = Higher-order contiguous bits of the address that comprise the network portion of the address

Table 4: Ascend-Data-Filter Fields (*continued*)

Action or Classifier	Format	Comments
Protocol	1 byte	-
Established	1 byte	Non implemented
Source port	2 bytes	-
Destination port	2 bytes	-
Source port qualifier	1 byte	0= no compare 1= less than 2= equal to 3= greater than 4= not equal to
Destination port qualifier	1 byte	0= no compare 1= less than 2= equal to 3= greater than 4= not equal to
Reserved	2 bytes	-
Marking value	1 byte	Type of Service (ToS)—for IPv4 Differentiated Services Code Point (DSCP)—for IPv6
Marking mask	1 byte	0= no packet marking
Traffic class	1–41 bytes	<ul style="list-style-type: none"> • 0= no traffic class (required if there is no profile) • First byte specifies the length of the ASCII name of the traffic class • Traffic class must be statically configured • Name can optionally be null terminated, which consumes 1 byte • Although the traffic class name field supports up to 41 bytes, you can create an Ascend-Data-Filter attribute with the traffic class name field set to a maximum of 32 bytes only (including null characters). This restriction occurs because the traffic class group configuration enables a traffic class name of up to 31 characters only.

Table 4: Ascend-Data-Filter Fields (*continued*)

Action or Classifier	Format	Comments
Rate-limit profile	1–41 bytes	<ul style="list-style-type: none"> • 0= no rate limit (required if there is no profile) • First byte specifies the length of the ASCII, followed by the ASCII name of the profile • Profile must be statically configured • Name can optionally be null terminated, which consumes 1 byte



NOTE: To create a rate-limit profile, traffic class, or marking rule, you must first configure the filter/forward field as forward.

A single RADIUS record can contain two policies—one ingress policy and one egress policy. Each policy can have a maximum of 512 ascend-data filters. Each ascend data-filter creates a classifier group and the action associated with the classifier group.

Construction of IPv6 Classifiers from the Hexadecimal Ascend-Data-Filter Attribute

If both the source and destination IP prefixes are 128, the IPv6 classifier is created using the IPv6 host argument as follows:

```
IPv6 classifier-list testipv6 source-host 2001:db8:85a3::8a2e:370:7334 destination-host
2001:db8::1428:57ab
```

If either the source or destination IP prefix is non-zero, but less than 128 bits, (for example, 64 bits), the IPv6 classifier is created using the IPv6 address argument as follows:

```
IPv6 classifier-list v6cl4 source-address 2001:db8:85a3::8a2e:370:7334/64
destination-address 2001:db8::1428:57ab/64
```



NOTE: In JunosE Release 10.1.x and earlier, the maximum width of a CAM hardware classifier entry for IPv4 or IPv6 in a single policy was 128 bits. In JunosE Release 10.2.x and later, based on the size limit for a combined IPv6 classifier entry, a maximum of 336 bits of CAM entry is supported for full IPv6 classification with an additional 16 bits for rule set ID. However, OC48/STM16 line modules on ERX14xx models, ERX7xx models, and the ERX310 router support only 128-bit IPv6 classification. For more information on size limits for IP and IPv6 classifiers, see [Size Limit for IP and IPv6 CAM Hardware Classifiers](#).

Ascend-Data-Filter Attribute for IPv4/IPv6 Subscribers in a Dual Stack

The PPP link between the customer premises equipment (CPE) and the provider edge (PE) device or E Series router equipment might require both IPv4 and IPv6 protocols for transmission of data. Such networks require that PE devices run a dual stack of IPv4 and IPv6 services. Dual-stack routers allow simultaneous support for both IPv4 and IPv6 applications. The following guidelines are used to create a policy defined in the Ascend-Data-Filter attribute when IPv4 and IPv6 subscribers are in a network:

- If a subscriber requires only IPv4 services, only the Type 1 action is used in the Access-Accept message returned from the RADIUS server in response to the client authentication request.
- If a subscriber requires only IPv6 services, only the Type 3 action is used in the Access-Accept message returned from the RADIUS server.
- If both IPv4 and IPv6 addresses are assigned to the subscriber interface, then either Type 1 or Type 3 or both the actions are used in the Access-Accept message.
- If the Type 1 action is used and the Indirection action field is set to 01 in the Ascend-Data-Filter attribute, one primary input policy is created and applied on the ingress IPv4 interface.
- If the Type 3 action is used and the Indirection action field is set to 01 in the Ascend-Data-Filter attribute, one primary input policy is created and applied on the ingress IPv6 interface.
- If the Type 1 action is used and the Indirection action field is set to 00 in the Ascend-Data-Filter attribute, one primary output policy is created and applied on the egress IPv4 interface.
- If the Type 3 action is used and the Indirection action field is set to 00 in the Ascend-Data-Filter attribute, one primary output policy is created and applied on the egress IPv6 interface.
- Ascend-Data-Filter attributes for both IPv4 and IPv6 interfaces are stored on the RADIUS server and the appropriate policies are created and applied to the corresponding interfaces when they come up, depending on the type of subscribers.

In lower-numbered releases, the formats of the input and output classifier list names and policy list names were as follows:

- clin_<InterfaceId>_<filterNum>
- clout_<InterfaceId>_<filterNum>
- plin_<InterfaceId>
- plout_<InterfaceId>

where:

- clin—Classifier list included in an input policy list
- clout—Classifier list included in an output policy list
- plin—Policy list applied to the ingress interface
- plout—Policy list applied to the egress interface
- InterfaceId—A unique identifier for the interface to which the policy is applied
- filterNum—A value that denotes the sequence of Ascend-Data-Filter attribute configured on the RADIUS server

In this release, the formats of the input and output classifier list names and policy list names are modified to support IPv6 subscribers. The following is the new format of the input and output classifier list and policy list:

- clin_<AuthId>_<filterNum>
- clout_<AuthId>_<filterNum>
- plin_<ip/ipv6>_<AuthId>
- plout_<ip/ipv6>_<AuthId>

where:

- AuthId—A unique identifier that is used during the authentication of the client with the RADIUS server
- ip/ipv6—Type of protocol used based on the Type action field

**Related
Documentation**

- [Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers on page 59](#)
- [Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers on page 64](#)

CHAPTER 2

Merging Policies

- [Merging Policies Overview on page 17](#)
- [Resolving Policy Merge Conflicts on page 19](#)
- [Merged Policy Naming Conventions on page 21](#)
- [Reference Counting for Merged Policies on page 22](#)
- [Persistent Configuration Differences for Merged Policies Through Service Manager on page 22](#)
- [Policy Attachment Sequence at Login Through Service Manager on page 22](#)
- [Policy Attachment Rules for Merged Policies on page 23](#)
- [Error Conditions for Merged Policies on page 24](#)
- [Parent Group Merge Algorithm on page 24](#)
- [Overlapping Classification for IP Input Policy on page 26](#)

Merging Policies Overview

Merging policies enables you to create multiple policy attachments at an attachment point, resulting in a merged policy that is created and attached at this interface. Executing more than one policy attachment command with the same attachment type at an interface triggers a policy merge through the CLI.

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. 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 are persistent on subsequent reloads.

An interface and an attachment type identify an attachment point. The policies referenced by the component attachments merge into a new policy, which then attaches at the attachment point. The set of component policies are ordered alphabetically by name. This order determines how any merge conflicts are resolved, with the most recently executed command taking precedence.

With policy merging, a set of policies is combined to form a single new policy, which is a union of all the component policies. Classifier groups and policy rules from each component combine to create the merged policy as in the following example:

```
host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input p1 statistics enable merge
host1(config-subif)#ip policy input p2 statistics enable merge
host1(config-subif)#ip policy input p3 statistics enable merge
host1(config-subif)#ip policy output p4 statistics enable merge
host1(config-subif)#ip policy output p5 statistics enable merge
host1(config-subif)#exit
```

The example internally results in the following, where policies p1 + p2 + p3 = mpl_10 and policies p4 + p5 = mpl_11.

```
interface atm 5/0.1
ip policy input mpl_10 statistics enable merge
ip policy output mpl_11 statistics enable merge
exit
```

The classifier list referenced by the classifier group is neither split or merged. If a merged policy already exists for a set of component policies, then the merged policy is used for the attachment. An attachment enables a merged policy to have one or more attachments.

The CLI and the Service Manager applications are the only clients of policy management that can request merging of policy attachments. With policy merging, classifier groups and policy rules from each component policy combine into the merged policy.

Policy merging follows these rules:

- The Classifier list referenced by the classifier group cannot be split or merged.
- Policy merging combines classifier groups from all component policies into the merged policy. In the previous example, policies p1, p2, and p3 are the component policies and mpl_10 is the merged policy. The merge policy is created as if all CLI commands for each component policy are run in the context of the merged policy. The merged policy result is the sum of all commands executed in the respective component policies CLI context in a predetermined merge order.
- If a merged policy already exists for a set of component policies, the merged policy is used for the attachment instead of creating a new one. This functionality allows a merged policy to have one or more attachments. A merge policy is automatically deleted when the last reference is removed.

The following restrictions apply to policy merging:

- Classifier lists cannot be merged.
- Secure policies cannot be merged.
- Policies created using ascend-data-filters cannot be merged.
- Existing policy VSAs in RADIUS are not changed; attachments created by this method cannot be merged. Ascend data filter policies can be attached at input and output attachment points.

- SNMP support for polling statistics based on component policy attachments is not available.
- The merge policy naming convention is not configurable.

**Related
Documentation**

- [Error Conditions for Merged Policies on page 24](#)
- [Merging Policies on page 47](#)
- [Merged Policy Naming Conventions on page 21](#)
- [Resolving Policy Merge Conflicts on page 19](#)

Resolving Policy Merge Conflicts

The set of component policies are first ordered by their name to form the final merged policy. For example, if the component policies sets contain cp_1, cp_3, cp_9, cp_2, the order in which these policies are merged is cp_1, cp_2, cp_3, and cp_9. The merge order is important for resolving merge conflicts.

Various conflicting combinations of component policies can result in a merged policy that is not a perfect union of the component policies. These conflicts are resolved as they currently are in policy CLI context, where, in any conflict, the most recently executed command takes precedence.

More than one component policy can contain the same classifier group. If the precedence does not match, the precedence of the classifier group defined in the last component policy becomes the final precedence for this classifier group in the merged policy, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 100
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C1 precedence 130
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
```

If you combine p1, p2, and p3, you get the following with p1, p2, p3 as the merge order for the set of component policies.

