Chapter 2

Routing Policy Framework Overview

All routing protocols store their routing information in routing tables. From these tables, the routing protocols calculate the best route to each destination and place these routes in a forwarding table. These routes are then used to forward routing protocol traffic toward a destination, and they can be advertised to neighbors using one or more routing protocols.

NOTE: Instead of referring to the multiple routing tables that the JUNOS software maintains, the discussion in the rest of this chapter assumes the inet.0 routing table unless explicitly stated otherwise. By default, the JUNOS software stores unicast IP version 4 (IPv4) routes in the inet.0 routing table. For information about all the routing tables, see "Routing Tables Affected by Routing Policies" on page 20.

In general, the routing protocols place all their routes in the routing table and advertise a limited set of routes from the routing table. The general rules for handling the routing information between the routing protocols and the routing table are known as the *routing policy framework*.

The routing policy framework is composed of default rules for each routing protocol that determine which routes the protocol places in the routing table and advertises from the routing table. The default rules for each routing protocol are known as *default routing policies*.

You can create routing policies to preempt the default policies, which are always present. A *routing policy* is a mechanism in the JUNOS software that allows you to modify the routing policy framework to suit your needs. You can create and implement your own routing policies to do the following:

- Control which routes a routing protocol places in the routing table.
- Control which active routes a routing protocol advertises from the routing table. (An *active route* is a route that is chosen from all routes in the routing table to reach a destination. For information about the active route selection process, see the *JUNOS Routing Protocols Configuration Guide*.)
- Manipulate the route characteristics as a routing protocol places it in the routing table or advertises it from the routing table.

You can manipulate the route characteristics to control which route is selected as the active route to reach a destination. The active route is placed in the forwarding table and used to forward traffic toward the route's destination. In general, the active route is also advertised to a router's neighbors.

To create a routing policy, you must define the policy and apply it. You define the policy by specifying the criteria that a route must match and the actions to perform if a match occurs. You then apply the policy to a routing protocol or to the forwarding table.

This chapter discusses the following topics related to understanding and creating routing policies:

NOTE: Before you create your routing policies, we recommend that you read through this entire section to become familiar with the terminology, concepts, and configuration guidelines.

Importing and Exporting

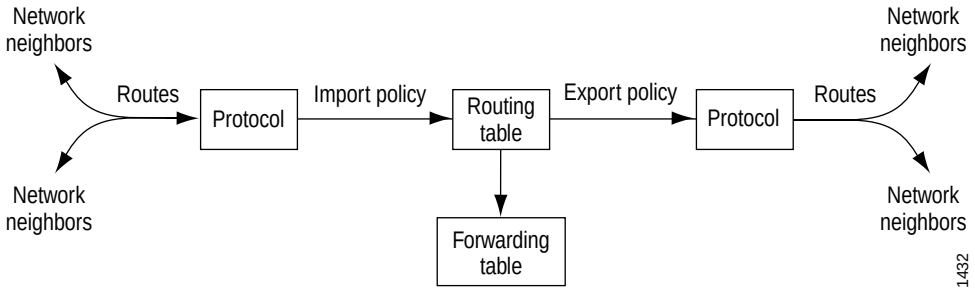
Two terms—*import* and *export*—explain how routes move between the routing protocols and the routing table (see Figure 5):

When the Routing Engine places the routes of a routing protocol into the routing table, it is *importing* routes into the routing table.

When the Routing Engine uses active routes from the routing table to send a protocol advertisement, it is *exporting* routes from the routing table.

NOTE: The process of moving routes between a routing protocol and the routing table is described always *from the point of view of the routing table*. That is, routes are *imported int o* a routing table from a routing protocol and they are *exported from* a routing table to a routing protocol. Remember this distinction when working with routing policies.

Figure 5: Importing and Exporting Routes



When evaluating routes for export, the Routing Engine uses only active routes from the routing table. For example, if a routing table contains multiple routes to the same destination and one route has a preferable metric, only that route is evaluated. In other words, an export policy does not evaluate all routes; it evaluates only those routes that a routing protocol is allowed to advertise to a neighbor. For more information about the active path selection algorithm, see the *JUNOS Routing Protocols Configuration Guide*.