```
ip policy-list mpl_10
classifier-group C1 precedence 130
forward
exit
```

For IP, the forward, filter, next-hop, and next-interface rules are mutually exclusive within a classifier group. For all other types, filter and forward rules are mutually exclusive.

A conflict arises when more than one component policy has the same classifier group and when the rule sets defined in these classifier groups conflict. To resolve the merge conflict, the last command entered replaces any previous conflicting commands for a classifier group, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#next-hop 1.1.1.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#filter
host1(config-classifier-group)#exit
```

Combining p1 and p2 internally results in:

```
ip policy-list mpl_20
classifier-group C1 precedence 90
next-hop 1.1.1.1
exit
```

Combining p2 and p3 internally results in:

```
ip policy-list mpl_21
classifier-group C1 precedence 90
filter
exit
```

Combining p1, p2, and p3 internally results in:

```
ip policy-list mpl_22

classifier-group C1 precedence 90
filter
exit
```

If you have the same policy rule with different parameters, the parameter of the last rule entered with the same type is used, with the exception of IP forward rule, to resolve the conflict, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#color red
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#color yellow
host1(config-classifier-group)#exit
```

Combining p1 and p2 internally results in:

```
ip policy-list mpl_20
classifier-group C1 precedence 90
color yellow
```


exit

With the IP policy forward rule, when more forward rules are added to an existing classifier group, the list of forward rules is created. This is also true during merging, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-hop 1.1.1.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-interface atm 5/0.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-interface fastEthernet 4/0.1
                        next-hop 1.1.1.2
host1(config-classifier-group)#exit
```

Combining p1, p2, and p3, internally results in the following:

```
ip policy-list mpl_10
classifier-group C1 precedence 90
forward next-hop 1.1.1.1
forward next-interface atm 5/0.1
forward next-interface fastEthernet 4/0.1 next-hop 1.1.1.2
exit
```

Policy management enables multiple policy attachments at the same attachment point, which results in a merged policy that is created and attached at the specified attachment point. The logical OR of the **statistics** and **baseline** keywords of all attachments are used as the **statistics** and **baseline** keyword for the merged policy attachment, as in the following example:

```
host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input p1 statistics enable baseline enable merge
host1(config-subif)#ip policy input p2 merge
host1(config-subif)#ip policy input p3 statistics enable merge
host1(config-subif)#exit
```

Results in the following:

```
interface atm 5/0.1
ip policy input mpl_5 statistics enable baseline enable merge
exit
```

Related Documentation

- [Merging Policies Overview on page 17](#)

Merged Policy Naming Conventions

Merged policies are dynamically created. The naming convention is `mpl_hex_of_internally_generated_policy ID`, such as `mpl_10`. If the newly generated name already exists, then a sequence number is appended to the new name to make it unique. The sequence number starts at 1 and increments until the name is unique, such as `mpl_10_2`.

Related Documentation

- [Merging Policies Overview on page 17](#)

Reference Counting for Merged Policies

The reference counts in all containers referenced within a merged policy are incremented by the number of times they are referenced within the merged policy. Also, the reference counts of all component policies of a merged policy are incremented because of the association of the component policies with the merged policy. This means you cannot delete a component policy while a merged policy is still associated with it.

Related Documentation

- [Merging Policies Overview on page 17](#)

Persistent Configuration Differences for Merged Policies Through Service Manager

Service Manager can specify whether a component policy attachment is nonvolatile. 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 nonvolatile interface can have both volatile and nonvolatile 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.

Related Documentation

- [Merging Policies Overview on page 17](#)
- [Policy Attachment Sequence at Login Through Service Manager on page 22](#)

Policy Attachment Sequence at Login Through Service Manager

During a user login, you can specify policy attachments through Service Manager, RADIUS, and Interface Profile. The order that is used to select the policy attachment source is Service Manager, RADIUS, and Interface Profile.

For example, if you configure Ingress-Policy-Name VSA for a user in RADIUS and also have a profile with an input policy reference applied to this user's interface column, when the user logs in, the RADIUS VSA is selected as the source for the input policy attachment. If you also have service profiles applied to the user's interface column, the service profiles override both RADIUS VSA and the policy name specified in the interface profile.



NOTE: Policy merging is not supported with ascend data filter policies.

Policy management does not reselect the source if the policy attachment fails for the selected source. If the policy attachment via service profiles fails, policy management does not reselect RADIUS VSA as the next source. This means the interface does not have any input policy attachment.

- Related Documentation**
- [Persistent Configuration Differences for Merged Policies Through Service Manager on page 22](#)
 - [Policy Attachment Rules for Merged Policies on page 23](#)

Policy Attachment Rules for Merged Policies

The attributes of a policy attachment are as follows:

- Policy name—Name of policy to be attached.
- Attachment type—Type of attachment.
- Statistics enable/disable—Enable or disable statistics for the attachment.
- Baseline enable/disable—Enable or disable baselining for the attachment.
- Merge or Replace—Allow an attachment to become merge-capable and merge with any other attachments that are merge-capable. If the **merge** keyword is not specified, then it replaces any existing attachments with the new attachment. Merging always preserves statistics.
- Preserve—Preserve statistics from earlier attachment when replacing an attachment. This keyword is mutually exclusive with **merge** keyword.

Various possibilities result from a policy attachment at an interface due to the presence or absence of these keywords. The same rules apply while attaching policies based on interface profiles provided by Service Manager except as noted.

Attachments made through Interface Configuration mode follow these rules:

- If an attachment is issued with the **merge** keyword specified:
 - Any existing attachment of the same type at the interface without the **merge** keyword is replaced by the new attachment, which then becomes merge-capable.
 - An attachment is merged with any existing attachments of the same type that have the **merge** keyword set. If a merged policy already exists for the set of component policies, then this merged policy is used or a new merged policy is created dynamically and attached. The statistics for common classifier groups are preserved when replacing the existing merged attachment.
- If an attachment is issued when no **merge** or **preserve** keyword is set, then it replaces all other attachments with the same type at the interface. This attachment is not merge-capable for future use and statistics from previous attachments are not preserved.
- If an attachment is issued when the **merge** keyword is not set, but the **preserve** keyword is set, it replaces all other attachments with the same type at the interface. This attachment is not merge-capable for future use. Statistics from existing attachments are preserved for all the common classifier-groups.

- You cannot have multiple attachments of the same policy on a single attachment point. Only Service Manager executes multiple attachments of the same policy at the same attachment point.
- A detachment based on the policy name removes all attachments for that policy at the specified attachment point in a single command regardless of creation source. A detachment based on attachment type detaches all attachments at that attachment point regardless of creation source. Service Manager can delete only one attachment at a time through service deactivation.
- The **statistics** and **baseline** keywords for the merged policy attachment are computed as a logical OR for all attachments at the specified attachment point.
- If you delete an attachment:
 - The merged policy is recomputed with the remaining attachments of the same type that have the **merge** keyword set. The statistics for common classifier groups are preserved when replacing the existing merged attachment.
 - The **statistics** and **baseline** keywords for the merged policy attachment are recomputed to be a logical OR of all remaining attachments at the specified attachment point.

**Related
Documentation**

- [Policy Attachment Sequence at Login Through Service Manager on page 22](#)

Error Conditions for Merged Policies

Most errors, such as mismatched interface types while merging attachments, are caught during configuration. If merging fails, the attachment at the given interface is not modified.

You can modify component policies manually. Although you might want to do this for debugging purposes, we highly discourage you doing this because it can affect synchronization with the Service Manager application. You cannot manually attach a final merged policy to any interfaces. Instead, attach the set of component policies that constitute this merged policy. If you want to modify the final merged policy, use existing policy merging or component policy modification to achieve this.

**Related
Documentation**

- [Resolving Policy Merge Conflicts on page 19](#)

Parent Group Merge Algorithm

The parent group merge algorithm enables the system to merge policies that contain references to parent groups and create an internal parent group for each internal parent group in a component policy in the final merged policy. There is a one-to-one correspondence between an internal parent group in the merged policy and an internal parent group in a component policy.



NOTE: The naive parent group merging algorithm is not compatible with this parent group merge algorithm. If you have service definitions that used the naive parent group algorithm, you need to modify those service definitions to work with this algorithm.

- If there is no existing internal parent group with the same name in the merged policy, the system creates a corresponding internal parent group with the same name.
- If an internal parent group with the same name already exists, the system uses a name built by appending an internally generated sequence number to the name of the internal parent group in the component policy.
- If the length of the name exceeds the maximum length allowed, the policy merge fails.
- If a classifier group in a component policy refers to an internal parent group, the same classifier group in the merged policy corresponds to the internal parent group in the merged policy.
- If a classifier group in a component policy refers to an external parent group, the same classifier group in the merged policy refers to the same external parent group.
- If there is a conflict where two or more component policies contain the same classifier group referring to an internal parent group in a corresponding component policy or to an external parent group, then last one is used.

In the following example, component policies P1 and P2 create the merged policy mpl_88000001.

host1#show policy-list P1

Policy Table

IP Policy P1

Administrative state: enable

Reference count: 1

Classifier control list: *, precedence 100, parent-group Z
forward

Classifier control list: A, precedence 100, parent-group X

forward

Classifier control list: B, precedence 100, parent-group X

forward

Classifier control list: C, precedence 100, external parent-group EPG1

parameter foo

forward

Classifier control list: D, precedence 100, external parent-group EPG1 parameter
foo

forward

Parent group: X, parent-group Z

rate-limit-profile R1

Parent group: Z

rate-limit-profile R2

host1#show policy-list P2

Policy Table

```
IP Policy P2
  Administrative state: enable
  Reference count:      1
  Classifier control list: B, precedence 100, parent-group X
forward
  Classifier control list: C, precedence 100, parent-group Y
  forward
  Classifier control list: D, precedence 100, external parent-group EPG2 parameter
abcd
forward

  Parent group: X, parent-group Y
    rate-limit-profile R3
  Parent group: Y
    rate-limit-profile R4

host1#show policy-list mpl_88000001
```

Policy Table

```
IP Policy mpl_88000001
Administrative state: enable
Reference count:      1
  Classifier control list: *, precedence 100, parent-group Z
forward
  Classifier control list: A, precedence 100, parent-group X
forward
  Classifier control list: B, precedence 100, parent-group X_1
forward
  Classifier control list: C, precedence 100, parent-group Y
forward
  Classifier control list: D, precedence 100, external parent-group EPG2 parameter
abcd
forward

  Parent group: X, parent-group Z
    rate-limit-profile R1
  Parent group: Z
    rate-limit-profile R2
  Parent group: X_1, parent-group P2_Y
    rate-limit-profile R3
  Parent group: Y
    rate-limit-profile R4

Referenced by interfaces:
  ATM5/0.1 input policy, statistics enabled, virtual-router default

Referenced by profiles:
  None

Component policies:
  P1
  P2
```

Related Documentation

- External Parent Groups

Overlapping Classification for IP Input Policy

IP auxiliary input policy can be used with IP input policy to provide overlapping classification. Two policies, each with a set of independent rules and actions, run in

sequence so that each policy can independently produce a set of actions in sequence. A packet that matches both the input policies and auxiliary input policies is subject to both sets of policy actions.