NOTE: By default, the Border Gateway Protocol (BGP) advertises active routes. However, you can configure BGP to advertise *inactive routes*, which go to the same destination as other routes but have less preferable metrics. For more information about advertising inactive routes, see the *JUNOS Routing Protocols Configuration Guide*.

Table 4 on page 20 lists the routing protocols from which the routing table can import routes and to which the routing table can export routes. Table 4 also lists direct and explicitly configured routes, which for the purposes of this table are considered a pseudoprotocol. (An *explicitly configured route* is a route that you have configured. *Direct routes* are not explicitly configured; they are created as a result of IP addresses being configured on an interface.) Explicitly configured routes include aggregate, generated, local, and static routes. (An *aggregate route* is a route that distills groups of routes with common addresses into one route. A *generated route* is a route used when the routing table has no information about how to reach a particular destination. A *local route* is an IP address assigned to a router interface. A *static route* is a nonchanging route to a destination.)

The policy framework software treats direct and explicitly configured routes as if they are learned through routing protocols; therefore, they can be imported into the routing table. Routes cannot be exported from the routing table to the pseudoprotocol, because this protocol is not a real routing protocol. However, aggregate, direct, generated, and static routes can be exported from the routing table to routing protocols, whereas local routes cannot.

For information about the default routing policies for each routing protocol, see Table 6 on page 21. For information about the import and export routing policies supported for each routing protocol and the level at which you can apply these policies, see Table 8 on page 28.

Table 4: Protocols That Can Be Imported to and Exported from the Routing Table

Protocol	Import	Export
BGP	Yes	Yes
Distance Vector Multicast Routing Protocol (DVMRP)	Yes	Yes
Intermediate System-to-Intermediate System (IS-IS)	Yes	Yes
Label Distribution Protocol (LDP)	Yes	Yes
Multiprotocol Label Switching (MPLS)	Yes	No
Open Shortest Path First (OSPF)	Yes	Yes
Protocol Independent Multicast (PIM) dense mode	Yes	Yes
PIM sparse mode	Yes	Yes
PIM sparse-dense mode	Yes	Yes
Pseudoprotocol: Direct routes Explicitly configured routes Aggregate routes Generated routes Local routes Static routes	Yes	No
Routing Information Protocol (RIP) and Routing Information Protocol next generation (RIPng)	Yes	Yes

Routing Tables Affected by Routing Policies

Table 5 lists the routing tables affected by default and user-defined routing policies and the types of routes that each routing table stores.

Table 5: Routing Tables Affected by Routing Policies

Routing Table	Type of Routes Stored
inet.0	Unicast IPv4 routes
instance-name.inet.0	Unicast IPv4 routes for a particular routing instance
inet.1	Multicast IPv4 routes
inet.2	Unicast IPv4 routes for multicast reverse-path forwarding (RPF) lookup
inet.3	MPLS routes
mpls.0	MPLS routes for label-switched path (LSP) next hops
inet6.0	Unicast IP version 6 (IPv6) routes

NOTE: The discussion in the rest of this chapter assumes the inet.0 routing table unless explicitly stated otherwise.

For more information about routing tables, see the *JUNOS Routing Protocols Configuration Guide*.

Default Routing Policies and Actions

You must be familiar with the default routing policies to know when you need to modify them to suit your needs. Table 6 summarizes the default routing policies for each routing protocol that imports and exports routes. The actions in the default routing policies are taken if you have not explicitly configured a routing policy. This table also shows direct and explicitly configured routes, which for the purposes of this table are considered a pseudoprotocol. Explicitly configured routes include aggregate, generated, and static routes.

The default import policy is always the same: accept all routes learned from the protocol. Table 6 includes information about the routing tables used by each protocol.