E Series routers allow four input and two output policies per IP interface:

- One secure input policy
- Three nonsecure input policies
- One secure output policy
- One nonsecure output policy

Each classifier-group has a set of associated actions that is taken if it is the highest priority match. The system performs only one set of actions per policy attachment. By using an input and secondary-input policy, you can have overlapping classification with multiple policy actions on ingress. Overlapping classification on egress is not supported.

An additional policy attachment point enables overlapping classification within the input classification stage, between the input and secondary-input stages. There are five attachment points for IP policies that are executed in series:

- input
- secondary-input
- secure-input
- output
- secure-output

An explicit filter action, a forward action with a null next-interface, or a rate-limit action can cause an immediate packet discard at any stage. Other actions, such as marking and coloring can be done at each stage, with the last of each of these actions taking precedence over the others.

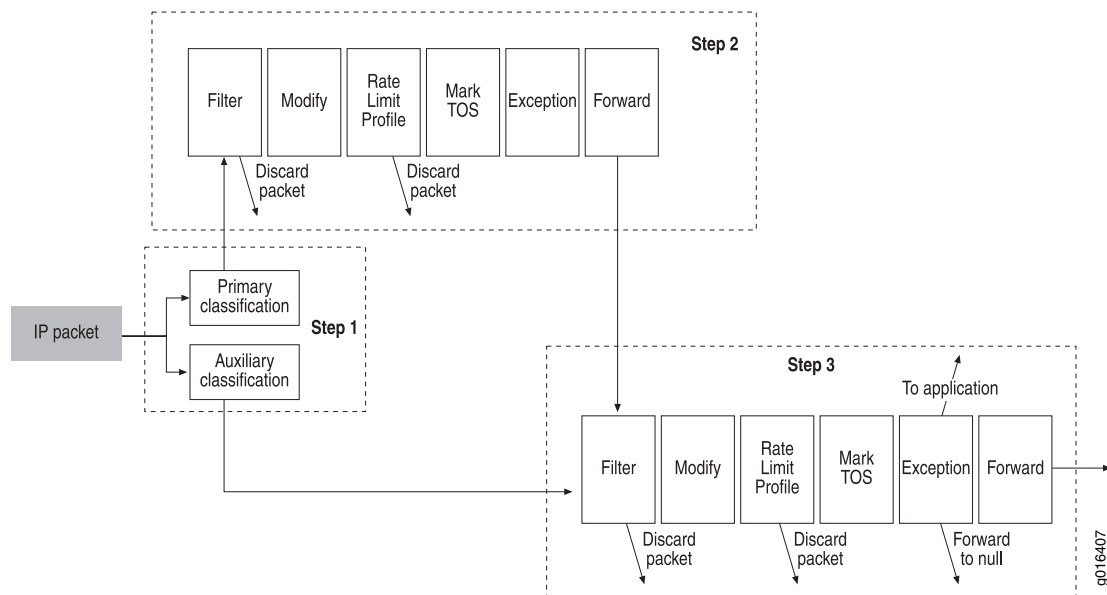
For example, unique policies can be attached at each stage, all of which mark the IP TOS field differently. The packet then exits the router with the TOS value that was set in the output policy stage. However, if TOS is also used as a classification (input) term for each of these policies, three different TOS values are presented to the classifier:

- Original TOS received
- TOS modified by the input policy
- TOS value modified by the secondary-input policy

[Figure 1 on page 28](#) shows the input policy stage after the addition of the auxiliary substage. It is divided into three steps:

1. Apply classification for both substages.
2. Perform policy actions (if any) for the primary attachment.
3. Perform policy actions (if any) for the auxiliary attachment.

Figure 1: Input Policy with Primary Stage and Auxiliary Substage



The order of policy action execution for each attachment is:

1. Filter
2. Modify (includes setting of color, traffic class, user packet class) and Log
3. Rate limit profile/color
4. Mark TOS
5. Exception
6. Forward

Starting Policy Processing

Input and auxiliary-input classification operations, specified by the details of each policy, are performed in parallel. Classifier inputs for both policies are determined concurrently using the initial values of the classification terms. Policy attachments within a stage cannot communicate between the input and auxiliary-input classification operations. For example, any changes made by the input attachment to traffic-class, color, TOS, or user packet class are not visible in the auxiliary-input policy classification. If this communication is needed, it can only be done between different policy stages, rather than within a single stage.

The results of the input policy actions are passed forward to the auxiliary-input policy action processing. This means that a color-aware rate limit profile action in the auxiliary substage recognizes any change in color caused by primary policy actions.

Processing the Classifier Result

The classifier result of the input policy attachment is processed and a set of actions is identified. When you configure filter, it is the first action taken and immediately discards

the packet. This is followed by any modification, such as mark or logging. If a rate limit profile is configured, the packet is dropped or colored. If the packet is not dropped, it is sent to the exception path (if configured). If the packet is not exceptioned, any configured forward action is saved in the packet for use later (unless overridden in Step 3). (See [Figure 1 on page 28.](#))

Some information generated by the action processing in Step 2 is forwarded to Step 3, where it may affect the action processing for the auxiliary-input attachment. This information can include color, exception information, and forwarding information. The color can affect a rate-limit in the auxiliary-input attachment. Step 3 acts on the exception and forwarding information, if it is not overridden by similar actions from the auxiliary-input attachment.

The transmit information (transmit conditional, transmit unconditional, transmit final) generated with hierarchical policies does not carry forward from input to auxiliary-input action processing.

Processing the Auxiliary-Input Policy Attachment

If the packet is not filtered or exceptioned in policy Step 2, the classifier result of the auxiliary policy attachment is processed and a set of actions identified. The packet can be filtered or exceptioned at this time. These operations, if configured, are performed regardless of whether a forward action was performed in Step 2. If the packet is not discarded, either by a filter action or a rate limit, it can be exceptioned (if configured). If the packet is not filtered, rate-limited, or exceptioned, any configured forward action is applied and overrides any forward action from Step 2. If no forward action is configured, any forward action from Step 2 applies.

Policy Actions

The set of actions in the following list specified by the input and auxiliary-input policy attachments are executed in the order: input, auxiliary-input.

- **Color packet action**—Explicitly sets the packet color. Each policy attachment can set the color and the final value persists. A rate limit profile action can also set the color, which overrides the value of the color packet action.
- **Mark action**—Each attachment can set the TIP TOS, TOS precedence, and DS fields. The cumulative result of all configured mark actions determines the resulting value of these fields.
- **Mirror action**—Executes in the order: secure input policy follows secondary input policy, secure output policy follows output policy. Mirror is the only supported action for secure policies.
- **Rate-limit profile action**—Can be specified by any nonsecure input policy attachment. This enables the application of multiple rate limits either within a policy stage or across policy stages. These rate limits run serially; if the rate limit imposed in the primary substage causes the packet to drop, the auxiliary rate limit does not run and the associated token buckets are not affected. If you configure more than a single rate limit per interface, it significantly impacts forwarding performance. Attaching two

policies with rate limit profiles in the same policy stage is equivalent to having two policies attached in the same order, but in separate stages.

- Traffic class action—If both the input and auxiliary-input attachments need this action, the value configured in the auxiliary policy overwrites that of the primary policy.
- User packet class action—Can be set twice per stage, with the second value overriding the first.
- The filter, next-hop, forward interface, and forward next-hop actions are mutually exclusive within a classifier group. However, two policies in series can result in conflicting actions, which are resolved using the following precedence rules:
 - The filter action has highest priority. A filter action in input or auxiliary-input policy always prevails.
 - The exception action takes precedence over forward actions.
 - If multiple exception actions are required by the policy attachments, the last one takes precedence.
 - If forward operations are required by both input and auxiliary-input policy attachments, the auxiliary-input forward action takes precedence.

Input policy attachments depend on the **local** keyword in classifier list entries. Using the **local false** keyword or using no local keyword (default) treats both local and non-local traffic equally and ignores the local true classifier list entries.

Secondary input policies affect both local and non-local traffic and are processed by policies attached with the **secondary-input** keyword. Secondary input policies are controlled by the **local** keyword in the classifier list entries as follows:

- **local true** keyword only affects local traffic
- **local false** keyword only affects non-local traffic
- no **local** keyword (default) affects both local and non-local traffic

In [Table 5 on page 30](#), the filter action for the input policy takes precedence over the others so that if a filter action is configured for either policy, the packet is filtered. If neither policy has a filter action, but both policies specify a forward action, the action specified by the auxiliary policy takes precedence. If only one policy specifies a forwarding action, that action is executed. The next-hop rule is inoperative for auxiliary-input policies, just as it is for secondary input policies. This policy rule has been superseded (but not replaced) by the forward next-hop rule, which is operative for auxiliary-input policies.

Table 5: Input Action and Secondary Input Actions

Input Action	Secondary Input Action					
	None	Exception	Filter	Next-hop	Forward Interface	Forward Next-hop

Table 5: Input Action and Secondary Input Actions (*continued*)

Input Action	Secondary Input Action					
None	None	Exception Auxiliary	Filter	None	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Exception	Exception Primary	Exception Auxiliary	Filter	Exception	Exception Primary	Exception Primary
Filter	Filter	Filter	Filter	Filter	Filter	Filter
Next-hop	Next-hop Primary	Exception Auxiliary	Filter	Next-hop Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Fwd Interface	Forward Interface Primary	Exception Auxiliary	Filter	Forward Interface Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Fwd next-hop	Forward Next-hop Primary	Exception Auxiliary	Filter	Forward Next-hop Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary

Related Documentation • [Policy Attachment Rules for Merged Policies on page 23](#)

PART 2

Configuration

- [Configuration Tasks for Managing Classifier Groups and Policy Rules on page 35](#)
- [Configuration Tasks for Merging Policies on page 47](#)
- [Examples on page 59](#)

CHAPTER 3

Configuration Tasks for Managing Classifier Groups and Policy Rules

- [Using Policy Rules to Provide Routing Solutions on page 35](#)
- [Configuring Policies to Provide Network Security on page 36](#)
- [Creating an Exception Rule within a Policy Classifier Group on page 37](#)
- [Defining Policy Rules for Forwarding on page 38](#)
- [Forwarding Based on Next-Hop Addresses for Input IPv4 and IPv6 Policies on page 39](#)
- [Assigning Values to the ATM CLP Bit on page 41](#)
- [Enabling ATM Cell Mode on page 42](#)
- [Enabling IP Options Filtering on page 42](#)
- [Creating Multiple Forwarding Solutions with IP Policy Lists on page 43](#)
- [Creating a Classifier Group for a Policy List on page 44](#)

Using Policy Rules to Provide Routing Solutions

The next-interface, next-hop, filter, and forward rules provide routing solutions for traffic matching a classifier. A classifier can have only one action that provides a routing solution.

If you configure two routing solution rules, such as filter and forward, in the same classifier group, the router displays a warning message, and the rule configured last replaces the previous rule.

For IP policy lists, policy rules are available to enable you to make a forwarding decision that includes the next interface and next hop:

- **Forward next interface**—Causes an interface to forward all packets that satisfy the classification associated with that rule to the next interface specified
- **Forward next hop**—Causes an interface to forward all packets that satisfy the classification associated with that rule to the next-hop address specified

For example, you can route packets arriving at IP interface ATM 0/0.0 so that they are handled as indicated:

- Packets from source 1.1.1.1 are forwarded out of interface ATM 0/0.1.

- Packets from source 2.2.2.2 are forwarded out of interface ATM 2/1.1.
- All other packets are dropped.

To configure this routing policy, issue the following commands:

```
host1(config)#ip classifier-list clacA ip host 1.1.1.1 any
host1(config)#ip classifier-list clacB ip host 2.2.2.2 any
host1(config)#ip policy-list IpPolicy100
host1(config-policy-list)#classifier-group clacA
host1(config-policy-list-classifier-group)#forward interface atm 0/0.1
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacB
host1(config-policy-list-classifier-group)#forward interface atm 2/1.1
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input IpPolicy100 statistics enabled
```

**Related
Documentation**

- [Classifier Groups and Policy Rules Overview on page 3](#)
- classifier-group
- ip classifier-list
- ip policy-list

Configuring Policies to Provide Network Security

You can configure policy management to provide a level of network security by using policy rules that selectively forward or filter packet flows:

- Forward—Causes the packet flows that satisfy the classification associated with the rule to be routed by the virtual router
- Filter—Causes the interface to drop all packets of the packet flow that satisfy the classification associated with the rule

To stop a denial-of-service attack, you can use a policy with a filter rule. You need to construct the classifier list associated with the filter rule so that it isolates the attacker's traffic into a flow. To determine the criteria for this classifier list, you need to analyze the traffic received on an interface. Monitoring Policy Management Overview describes how to capture packets into a log.

For example, you can route packets entering an IP interface (ATM 0/0.0) so that they are handled as indicated:

- Packets from source 1.1.1.1 are routed.
- TCP packets from source 2.2.2.2 with the IP fragmentation offset set to one are dropped.
- All other TCP packets are routed.
- All other packets are dropped.

To configure this policy, issue the following commands:

```
host1(config)#ip classifier-list clacA ip host 1.1.1.1 any
host1(config)#ip classifier-list clacB tcp host 2.2.2.2 any ip-frag-offset eq 1
host1(config)#ip classifier-list clacC tcp any any
host1(config)#ip policy-list IpPolicy100
host1(config-policy-list)#classifier-group clacA
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacB
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacC
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input IpPolicy100 statistics enabled
```

Related Documentation

- [Classifier Groups and Policy Rules Overview on page 3](#)
- classifier-group
- ip classifier-list
- ip policy-list

Creating an Exception Rule within a Policy Classifier Group

To create the exception rule within an IP policy classifier group to specify the client application for the destination of packets rather than forwarding them by the forwarding controller (FC), use the **exception http-redirect** command. Doing this enables the application to then perform an application-dependent action on the content of the packet. The exception rule applies to input and secondary-input policies.

The guidelines for creating exception rules within an IPv6 policy classifier group are the same as those for creating exception rules within an IPv4 policy classifier group.



NOTE: The **exception http-redirect** command is not supported for the ES2 10G Uplink LM.

An exception rule in the input policy only takes effect if neither the input policy nor the secondary policy drops the packet. Packets dropped by input or secondary policies are not excepted to the SRP module. HTTP redirect is the only application that is available as a destination of the **exception** rule.

Because classifier groups can contain multiple actions, the following list describes how each rule interacts with the exception rule:

- **color**—Packets are colored and the exception rule is applied.
- **filter**—Packets are filtered and the exception rule is not applied. When the filter rule is present, other rules are not applied.
- **forward**—Forward rule is ignored and the exception rule is applied to packets.
- **log**—Packets are logged and the exception rule is applied.
- **mark**—Packets are marked and the exception rule is applied.
- **next-hop**—Next-hop rule is ignored and the exception rule is applied to packets.
- **next-interface**—Next-interface rule is ignored and the exception rule is applied to packets.
- **rate-limit-profile**—Rate limit is applied and the exception rule is applied to packets.
- **traffic-class**—Traffic class is set and the exception rule is applied to packets.
- **user-packet-class**—User packet class is set and the exception rule is applied to packets.
- **exception**—Exception rule is applied to packets.

**Related
Documentation**

- [Classifier Groups and Policy Rules Overview on page 3](#)
- `exception http-redirect`

Defining Policy Rules for Forwarding

The **forward next-hop** command defines a rule that creates the forwarding solution for packets matching the current CLACL. The **forward** command can be used while the policy list is referenced by interfaces. The **suspend** version suspends the forward rule within the classifier group.

For IPv4 and IPv6 policy lists:

- You can use the **forward interface** command to specify multiple interfaces for IPv4 policies and the **forward next-hop** command to specify next-hop IPv4 or IPv6 addresses as possible forwarding solutions for IPv4 or IPv6 policies. If you define multiple forwarding solutions for a single CLACL, use the **order** keyword to specify the order in which the router chooses the solutions. The router uses the first reachable solution in the list, starting with the solution with the lowest order value. The default order value is 100.



NOTE: The **forward interface** and **forward next-hop** commands replace the **next-interface** and **next-hop** commands.

The switch route processor (SRP) module Fast Ethernet port cannot be the destination of the **forward next-hop** and **forward next-interface** commands.

- If you specify a next-hop address as the forwarding solution, you can specify that the default route is not used as a routing solution for the next-hop address when selecting a reachable forward rule entry.
- IP interfaces referenced with this command can be tracked if they move. Policies attached to an interface also move if the interface moves. However, statistics are not maintained across the move.
- You can no longer use an interface specifier of **tunnel:mpls** with the **forward interface** command, because that usage requires IP interfaces on top of RSVP-TE tunnels. Such interfaces are no longer present in the redesigned MPLS architecture. However, you can configure a static route for an address that is not otherwise used to point to a tunnel, and then use the **forward next-hop** command in the policy:

```
host1(config)#ip route 10.10.10.10/32 tunnel mpls:foo
host1(config)#ip policy-list bar
host1(config-policy-list-classifier-group)#forward next-hop 10.10.10.10
```

Related Documentation

- [Classifier Groups and Policy Rules Overview on page 3](#)
- [forward](#)
- [forward interface](#)
- [forward next-hop](#)

Forwarding Based on Next-Hop Addresses for Input IPv4 and IPv6 Policies

You can define policies for incoming IPv4 and IPv6 traffic and apply the policy lists to the ingress of an interface to enable packet forwarding and routing operations to be performed based on the configured rules and actions. The forward rules that you define in classifier groups contained in a policy list define the forwarding mechanism for IPv4 and IPv6 packets that match the specified classifier access list (CLACL). You can use the **forward interface** command to specify multiple IPv4 interfaces for IPv4 policy lists and the **forward next-hop** command to specify next-hop addresses as possible forwarding solutions for IPv4 and IPv6 policy lists.

The next-hop and next-interface actions override the routing table lookup. In an environment in which Gigabit Ethernet uplink modules are connected to broadcast networks, you can use the next-hop actions for routing and forwarding of traffic. For IPv6 traffic, you cannot configure a forward rule to transmit packets that match a specific CLACL to a specific interface or multiple interfaces. However, you can configure a rule to forward packets that match a CLACL to multiple interfaces for IPv4 traffic.

You can specify multiple next-hop addresses or actions in a single forwarding policy rule. In such a case, packets are forwarded to the first available next-hop address that contains a route in the routing table. You can use the **order** keyword with the **forward next-hop** command in Classifier Group Configuration mode to specify the order of the group of forwarding solutions within a single forward rule.

To enable a forwarding solution to function by overriding the routing table lookup, you can configure policies with one or multiple next-hop addresses. Dynamic selection of the next-hop address is available. If a next-hop with the lowest order becomes reachable or is added freshly to a forward rule, the currently processed element is disregarded and the new next-hop entry is considered. If multiple next-hop addresses specified in the policy list have the same order, the selection is done based on the reachability and the first configured entry. You can specify a maximum of 20 forwarding solutions for a classifier. This limit encompasses the forward next-hop and the next-interface actions.

You can configure multiple next-hop elements in a forward rule for only the same virtual router. You cannot configure multiple forward next-hop rules in a policy that spans across different VRs. If only next-hop elements exist and you do not use the **virtual-router** option with the **forward next-hop** command, then the policy assumes the virtual router context of the CLI, making the policy specific to that VR. The policy can be attached only to interfaces that belong to that VR. You can use the **virtual-router** keyword with the **forward next-hop** command to specify a VR other than the default VR to enable the configuration of next-hop elements for that VR.

When a next-hop address is reachable, only if it has an entry in the routing table, this next-hop can be a default route in certain scenarios. In such cases, you can include the **ignore-default-route** keyword with the **forward next-hop** command to cause the default route to be not considered for the next-hop determination.