Table 6: Default Routing Policies

Importing or Exporting Protocol	Default Import Policy	Default Export Policy
BGP	Accept all BGP IPv4 routes learned from configured neighbors and import into the inet.0 routing table. Accept all BGP IPv6 routes learned from configured neighbors and import into the inet6.0 routing table.	Accept and export active BGP routes.
DVMRP	Accept all DVMRP routes and import into the inet.1 routing table.	Accept and export active DVMRP routes.
IS-IS	Accept all IS-IS routes and import into the inet.0 and inet6.0 routing tables. (You cannot override or change this default policy.)	Reject everything. (The protocol uses flooding to announce local routes and any learned routes.)
LDP	Accept all LDP routes and import into the inet.3 routing table.	Accept and export active LDP routes.
MPLS	Accept all MPLS routes and import into the inet.3 routing table.	Accept and export active MPLS routes.
OSPF	Accept all OSPF routes and import into the inet.0 routing table. (You cannot override or change this default policy.)	Reject everything. (The protocol uses flooding to announce local routes and any learned routes.)
PIM dense mode	Accept all PIM dense mode routes and import into the inet.1 routing table.	Accept active PIM dense mode routes.
PIM sparse mode	Accept all PIM sparse mode routes and import into the inet.1 routing table.	Accept and export active PIM sparse mode routes.
Pseudoprotocol: Direct routes Explicitly configured routes: Aggregate routes Generated routes Static routes	Accept all direct and explicitly configured routes and import into the inet.0 routing table.	The pseudoprotocol cannot export any routes from the routing table because it is not a routing protocol. Routing protocols can export these or any routes from the routing table.

Importing or Exporting Protocol	Default Import Policy	Default Export Policy
RIP	Accept all RIP routes learned from configured neighbors and import into the inet.0 routing table.	Reject everything. To export RIP routes, you must configure an export policy for RIP.
RIPng	Accept all RIPng routes learned from configured neighbors and import into the inet6.0 routing table.	Reject everything. To export RIPng routes, you must configure an export policy for RIPng.
Test policy	Accept all routes. For additional information about test policy, see “Routing Policy Tests” on page 35.	

When multiple routes for a destination exist in the routing table, the protocol selects an active route and that route is placed in the appropriate routing table. For equal-cost routes, the JUNOS software places multiple next hops in the appropriate routing table.

When a protocol is exporting routes from the routing table, it exports active routes only. This applies to actions specified by both default and user-defined export policies.

You cannot change the default import policy for the link-state protocols IS-IS and OSPF. As link-state protocols, IS-IS and OSPF exchange routes between systems within an autonomous system (AS). All routers and systems within an AS must share the same link-state database, which includes routes to reachable prefixes and the metrics associated with the prefixes. If an import policy is configured and applied to IS-IS or OSPF, some routes might not be learned or advertised or the metrics for learned routes might be altered, which would make a consistent link-state database impossible.

The default export policy for IS-IS and OSPF protocols is to reject everything. These protocols do not actually export their internally learned routes (the directly connected routes on interfaces that are running the protocol). Both IS-IS and OSPF protocols use a procedure called *flooding* to announce local routes and any routes learned by the protocol. The flooding procedure is internal to the protocol, and is unaffected by the policy framework. Exporting can be used only to announce information from other protocols, and the default is not to do so.

For information about the routing protocols from which the routing table can import routes and to which routing protocols the routing table can export routes, see Table 4 on page 20. For information about the user-defined import and export policies supported for each routing protocol and the level at which you can apply these policies, see Table 8 on page 28.

The following default actions are taken if the following situations arise during policy evaluation:

If a policy does not specify a match condition, all routes evaluated against the policy match.

If a match occurs but the policy does not specify an accept, reject, next term, or next policy action, one of the following occurs:

The next term, if present, is evaluated.

If no other terms are present, the next policy is evaluated.

If no other policies are present, the action specified by the default policy is taken.

If a match does not occur with a term in a policy and subsequent terms in the same policy exist, the next term is evaluated.

If a match does not occur with any terms in a policy and subsequent policies exist, the next policy is evaluated.

If a match does not occur by the end of a policy or all policies, the accept or reject action specified by the default policy is taken.

Creating Routing Policies

The following are typical circumstances under which you might want to preempt the default routing policies in the routing policy framework by creating your own routing policies:

You do not want a protocol to import all routes into the routing table. If the routing table does not learn about certain routes, they can never be used to forward packets and they can never be redistributed into other routing protocols.

You do not want a routing protocol to export all the active routes it learns.

You want a routing protocol to announce active routes learned from another routing protocol, which is sometimes called *route redistribution* .

You want to manipulate route characteristics, such as the preference value, AS path, or community. You can manipulate the route characteristics to control which route is selected as the active route to reach a destination. In general, the active route is also advertised to a router's neighbors.

You want to change the default BGP route flap-damping parameters.

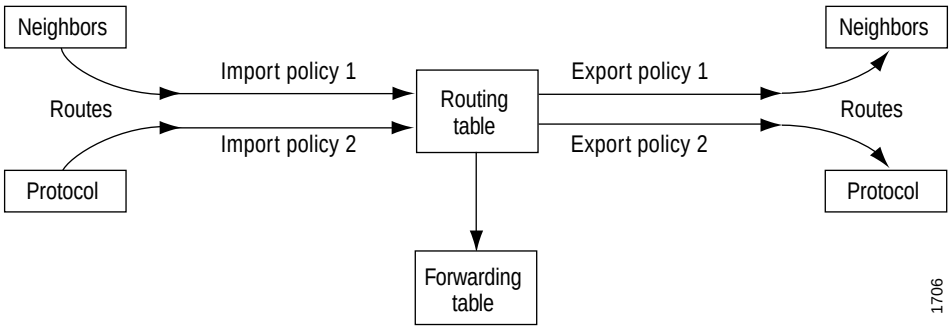
You want to perform per-packet load balancing.

You want to enable class of service (CoS).

Configuring a Routing Policy

As shown in Figure 6 on page 24, you use *import routing policies* to control which routes routing protocols place in the routing table, and *export routing policies* to control which routes a routing protocol advertises from the routing table to its neighbors.

Figure 6: Importing and Exporting Routing Policies



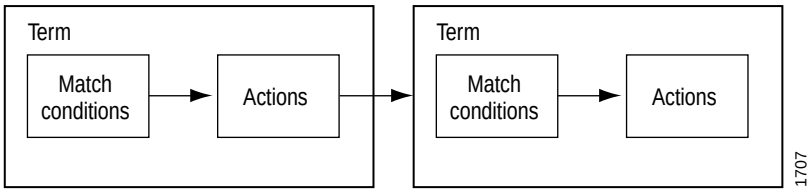
To create a routing policy, you must define the following components:

Match conditions —Criteria that a route must match. If a route matches all of the criteria, one or more actions are applied to the route.

Actions—What to do if a route matches. The actions can specify whether to accept or reject the route, control how a series of policies is evaluated, and manipulate the characteristics associated with a route. You can configure one or more actions.

You typically define match conditions and actions within a *term*. Figure 7 shows the routing policy components, including the term.

Figure 7: Routing Policy Components



After defining a routing policy, you then apply it to a routing protocol or to the forwarding table.

This section provides more information about creating routing policies:

Match Conditions

A *match condition* defines the criteria that a route must match. You can define one or more match conditions. If a route matches all match conditions, one or more actions are applied to the route.

Match conditions fall into two categories: standard and extended. In general, the extended match conditions include criteria that are defined separately from the routing policy (AS path regular expressions, communities, and prefix lists) and are more complex than standard match conditions. The extended match conditions provide many powerful capabilities. For more information about them, see “Extended Match Conditions Configuration” on page 93. The standard match conditions include criteria that are defined within a routing policy and are less complex than the extended match conditions, also called Named Match Conditions.

Table 7 describes each match condition, including its category, when you typically use it, and any relevant notes about it. For more information about match conditions, see Table 9 on page 44.

Table 7: Match Conditions

Match Condition	Category	When to Use	Notes
AS path regular expression—A combination of AS numbers and regular expression operators.	Extended	(BGP only) Match a route based on its AS path. (An AS path consists of the AS numbers of all routers a packet must go through to reach a destination.) You can specify an exact match with a particular AS path or a less precise match.	You use regular expressions to match the AS path.
Community—A group of destinations that share a property. (Community information is included as a path attribute in BGP update messages.)	Extended	Match a group of destinations that share a property. Use a routing policy to define a community that specifies a group of destinations you want to match and one or more actions that you want taken on this community.	Actions can be performed on the entire group. You can create multiple communities associated with a particular destination. You can create match conditions using regular expressions.
Prefix list—A named list of IP addresses.	Extended	Match a route based on prefix information. You can specify an exact match of a particular route only.	You can specify a common action only for all prefixes in the list.