If next-hop selection changes dynamically, because of changes in the order of the action or changes in the reachability state of the next-hop, the statistics associated with the next-hop action are preserved, if collection of statistical details is enabled in the policy list. The statistical information is used per classifier rule that has a list of multiple next-hop actions.

Keep the following guidelines in mind while configuring forwarding rules based on next-hop addresses for input IPv6 policies:

- You can configure the rule to forward all packets that match a CLACL to a particular next-hop address only for input IPv6 policies on routers with ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs (policies applied to ingress interfaces) or IPv6 policies on ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs that function as access line modules (line modules with policies that receive traffic from low-speed circuits and route it to uplink modules).
- You cannot configure next-hop addresses as forwarding rules for IPv6 policies when the ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs are core-facing, uplink modules. However, when the ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs operate as access modules for forwarding rules for IPv6 policies, you can configure the core-facing modules as ES2 4G LMs, ES2 10G LMs, ES2 10G Uplink LMs, or ES2 10G ADV LMs.
- The performance of the policy manager application might be slightly impacted if you configure a significant number of IPv6 policies with forward rules and the reachability states of the configured next-hop addresses transition frequently.
- Forwarding of traffic based on next-hop addresses in input IPv6 policy lists is available only for ingress IPv6 interfaces that are configured over Ethernet or MPLS interfaces.
- You cannot configure forward rules based on next-hop addresses in policy lists for IPv6 interfaces over GRE tunnels.
- You can configure only indirect next-hop addresses while configuring forwarding rules based on next-hop addresses for input IPv6 policies.
- You cannot configure link-local, loopback, or multicast addresses for forwarding of traffic based on next-hop addresses in a classifier group in an IPv6 policy list. If you attempt to configure these types of addresses as next-hop addresses for forwarding of traffic using the **forward next-hop** command for IPv6 policy lists, an error message is displayed.

Related Documentation

- [Defining Policy Rules for Forwarding on page 38](#)
- [Creating Multiple Forwarding Solutions with IP Policy Lists on page 43](#)
- Creating Policy Lists for IP
- Creating Policy Lists for IPv6
- forward
- forward next-hop

Assigning Values to the ATM CLP Bit

The **mark-clp** command assigns a value of 0 or 1 to the ATM CLP bit for packets conforming to the current classifier control list.

Modules on E Series routers support classifying and marking of the ATM CLP bit according to the following rules:

- Modules on E120 and E320 routers support classifying of the ATM CLP bit only for frame-based interfaces (ATM Adaptation Layer 5 [AAL5] encapsulation), but not for

individual ATM cells (ATM Adaptation Layer 0 [AAL0] encapsulation). In this case, if the CLP bit in any cell in the frame has a value of 1, the router treats the reassembled AAL5 frame as if it also had a CLP value of 1.

- Modules on E120 and E320 routers support marking of the ATM CLP bit on frame-based interfaces. In this case, every cell of the segmented frame leaves the router with the same CLP value.
- Modules on ERX7xx models, ERX14xx models, and the ERX310 Broadband Services Router support classifying and marking of the ATM CLP bit for individual ATM cells (AAL0 encapsulation), but not for frame-based interfaces (AAL5 encapsulation).

Related Documentation

- [mark-clp](#)

Enabling ATM Cell Mode

When you configure a rate limit profile to account for ATM cell tax, the forwarding code calculates this information to determine the size of a frame instead of using only the frame size.

- Issue the **atm-cell-mode** command to account for the ATM cell tax in statistics and rate calculations:

```
host1(config-policy-list)#atm-cell-mode
```

Use the **show rate-limit-profile** command to display the state of the mode.

Related Documentation

- [Monitoring Policy Management Overview](#)
- [atm-cell-mode](#)
- [show rate-limit-profile](#)

Enabling IP Options Filtering

You can filter packets with IP options on an interface:

- Issue the **ip filter-options all** command.

```
host1(config-if)#ip filter-options all
```

When a packet arrives on an interface, the router checks to see if the packet contains IP options. If it does and if IP options filtering is enabled, that packet is dropped. IP options filtering is disabled by default.

Related Documentation

- [Classifier Groups and Policy Rules Overview on page 3](#)
- [ip filter-options all](#)

Creating Multiple Forwarding Solutions with IP Policy Lists

By default, the router uses a single route table lookup to determine the forwarding solution for packets. For IP policy lists only, the **forward** command enables you to configure one or more unique forwarding solutions (interfaces or next-hop addresses) that override the route table lookup. By creating a group of forwarding solutions, you can ensure that there is a reachable solution for the packets.

You can use the **order** keyword to specify the order of the group of forwarding solutions within a single forward rule. If no order value is specified, then the default order of 100 is assigned to a solution. The router evaluates the forwarding solutions in the group, starting at the solution with the lowest order value, and then uses the first reachable solution. To be considered a reachable solution, a solution must be a reachable interface or a next-hop address that has a route in the routing table. If no solutions are reachable, the traffic is dropped.

The following guidelines apply when you create a group of forwarding solutions in an IP policy list:

- You can specify a maximum of 20 forwarding solutions for a classifier.
- The interface and next-hop elements of a forwarding solution must exist within a single virtual router:
 - Next-interface elements are associated with the virtual router where that interface exists.
 - You can include an optional parameter to specify the virtual router when you define next-hop elements.
 - If only next-hop elements exist and you do not use the virtual router option, then the policy assumes the virtual router context of the command-line interface (CLI), making the policy specific to that VR. The policy can be attached only to interfaces that belong to that VR. However, the policy can still be displayed and modified from any VR. The output of the **show configuration** command displays the policy in the section of output related to that VR rather than in the section for the default VR. This behavior ensures that when you use that output for a configuration script, the policy is specific to the correct VR, and the original configuration is re-created.
- If you specify both an interface element and a next-hop address element, then they both must be reachable to be used. Also, the interface must be the correct interface for the next-hop address.
- If you specify a next-hop address, then you can optionally specify that the default route be ignored.
- If you delete the target (interface or next-hop address) referenced in a rule, that solution is replaced by the null interface but retains the same order number in the policy list. The null interface is always considered unreachable.

- When a forwarding solution with a lower order value than the currently active solution becomes reachable, the router switches to the lower-ordered solution.
- If two rules that have the same order value are reachable, then the rule that was created first is used.



NOTE: The `forward interface` and `forward next-hop` commands are replacing the `next-interface` and `next-hop` commands, which do not support multiple forwarding solutions in a single forward rule.

In the following sample classifier group of a policy list, the forwarding solution of ATM interface 0/0.1 has the lowest order value in the group, and would therefore be selected as the solution for the policy list. However, if this interface is not reachable, the router then attempts to use the solution with the next higher order; which would be ATM interface 12/0.1. If none of the solutions in the group is reachable, the traffic is dropped.

```
host1(config-policy-list)#classifier-group westfordClac1 precedence 200
host1(config-policy-list-classifier-group)#forward interface atm 0/0.1 order 10
host1(config-policy-list-classifier-group)#forward interface atm 12/0.1 order 50
host1(config-policy-list-classifier-group)#forward interface atm 3/0.25 order 300
```



NOTE: You can use the `suspend` version of the command to suspend an individual entry in a group of forwarding solutions. The forward rule remains active as long as there is a reachable or active entry in the group of forwarding solutions. If you suspend all entries in the group, the status of the forward rule is changed to suspended.

Related Documentation

- [Creating or Modifying Classifier Control Lists for IP Policy Lists](#)
- [Creating Policy Lists for IP](#)

Creating a Classifier Group for a Policy List

To create a classifier group for a policy list and assigns precedence to the specific CLACL that is referenced in the group:

1. Create a classifier group.

```
host1(config-policy-list)#classifier-group C1 parent-group IPG1
```

2. Assign a precedence to the CLACL.

```
host1(config-policy-list)#classifier-group westfordClac1 precedence 150
```

3. Create a hierarchical policy parameter list.

```
host1(config)#policy-parameter A hierarchical
host1(config)#parent-group EPG1
host1(config-parent-group)#exit
host1(config)#ip policy-list POL
```



```

host1(config-policy-list)#classifier-group C1 external parent-group EPG1 parameter
A
host1(config-policy-list)#exit

```

The **no** version removes the classifier group and its rules from a policy list. The **precedence** keyword specifies the order in which a classifier group is evaluated compared to other classifier groups. Classifier groups are evaluated from lowest to highest precedence value (for example, a classifier group with a precedence of 1 is used before a classifier group with a precedence of 2). Classifier groups with equal precedence are evaluated in the order of creation, with the group created first having precedence. A default value of 100 is used if no precedence is specified.

The **parent-group** keyword creates a parent group in a rate-limit hierarchy for IP, IPv6, L2TP, and MPLS. The **external parent-group** keyword creates an external parent group in a rate-limit hierarchy for IP, IPv6, L2TP, and MPLS. All packets matching the classifier are sent to the parent group for further processing, except for packets dropped by the classifier using the filter rule.

More than one classifier group can have the same parent group, which enables you to create hierarchies.



NOTE: Empty classifier groups have no effect on the router's classification of packets and are ignored by the router. You might inadvertently create empty classifier groups in a policy if you use both the newer CLI style and the older CLI style, which used the Policy List Configuration mode version of the classifier list commands.

Related Documentation

- [Classifier Groups and Policy Rules Overview on page 3](#)
- Creating Rate-Limit Profiles
- Monitoring Policy Management Overview
- aggregation-node
- classifier-group
- ip policy-parameter hierarchical
- ip policy-parameter reference-rate
- ipv6 policy-parameter hierarchical
- ipv6 policy-parameter reference-rate
- l2tp policy-parameter hierarchical
- l2tp policy-parameter reference-rate
- mpls policy-parameter hierarchical
- mpls policy-parameter reference-rate
- next-parent

- parent-group
- policy-parameter hierarchical

Configuration Tasks for Merging Policies

- [Merging Policies on page 47](#)

Merging Policies

In the following example IP policy p1 and IP policy p2 are attached at interface atm5/0.1 as input attachments. Subsequently, policy p3 is attached at the same point. Then policies p1 and p2 are attached as output at atm 5/0.2.