Match Condition	Category	When to Use	Notes
Route list—A list of destination prefixes.	Extended	Match a route based on prefix information. You can specify an exact match of a particular route or a less precise match.	You can specify an action for each prefix in the route list or a common action for all prefixes in the route list.
Standard—A collection of criteria that can match a route.	Standard	Match a route based on one of the following criteria: area ID, color, external route, family, instance (routing), interface name, level number, local preference, metric, neighbor address, next-hop address, origin, preference, protocol, routing table name, or tag. For the protocol criterion, you can specify one of the following: BGP, direct, DVMRP, IS-IS, local, MPLS, OSPF, PIM dense mode, PIM sparse mode, RIP, RIPng, static, and aggregate.	None.
Subroutine—A routing policy that is called repeatedly from another routing policy.	Extended	Use an effective routing policy in other routing policies. You can create a subroutine that you can call over and over from other routing policies.	The subroutine action influences but does not necessarily determine the final action. For more information, see “How a Routing Policy Subroutine Is Evaluated” on page 33.

Named Match Conditions

Some match conditions are defined separately from the routing policy and are given names. You then reference the name of the match condition in the definition of the routing policy itself. Named match conditions allow you to do the following:

- Reuse match conditions in other routing policies.
- Read configurations that include complex match conditions more easily.

Named match conditions include communities, prefix lists, and AS path regular expressions. For more information about these match conditions, see Table 7 on page 25.

Actions

An *action* is what the policy framework software does if a route matches all criteria defined in a match condition. You can configure one or more actions in a term. The policy framework software supports the following types of actions:

- Flow control actions, which affect whether to accept or reject the route or whether to evaluate the next term or routing policy
- Actions that manipulate route characteristics
- Trace action, which logs route matches

Manipulating the route characteristics allows you to control which route is selected as the active route to reach a destination. In general, the active route is also advertised to a routing platform's neighbors. You can manipulate the following route characteristics: AS path, class, color, community, damping parameters, destination class, external type, next hop, load balance, local preference, metric, origin, preference, and tag.

For the numeric information (color, local preference, metric, preference, and tag), you can set a specific value or change the value by adding or subtracting a specified amount. The addition and subtraction operations do not allow the value to exceed a maximum value and drop below a minimum value.

For more information about the properties you can change and the addition and subtraction operations, see Table 11 on page 49.

Terms

A *term* is a named structure in which match conditions and actions are defined. You can define one or more terms.

In general, the policy framework software compares a route against the match conditions in the first term in the first routing policy, then goes on to the next term and the next policy if present, and so on, until an explicitly configured or default action of accept or reject is taken. Therefore, the order in which you arrange terms in a policy is relevant.

The order of match conditions in a term is not relevant since a route must match all match conditions in a term for an action to be taken.

Routing Policy Application

After defining a routing policy, as discussed in "Match Conditions" on page 25 and "Actions" on page 26, you can apply it to one of the following:

Routing protocols—BGP, DVMRP, IS-IS, LDP, MPLS, OSPF, PIM dense mode, PIM sparse mode, PIM sparse-dense mode, RIP, and RIPng

Pseudoprotocol—Explicitly created routes, which include aggregate and generated routes

Forwarding table

The following sections discuss the following topics:

Routing Protocols on page 28

Forwarding Table on page 29

Routing Protocols

When applying routing policies to routing protocols, you must know whether each protocol supports import and export policies and the level at which you can apply these policies. Table 8 summarizes the import and export policy support for each routing protocol. Table 8 also lists explicitly configured routes, which for the purposes of this table are considered a pseudoprotocol. Explicitly configured routes include aggregate and generated routes.

You can apply an import policy to aggregate and generated routes, but you cannot apply an export policy to these routes. These routes cannot be exported from the routing table to the pseudoprotocol, because this protocol is not a real routing protocol. However, aggregate and generated routes can be exported from the routing table to routing protocols.

You cannot apply import policies to the link-state protocols IS-IS and OSPF. As link-state protocols, IS-IS and OSPF exchange routes between systems within an AS. All routers and systems within an AS must share the same link-state database, which includes routes to reachable prefixes and the metrics associated with the prefixes. If an import policy is configured and applied to IS-IS or OSPF, some routes might not be learned or advertised or the metrics for learned routes might be altered, which would make a consistent link-state database impossible.

For BGP only, you can also apply import and export policies at group and peer levels as well as at the global level. A peer import or export policy overrides a group import or export policy. A group import or export policy overrides a global import or export policy.

For example, if you define an import policy for an individual peer at the peer level and also define an import policy for the group to which it belongs, the import policy defined for the peer level only is invoked. The group import policy is not used for that peer, but it is applied to other peers in that group.

For RIP and RIPng only, you can apply import policies at the global and neighbor levels and export policies at a group level. For more information about RIP and RIPng, see the *JUNOS Routing Protocols Configur ation Guide* .

For information about the routing protocols from which the routing table can import routes and to which routing protocols the routing table can export routes, see Table 4 on page 20. For information about the default routing policies for each routing protocol, see Table 6 on page 21.

Table 8: Apply Routing Policies to Protocols

Protocol	Import Policy	Export Policy	Supported Levels
BGP	Yes	Yes	Import: global, group, peer Export: global, group, peer
DVMRP	Yes	Yes	Global
IS-IS	No	Yes	Export: global
LDP	Yes	Yes	Global
MPLS	No	No	–

Protocol	Import Policy	Export Policy	Supported Levels
OSPF	No	Yes	Export: global
PIM dense mode	Yes	Yes	Global
PIM sparse mode	Yes	Yes	Global
Pseudoprotocol—Explicitly configured routes, which include the following: Aggregate routes Generated routes	Yes	No	Import: global
RIP and RIPng	Yes	Yes	Import: global, neighbor Export: group

Routing Policy Application to Routing Protocols

You can apply the following routing policy elements to a routing protocol:

Routing policy—You can apply a single routing policy to a routing protocol.

Chain of routing policies—You can apply multiple routing policies (chains) to a routing protocol.

Policy expression—You can apply a policy expression to a routing protocol. A *policy expression* uses Boolean logical operators with a routing policy and routing policy chains. The logical operators establish rules by which the policy or chains are evaluated.

Forwarding Table

You can apply export policies to routes being exported from the routing table into the forwarding table for the following features:

Per-packet load balancing

CoS

For more information about per-packet load balancing, see “Configuring Load-Balance Per-Packet Action” on page 138. For more information about CoS, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Evaluating a Routing Policy

This section provides information about how routing policies are evaluated. It discusses the following topics:

For specific information about how the various match conditions are evaluated, see “Match Conditions” on page 43 and “Extended Match Conditions Configuration” on page 93.

How a Routing Policy Is Evaluated

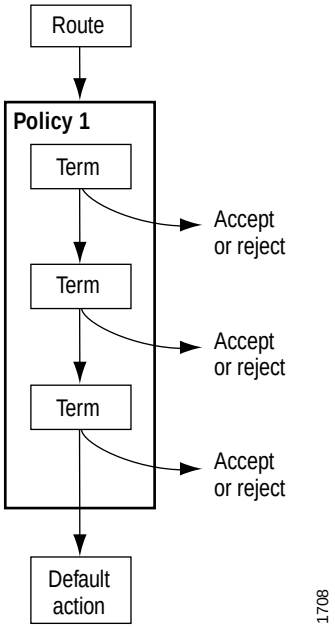
Figure 8 on page 31 shows how a single routing policy is evaluated. This routing policy consists of multiple terms. Each term consists of match conditions and actions to apply to matching routes. Each route is evaluated against the policy as follows:

1. The route is evaluated against the first term. If it matches, the specified action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. If the next term action is specified, if no action is specified, or if the route does not match, the evaluation continues as described in Step 2. If the next policy action is specified, any accept or reject action specified in this term is skipped, all remaining terms in this policy are skipped, all other actions are taken, and the evaluation continues as described in Step 3.

2. The route is evaluated against the second term. If it matches, the specified action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. If the next term action is specified, if no action is specified, or if the route does not match, the evaluation continues in a similar manner against the last term. If the next policy action is specified, any accept or reject action specified in this term is skipped, all remaining terms in this policy are skipped, all other actions are taken, and the evaluation continues as described in Step 3.

3. If the route matches no terms in the routing policy or the next policy action is specified, the accept or reject action specified by the default policy is taken. For more information about the default routing policies, see “Default Routing Policies and Actions” on page 21.