1. Create IP policy p1.

```
host1(config)#ip classifier-list C1 tcp host 1.1.1.1 any eq 80
host1(config)#ip classifier-list C2 icmp any any 8 0
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-policy-classifier-group)#forward next-hop 10.1.1.1
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group C2 precedence 10
host1(config-policy-classifier-group)#filter
host1(config-policy-classifier-group)#exit
```

2. Create IP policy p2.

```
host1(config)#ip classifier-list C1 tcp host 1.1.1.1 any eq 80
host1(config)#ip classifier-list C3 ip any host 2.2.2.2
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-policy-classifier-group)#forward next-hop 20.1.1.1
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group C3 precedence 10
host1(config-policy-classifier-group)#filter
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group * precedence 1000
host1(config-policy-classifier-group)#forward
host1(config-policy-classifier-group)#exit
```

3. Attach IP policy p1 as input at interface atm5/0.1.

```
host1(config)#Interface atm 5/0.1
host1(config-subif)#ip policy input p1 statistics enable merge
host1(config-subif)#exit
```

4. Attach IP policy p2 as input at interface atm 5/0.1. A merged policy is created.

```
host1(config)#Interface atm 5/0.1
```

```
host1(config-subif)#ip policy input p2 statistics enable merge
host1(config-subif)#exit
```

5. Display the policy lists.

```
host1#show policy-list
```

```

                                Policy Table
                                -----
IP Policy p1
  Administrative state: enable
  Reference count:      1
  Classifier control list: C2, precedence 10
    filter
  Classifier control list: C1, precedence 90
    forward
    Virtual-router: default
    List:
      next-hop 10.1.1.1, order 100, rule 2 (active)

  Referenced by interfaces:
    None

  Referenced by profiles:
    None

  Referenced by merge policies:
    mpl_5

IP Policy p2
  Administrative state: enable
  Reference count:      1
  Classifier control list: C3, precedence 10
    filter
  Classifier control list: C1, precedence 90
    forward
    Virtual-router: default
    List:
      next-hop 20.1.1.1, order 100, rule 3 (active)
  Classifier control list: *, precedence 1000
    forward

  Referenced by interfaces:
    None

  Referenced by profiles:
    None

  Referenced by merge policies:
    mpl_5

IP Policy mpl_5
  Administrative state: enable
  Reference count:      1
  Classifier control list: C2, precedence 10
    filter
  Classifier control list: C3, precedence 10
    filter
  Classifier control list: C1, precedence 90
    forward
    Virtual-router: default
    List:
      next-hop 10.1.1.1, order 100, rule 2 (active)
      next-hop 20.1.1.1, order 100, rule 3 (reachable)
```

```

Classifier control list: *, precedence 1000
  forward

Referenced by interfaces:
  ATM5/0.1 input policy, statistics enabled, virtual-router default

Referenced by profiles:
  None

Component policies:
  p1
  p2

```

6. Show configuration.

```

host1#show conf

! Configuration script being generated on TUE APR 26 2005 17:33:01 UTC
! Juniper Edge Routing Switch ERX1440
! Version: 9.9.9 development-4.0 (April 4, 2005 15:39)
! Copyright (c) 1999-2005 Juniper Networks, Inc. All rights reserved.
!
! Commands displayed are limited to those available at privilege level 15
!
...
interface atm 5/0.1
  ip policy input p1 statistics enabled merge
  ip policy input p2 statistics enabled merge
exit
...
ip policy-list p1
  classifier-group C2 precedence 10
  filter
  classifier-group C1 precedence 90
  forward next-hop 10.1.1.1
!
ip policy-list p2
  classifier-group C3 precedence 10
  filter
  classifier-group C1 precedence 90
  forward next-hop 20.1.1.1
  classifier-group * precedence 1000
  forward
!
...
! End of generated configuration script.

```

7. Display interface statistics.

```

host1#show ip interface atm 5/0.1

ATM5/0.1 line protocol Atm1483 is up, ip is up
Network Protocols: IP
Internet address is 99.99.99.2/255.255.255.0
Broadcast address is 255.255.255.255
Operational MTU = 9180 Administrative MTU = 0
Operational speed = 155520000 Administrative speed = 0
Discontinuity Time = 721112
Router advertisement = disabled
Proxy Arp = disabled
Network Address Translation is disabled
TCP MSS Adjustment = disabled

```

```

Administrative debounce-time = disabled
Operational debounce-time    = disabled
Access routing = disabled
Multipath mode = hashed
Auto Configure = disabled
Auto Detect = disabled
Inactivity Timer = disabled

In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy input mpl_5
  classifier-group C2 entry 1
    0 packets, 0 bytes
    filter
  classifier-group C3 entry 1
    0 packets, 0 bytes
    filter
  classifier-group C1 entry 1
    0 packets, 0 bytes
    forward
  classifier-group *
    0 packets, 0 bytes
    forward
queue 0: traffic class best-effort, bound to ip ATM5/0.1
  Queue length 0 bytes
  Forwarded packets 0, bytes 0
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0

```

8. Attach IP policy p1 at atm 5/0.2 as output.

```

host1(config)#interface atm 5/0.2
host1(config-subif)#ip policy output p1 statistics enable merge
host1(config-subif)#exit

```

9. Attach IP policy p2 at atm 5/0.2 as output. Merge policy mpl_5 is now attached.

```

host1(config)#interface atm 5/0.2
host1(config-subif)#ip policy output p2 merge
host1(config-subif)#exit

```

10. Display policies to verify that mpl_5 is created.

```
host1#show policy-list
```

Policy Table

```

IP Policy p1
Administrative state: enable
Reference count:      1

```

```

Classifier control list: C2, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
    Virtual-router: default
    List:
      next-hop 10.1.1.1, order 100, rule 2 (active)

Referenced by interfaces:
  None

Referenced by profiles:
  None

Referenced by merge policies:
  mpl_5

IP Policy p2
Administrative state: enable
Reference count:      1
Classifier control list: C3, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
    Virtual-router: default
    List:
      next-hop 20.1.1.1, order 100, rule 3 (active)
Classifier control list: *, precedence 1000
  forward

Referenced by interfaces:
  None

Referenced by profiles:
  None

Referenced by merge policies:
  mpl_5

IP Policy mpl_5
Administrative state: enable
Reference count:      2
Classifier control list: C2, precedence 10
  filter
Classifier control list: C3, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
    Virtual-router: default
    List:
      next-hop 10.1.1.1, order 100, rule 2 (active)
      next-hop 20.1.1.1, order 100, rule 3 (reachable)
Classifier control list: *, precedence 1000
  forward

Referenced by interfaces:
  ATM5/0.1 input policy, statistics enabled, virtual-router default
  ATM5/0.2 output policy, statistics enabled, virtual-router default

Referenced by profiles:
  None

Component policies:
  p1
  p2

```

11. Create and attach IP policy p3 at atm 5/0.1. A new merge policy mpl_7 is created, which is a combination of p1, p2, and p3. The previous merge policy attachment is removed.

```

host1(config)#ip classifier-list C4 udp host 1.1.1.1 any eq 900
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C4 precedence 900
host1(config-policy-classifier-group)#color red
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group C1 precedence 80
host1(config-policy-classifier-group)#color yellow
host1(config-policy-classifier-group)#exit
host1(config-policy)#exit
host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input p3 statistics enable merge
host1(config-subif)#exit

```

12. Display policies to verify that mpl_5 and mpl_7 have been created.

```
host1#show policy-list
```

Policy Table

IP Policy p1

```

Administrative state: enable
Reference count:      2
Classifier control list: C2, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
  Virtual-router: default
  List:
    next-hop 10.1.1.1, order 100, rule 2 (active)

Referenced by interfaces:
  None

Referenced by profiles:
  None

Referenced by merge policies:
  mpl_5
  mpl_7

```

IP Policy p2

```

Administrative state: enable
Reference count:      2
Classifier control list: C3, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
  Virtual-router: default
  List:
    next-hop 20.1.1.1, order 100, rule 3 (active)
Classifier control list: *, precedence 1000
  forward

Referenced by interfaces:
  None

Referenced by profiles:
  None

```


Referenced by merge policies:

mpl_5
mpl_7

IP Policy p3

Administrative state: enable
Reference count: 1
Classifier control list: C1, precedence 80
color yellow
Classifier control list: C4, precedence 900
color red

Referenced by interfaces:

None

Referenced by profiles:

None

Referenced by merge policies:

mpl_7

IP Policy mpl_5

Administrative state: enable
Reference count: 1
Classifier control list: C2, precedence 10
filter
Classifier control list: C3, precedence 10
filter
Classifier control list: C1, precedence 90
forward
Virtual-router: default
List:
next-hop 10.1.1.1, order 100, rule 2 (active)
next-hop 20.1.1.1, order 100, rule 3 (reachable)
Classifier control list: *, precedence 1000
forward

Referenced by interfaces:

ATM5/0.2 output policy, statistics enabled, virtual-router default

Referenced by profiles:

None

Component policies:

p1
p2

IP Policy mpl_7

Administrative state: enable
Reference count: 1
Classifier control list: C2, precedence 10
filter
Classifier control list: C3, precedence 10
filter
Classifier control list: C1, precedence 80
forward
Virtual-router: default
List:
next-hop 10.1.1.1, order 100, rule 2 (active)
next-hop 20.1.1.1, order 100, rule 3 (reachable)
color yellow
Classifier control list: C4, precedence 900
color red
Classifier control list: *, precedence 1000
forward

Referenced by interfaces:

ATM5/0.1 input policy, statistics enabled, virtual-router default

Referenced by profiles:

None

Component policies:

p1

p2

p3

13. Detach p2 from atm 5/0.1. A new merge policy mpl_8 is created, which is a combination of p1 and p3. The previous merge policy mpl_7 is detached and, because this policy has no attachments, it is deleted.

```
host1(config)#interface atm 5/0.1
host1(config-subif)#no ip policy input p2
host1(config-subif)#exit
```

14. Display policies to verify that the mpl_7 is removed and the new merge policy mpl_8 is created.

```
host1#show policy-list
```

Policy Table

IP Policy p1

Administrative state: enable

Reference count: 2

Classifier control list: C2, precedence 10
filter

Classifier control list: C1, precedence 90
forward

Virtual-router: default

List:

next-hop 10.1.1.1, order 100, rule 2 (active)

Referenced by interfaces:

None

Referenced by profiles:

None

Referenced by merge policies:

mpl_5

mpl_8

IP Policy p2

Administrative state: enable

Reference count: 1

Classifier control list: C3, precedence 10
filter

Classifier control list: C1, precedence 90
forward

Virtual-router: default

List:

next-hop 20.1.1.1, order 100, rule 3 (active)

Classifier control list: *, precedence 1000
forward

Referenced by interfaces:

None

Referenced by profiles:

None

Referenced by merge policies:
mpl_5

IP Policy p3
Administrative state: enable
Reference count: 1
Classifier control list: C1, precedence 80
color yellow
Classifier control list: C4, precedence 900
color red

Referenced by interfaces:
None

Referenced by profiles:
None

Referenced by merge policies:
mpl_8

IP Policy mpl_5
Administrative state: enable
Reference count: 1
Classifier control list: C2, precedence 10
filter
Classifier control list: C3, precedence 10
filter
Classifier control list: C1, precedence 90
forward
Virtual-router: default
List:
next-hop 10.1.1.1, order 100, rule 2 (active)
next-hop 20.1.1.1, order 100, rule 3 (reachable)
Classifier control list: *, precedence 1000
forward

Referenced by interfaces:
ATM5/0.2 output policy, statistics enabled, virtual-router default

Referenced by profiles:
None

Component policies:
p1
p2

IP Policy mpl_8
Administrative state: enable
Reference count: 1
Classifier control list: C2, precedence 10
filter
Classifier control list: C1, precedence 80
forward
Virtual-router: default
List:
next-hop 10.1.1.1, order 100, rule 2 (active)
next-hop 20.1.1.1, order 100, rule 3 (reachable)
color yellow
Classifier control list: C4, precedence 900
color red

Referenced by interfaces:
ATM5/0.1 input policy, statistics enabled, virtual-router default

Referenced by profiles:
None

Component policies:

p1
p3

15. Detach p1 from atm 5/0.1. Merge policy mpl_8 is detached and deleted, and only p3 is attached to this interface.

```
host1(config)#interface atm 5/0.1
host1(config-subif)#no ip policy input p1
host1(config-subif)#exit
```

16. Display policies to verify that p3 is attached to atm 5/0.1 and mpl_8 is removed.

```
host1#show policy-list
```

Policy Table

IP Policy p1

```
Administrative state: enable
Reference count:      1
Classifier control list: C2, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
  Virtual-router: default
  List:
    next-hop 10.1.1.1, order 100, rule 2 (active)

Referenced by interfaces:
  None

Referenced by profiles:
  None

Referenced by merge policies:
  mpl_5
```

IP Policy p2

```
Administrative state: enable
Reference count:      1
Classifier control list: C3, precedence 10
  filter
Classifier control list: C1, precedence 90
  forward
  Virtual-router: default
  List:
    next-hop 20.1.1.1, order 100, rule 3 (active)
Classifier control list: *, precedence 1000
  forward

Referenced by interfaces:
  None

Referenced by profiles:
  None

Referenced by merge policies:
  mpl_5
```

IP Policy p3

```
Administrative state: enable
Reference count:      1
Classifier control list: C1, precedence 80
  color yellow
```

```
Classifier control list: C4, precedence 900
color red
```

```
Referenced by interfaces:
```

```
ATM5/0.1 input policy, statistics disabled, virtual-router default
```

```
Referenced by profiles:
```

```
None
```

```
Referenced by merge policies:
```

```
None
```

```
IP Policy mpl_5
```

```
Administrative state: enable
```

```
Reference count: 1
```

```
Classifier control list: C2, precedence 10
filter
```

```
Classifier control list: C3, precedence 10
filter
```

```
Classifier control list: C1, precedence 90
forward
```

```
Virtual-router: default
```

```
List:
```

```
next-hop 10.1.1.1, order 100, rule 2 (active)
```

```
next-hop 20.1.1.1, order 100, rule 3 (reachable)
```

```
Classifier control list: *, precedence 1000
forward
```

```
Referenced by interfaces:
```

```
ATM5/0.2 output policy, statistics enabled, virtual-router default
```

```
Referenced by profiles:
```

```
None
```

```
Component policies:
```

```
p1
```

```
p2
```

17. Detach p3 from atm 5/0.1.

```
host1(config)#interface atm 5/0.1
host1(config-subif)#no ip policy input p3
host1(config-subif)#exit
```

18. Detach p1 from atm 5/0.2. Merge policy mpl_5 is detached and deleted and only p2 is now attached.

```
host1(config)#interface atm 5/0.2
host1(config-subif)#no ip policy output p1
host1(config-subif)#exit
```

19. Detach p2 from atm 5/0.2.

```
host1(config)#interface atm 5/0.2
host1(config-subif)#no ip policy output p2
host1(config-subif)#exit
```

20. Display policies to verify that no merge policies exist and that all other policies have a 0 reference count because they are not attached anywhere.

```
host1#show policy-list
```

```
Policy Table
```

```
-----
```

```
IP Policy p1
```

```
Administrative state: enable
Reference count:      0
Classifier control list: C2, precedence 10
filter
Classifier control list: C1, precedence 90
forward
  Virtual-router: default
  List:
    next-hop 10.1.1.1, order 100, rule 2 (active)
```

```
IP Policy p2
Administrative state: enable
Reference count:      0
Classifier control list: C3, precedence 10
filter
Classifier control list: C1, precedence 90
forward
  Virtual-router: default
  List:
    next-hop 20.1.1.1, order 100, rule 3 (active)
Classifier control list: *, precedence 1000
forward
```

```
IP Policy p3
Administrative state: enable
Reference count:      0
Classifier control list: C1, precedence 80
color yellow
Classifier control list: C4, precedence 900
color red
```

- Related Documentation**
- [Merging Policies Overview on page 17](#)
 - [Monitoring Policy Management Overview](#)
 - [atm classifier-list](#)
 - [classifier-group](#)
 - [color](#)
 - [filter](#)
 - [forward next-hop](#)
 - [interface atm](#)
 - [show ip interface](#)
 - [show policy-list](#)

CHAPTER 5

Examples

- [Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers on page 59](#)
- [Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers on page 64](#)

Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers

This section provides examples showing the configuration of policies that use the Ascend-Data-Filter attribute for IPv4 subscribers.

In this example, the following Ascend-Data-Filter attribute creates a RADIUS record that configures an input policy. The policy filters all packets from network 10.2.1.0 with wildcard mask 0.0.0.255 to any destination.

```
Ascend-Data-Filter="01000100 0A020100 00000000 18000000 00000000  
00000000"
```

[Table 6 on page 59](#) lists the values specified in the Ascend-Data-Filter attribute.

Table 6: Ascend-Data-Filter Attribute for an Input Policy on an IPv4 Interface

Action or Classifier	Hex Value	Actual Value
Type	01	IPv4
Filter or Forward	00	Filter
Indirection	01	Ingress
Spare	00	None
Source IP address	0a020100	10.2.1.0
Destination IP address	00000000	Any
Source IP mask	18	24 (0.0.0.255)
Destination IP mask	00	0 (255,255,255,255)
Protocol	00	None

Table 6: Ascend-Data-Filter Attribute for an Input Policy on an IPv4 Interface (*continued*)

Action or Classifier	Hex Value	Actual Value
Established	00	None
Source port	0000	None
Destination port	0000	None
Source port qualifier	00	None
Destination port qualifier	00	None
Reserved	0000	None

Use the **show classifier-list** and **show policy-list** commands to view information about the policy:

```
host1#show classifier-list
```

```
Classifier Control List Table
-----
```

```
IP clin_1800020_00.1 ip 10.2.1.0 0.0.0.255 any
```

```
host1#show policy-list
```

```
Policy Table
-----
```

```
IP Policy plin_ip_1800020
Administrative state: enable
Reference count:      1
Classifier control list: clin_1800020_00, precedence 100
filter
```

```
Referenced by interface(s):
  ATM4/0.0 input policy, statistics enabled, virtual-router default
```

```
Referenced by profile(s):
  No profile references
```

In this example, the Ascend-Data-Filter attribute is used to create RADIUS records that configure two policies. The first policy is an input policy that filters all TCP packets that come from a port greater than 9000 on host 10.2.1.1 and that go to any destination. The second policy is an output policy that filters all UDP packets from network 20.1.0.0 to host 10.2.1.1, port 3090.

```
Ascend-Data-Filter = "01000100 0A020101 00000000 20000600 23280000
03000000"
Ascend-Data-Filter = "01000000 14010000 0A020101 10201100 00000C12 00020000"
```

Using the **show classifier-list** and **show policy-list** commands produces the following information about the new policies:

```
host1#show classifier-list
```


Classifier Control List Table

```

-----
IP clin_1800021_00.1 tcp 10.2.1.1 gt 9000 any
IP clout_1800021_01.1 udp 20.1.0.0 0.0.255.255 10.2.1.1 eq 3090

```

```
host1#show policy-list
```

Policy Table

```

-----
IP Policy plin_ip_1800021
Administrative state: enable
Reference count:      1
Classifier control list: clin_1800021_00, precedence 100
filter

Referenced by interface(s):
  ATM4/0.0 input policy, statistics enabled, virtual-router default

Referenced by profile(s):
  No profile references

IP Policy plout_ip_1800021
Administrative state: enable
Reference count:      1
Classifier control list: clout_1800021_01, precedence 100
filter

Referenced by interface(s):
  ATM4/0.0 output policy, statistics enabled, virtual-router default

Referenced by profile(s):
  No profile references

```

This example creates an input policy and an output policy, each with multiple rules. The rules for the two policies are shown in the following list:

- Input policy rules
 - Forward all TCP packets from host 10.2.1.1 to destination 20.0.0.0 0.255.255.255
 - Filter all TCP packets from host 10.2.1.1 to any destination.
 - Forward all packets from host 10.2.1.1 to any destination.
 - Filter all other traffic.

The rules for the input policy translate to the following VSAs. The VSAs must be specified in this order:

```

Ascend-Data-Filter = "01010100 0A020101 14000000 20080600 00000000
00000000"
Ascend-Data-Filter = "01000100 0A020101 00000000 20000600 00000000
00000000"
Ascend-Data-Filter = "01010100 0A020101 00000000 20000000 00000000
00000000"
Ascend-Data-Filter = "01000100 00000000 00000000 00000000 00000000
00000000"

```

- Output policy rules

- Forward all TCP packets from 20.0.0.0 0.255.255.255 to host 10.2.1.1.
- Filter all TCP packets from any source to host 10.2.1.1.
- Forward all packets from any source to host 10.2.1.1.
- Filter all other traffic.

The rules for the input policy translate to the following VSAs. The VSAs must be specified in this order:

```
Ascend-Data-Filter = "01010000 14000000 0A020101 08200600 00000000
00000000"
Ascend-Data-Filter = "01000000 00000000 0A020101 00200600 00000000
00000000"
Ascend-Data-Filter = "01010000 00000000 0A020101 00200000 00000000
00000000"
Ascend-Data-Filter = "01000000 00000000 00000000 00000000 00000000
00000000"
```

Using the **show classifier-list** and **show policy-list** commands produces the following information about the new policies:

host1#show classifier-list

```
Classifier Control List Table
-----
IP clin_1800022_00.1 tcp host 10.2.1.1 20.0.0.0 0.255.255.255
IP clin_1800022_01.1 tcp host 10.2.1.1 any
IP clin_1800022_02.1 ip host 10.2.1.1 any
IP clout_1800022_04.1 tcp 20.0.0.0 0.255.255.255 host 10.2.1.1
IP clout_1800022_05.1 tcp any host 10.2.1.1
IP clout_1800022_06.1 ip any host 10.2.1.1
```

host1#show policy-list

```
Policy Table
-----
IP Policy plin_ip_1800022
Administrative state: enable
Reference count: 1
Classifier control list: clin_1800022_00, precedence 100
forward
Classifier control list: clin_1800022_01, precedence 100
filter
Classifier control list: clin_1800022_02, precedence 100
forward
Classifier control list: *, precedence 100
filter

Referenced by interface(s):
ATM4/0.0 input policy, statistics enabled, virtual-router default

Referenced by profile(s):
No profile references

IP Policy plout_ip_1800022
Administrative state: enable
Reference count: 1
Classifier control list: clout_1800022_04, precedence 100
```

```

    forward
Classifier control list: clout_1800022_05, precedence 100
    filter
Classifier control list: clout_1800022_06, precedence 100
    forward
Classifier control list: *, precedence 100
    filter

Referenced by interface(s):
    ATM4/0.0 output policy, statistics enabled, virtual-router default

Referenced by profile(s):
    No profile reference

```

In this example, the following Ascend-Data-Filter attribute creates a RADIUS record that configures an input policy on an IPv4 interface. The policy filters TCP packets from host address 10.2.1.2 to any destination. The policy marks the packets with a ToS byte of 5 and a mask of 170. The policy also applies a traffic class named someTcl and a rate-limit profile named someRlp.

```

Ascend-Data-Filter="01010100 0a020102 00000000 20000600 045708ae 02010000
05aa0773 6f6d6554 636c0773 6f6d6552 6c70"

```

[Table 7 on page 63](#) lists the values specified in the Ascend-Data-Filter attribute.

Table 7: Ascend-Data-Filter Attribute Values for a RADIUS Record

Action or Classifier	Hex Value	Actual Value
Type	01	IPv4
Forward	01	Filter
Indirection	01	Ingress
Spare	00	None
Source IP address	0a020102	10.2.1.2
Destination IP address	00000000	Any
Source IP mask	20	32 (0.0.0.0)
Destination IP mask	00	0 (255,255,255,255)
Protocol	06	TCP
Established	00	None
Source port	0000	None
Destination port	0000	None
Source port qualifier	00	None

Table 7: Ascend-Data-Filter Attribute Values for a RADIUS Record (*continued*)

Action or Classifier	Hex Value	Actual Value
Destination port qualifier	00	None
Reserved	0000	None
Marking value	05	5
Marking mask	aa	170
Traffic class	0773 6f6d6554 636c	someTcl
Rate-limit profile	0773 6f6d6552 6c70	someRlp

```
host1#show classifier-list
```

```
Classifier Control List Table
```

```
IP clin_1800023_00.1 tcp host 10.2.1.2
```

```
host1#show policy-list
```

```
Policy Table
```

```
IP Policy plin_ip_1800023
Administrative state: enable
Reference count: 1
Classifier control list: clin_1800023_00, precedence 100
mark 5 mask 170
traffic-class someTcl
rate-limit-profile someRlp

Referenced by interface(s):
ATM11/0.0 input policy, statistics enabled, virtual-router default

Referenced by profile(s):
No profile references
```

- Related Documentation**
- [Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers on page 64](#)
 - [Using RADIUS to Create and Apply Policies Overview on page 10](#)

Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers

This section provides examples showing the configuration of policies that use the Ascend-Data-Filter attribute when there are IPv6 subscribers in a network.

In this example, the following two Ascend-Data-Filter attributes are used to create RADIUS records that configure two policies. The first policy is an output policy that filters all UDP packets from network 2001:82ab:1020:87ec::0/64 to host 2001:82ab:1020:87ec:1234:0917:3415:0012, port 3090. The second policy is an input

policy that filters all TCP packets that come from a port greater than 9000 on host 2001:82ab:1020:87ec:1234:0917:3415:0012 and that go to any destination.

```
Ascend-Data-Filter1 = "03000000 300182ab 102087ec 00000000 00000000
200182ab 102087ec 12340917 34150012 40801100 00000C12 00020000"
Ascend-Data-Filter2 = "03000100 200182ab 102087ec 12340917 34150012 00000000
00000000 00000000 00000000 80000600 23280000 03000000"
```

Table 8 on page 65 lists the values specified in the Ascend-Data-Filter1 attribute that are used to create an output policy.

Table 8: Ascend-Data-Filter Attribute for an Output Policy on an IPv6 Interface

Action or Classifier	Hex Value	Actual Value
Type	03	IPv6
Forward	00	Filter
Indirection	00	Egress
Spare	00	None
Source IPv6 address	300182ab 102087ec 00000000 00000000	3001:82ab:1020:87ec:0000:0000:0000:0000
Destination IPv6 address	200182ab 102087ec 12340917 34150012	2001:82ab:1020:87ec:1234:0917:3415:0012
Source IPv6 prefix	40	64
Destination IPv6 prefix	80	128
Protocol	11	UDP
Established	00	None
Source port	0000	None
Destination port	0C12	3090
Source port qualifier	00	None
Destination port qualifier	02	Equal to
Reserved	0000	None

Table 9 on page 66 lists the values specified in the Ascend-Data-Filter2 attribute that are used to create an input policy.

Table 9: Ascend-Data-Filter Attribute for an Input Policy on an IPv6 Interface

Action or Classifier	Hex Value	Actual Value
Type	03	IPv6
Forward	00	Filter
Indirection	01	Ingress
Spare	00	None
Source IPv6 address	200182ab102087ec1234091734150012	2001:82ab:1020:87ec:1234:0917:3415:0012
Destination IPv6 address	00000000 00000000 00000000 00000000	Any
Source IPv6 prefix	80	128
Destination IPv6 prefix	00	0
Protocol	06	TCP
Established	00	None
Source port	2328	9000
Destination port	0000	None
Source port qualifier	03	Greater than
Destination port qualifier	00	None
Reserved	0000	None

Use the **show classifier-list** and **show policy-list** commands to view information about the configured input and output policies:

host1#show classifier-list

Classifier Control List Table

```
IPv6 clout_1800020_00.1 udp source-address 3001:82ab:1020:87ec::/64
destination-host
2001:82ab:1020:87ec:1234:917:3415:12 destination-port eq 3090
IPv6 clin_1800020_01.1 tcp source-host 2001:82ab:1020:87ec:1234:917:3415:12
source-port gt 9000
```

host1#show policy-list

Policy Table

```
IPv6 Policy plout_ipv6_1800020
```

Administrative state: enable
 Reference count: 1
 Classifier control list: clout_1800020_00, precedence 100
 filter

Referenced by interface(s):
 GigabitEthernet10/0.2 output policy, statistics enabled, virtual-router
 default

Referenced by profile(s):
 None

Referenced by merged policies:
 None

IPv6 Policy plin_ipv6_1800020
 Administrative state: enable
 Reference count: 1
 Classifier control list: clin_1800020_01, precedence 100
 filter

Referenced by interface(s):
 GigabitEthernet10/0.2 input policy, statistics enabled, virtual-router
 default

Referenced by profile(s):
 None

Referenced by merged policies:
 None

Related Documentation

- [Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers on page 59](#)
- [Using RADIUS to Create and Apply Policies Overview on page 10](#)

PART 3

Administration

- [Monitoring Tasks on page 71](#)

CHAPTER 6

Monitoring Tasks

- [Monitoring Color-Mark Profiles on page 71](#)

Monitoring Color-Mark Profiles

Purpose Display information about color-mark profiles.

Action To display information about color-mark profiles:

host1#show color-mark-profile A

	Color Mark Profile Table

IP Color-Mark-Profile: A	
Mask:	255
Green mark:	64
Yellow mark:	-
Red mark:	8

Related Documentation • [show color-mark-profile](#)

PART 4

Index

- [Index on page 75](#)

Index

A

atm commands	
atm-cell-mode.....	42

C

classifier groups	
creating.....	3
classifier-group commands	
classifier-group.....	44
conventions	
notice icons.....	ix
text and syntax.....	x
customer support.....	xi
contacting JTAC.....	xi

D

documentation set	
comments on.....	xi

E

explicit packet coloring.....	6
-------------------------------	---

F

forward command.....	38, 43
forward interface command.....	38
forward next-hop command.....	38

I

IP auxiliary input policy.....	26
IP options, filtering	42

M

manuals	
comments on.....	xi
mark-exp command.....	6
merged policy naming conventions.....	21
merging policies.....	17
configuration example.....	47
error conditions.....	24
naming conventions.....	21
persistent configuration differences.....	22

policy attachment rules.....	23
policy attachment sequence.....	22
reference counting.....	22
resolving conflicts.....	19
restrictions.....	17
rules for attachment.....	17
multiple forwarding solutions.....	43

N

notice icons.....	ix
-------------------	----

O

overlapping classification.....	26
---------------------------------	----

P

packet coloring, explicit.....	6
packet tagging.....	42
parent group merge algorithm.....	24
policies	
using RADIUS to create	
overview.....	10
policy attachment rules.....	23
policy lists	
applying	
overview.....	7
policy management	
applications	
packet tagging.....	42
classifier groups, creating.....	3
explicit packet coloring.....	6
filtering IP options.....	42
merging policies.....	17
packet tagging.....	6, 42
policy rules, creating.....	3
policy rule commands	
forward.....	38
forward interface.....	38
forward next-hop.....	38
policy rules	
creating.....	3
precedence.....	4
supported commands.....	4

R

resolving merge conflicts.....	19
--------------------------------	----

S

Service Manager	
merging policies.....	17

show commands
 show color-mark-profile71
support, technical See technical support

T

technical support
 contacting JTAC.....xi
text and syntax conventions.....x
traffic-class command.....6