Figure 8: Routing Policy Evaluation



How a Routing Policy Chain Is Evaluated

Figure 9 on page 32 shows how a chain of routing policies is evaluated. These routing policies consist of multiple terms. Each term consists of match conditions and actions to apply to matching routes. Each route is evaluated against the policies as follows:

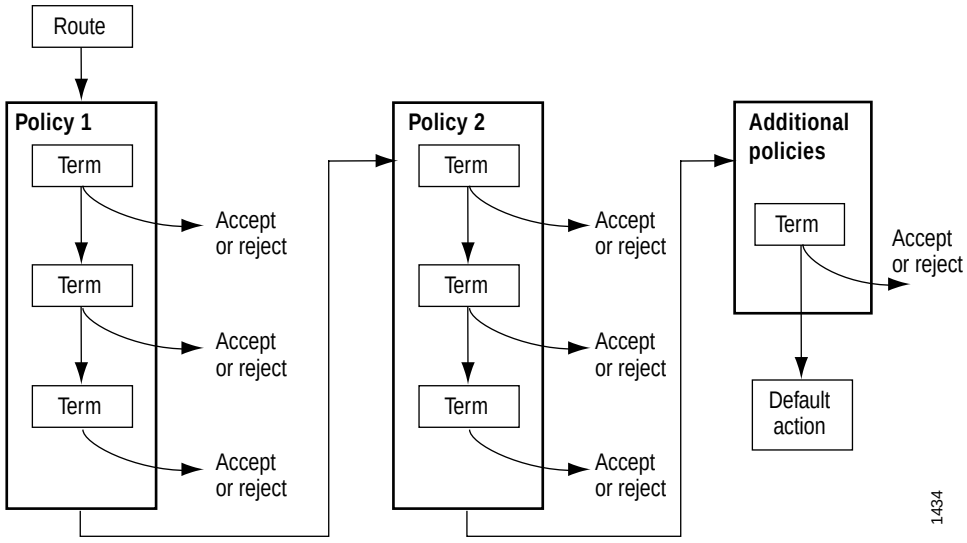
1. The route is evaluated against the first term in the first routing policy. If it matches, the specified action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. If the next term action is specified, if no action is specified, or if the route does not match, the evaluation continues as described in Step 2. If the next policy action is specified, any accept or reject action specified in this term is skipped, all remaining terms in this policy are skipped, all other actions are taken, and the evaluation continues as described in Step 3.

2. The route is evaluated against the second term in the first routing policy. If it matches, the specified action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. If the next term action is specified, if no action is specified, or if the route does not match, the evaluation continues in a similar manner against the last term in the first routing policy. If the next policy action is specified, any accept or reject action specified in this term is skipped, all remaining terms in this policy are skipped, all other actions are taken, and the evaluation continues as described in Step 3.

3. If the route does not match a term or matches a term with a next policy action in the first routing policy, it is evaluated against the first term in the second routing policy.

4. The evaluation continues until the route matches a term with an accept or reject action defined or until there are no more routing policies to evaluate. If there are no more routing policies, then the accept or reject action specified by the default policy is taken. For more information about the default routing policies, see “Default Routing Policies and Actions” on page 21.

Figure 9: Routing Policy Chain Evaluation



How a Routing Policy Expression Is Evaluated

To understand how a policy expression is evaluated, you must first understand the Boolean logical operators and the associated logic used in evaluating a policy expression. For more information about policy expressions, including how they are evaluated, see “Applying Policy Expressions” on page 59.

How a Routing Policy Subroutine Is Evaluated

Figure 10 on page 34 shows how a subroutine is evaluated. The subroutine is included in the first term of the first routing policy in a chain. Each route is evaluated against the subroutine as follows:

1. The route is evaluated against the first term in the first routing policy. If the route does not match all match conditions specified before the subroutine, the subroutine is skipped and the next term in the routing policy is evaluated (see Step 2). If the route matches all match conditions specified before the subroutine, the route is evaluated against the subroutine. If the route matches the match conditions in any of the subroutine terms, two levels of evaluation occur in the following order:

   a. The actions in the subroutine term are evaluated. If one of the actions is accept, evaluation of the subroutine ends and a Boolean value of TRUE is returned to the calling policy. If one of the actions is reject, evaluation of the subroutine ends and FALSE is returned to the calling policy. If one of the actions is meant to manipulate route characteristics, the characteristic is changed regardless of whether accept, reject, or neither action is specified.

      If the subroutine does not specify the accept, reject or next-policy action, it uses the accept or reject action specified by the default policy and the values of TRUE or FALSE are returned to the calling policy as described in the previous paragraph. (For information about what happens if a termination action is not specified in the term, see “Termination Actions” on page 126. For more information about the default routing policies, see “Default Routing Policies and Actions” on page 21.)

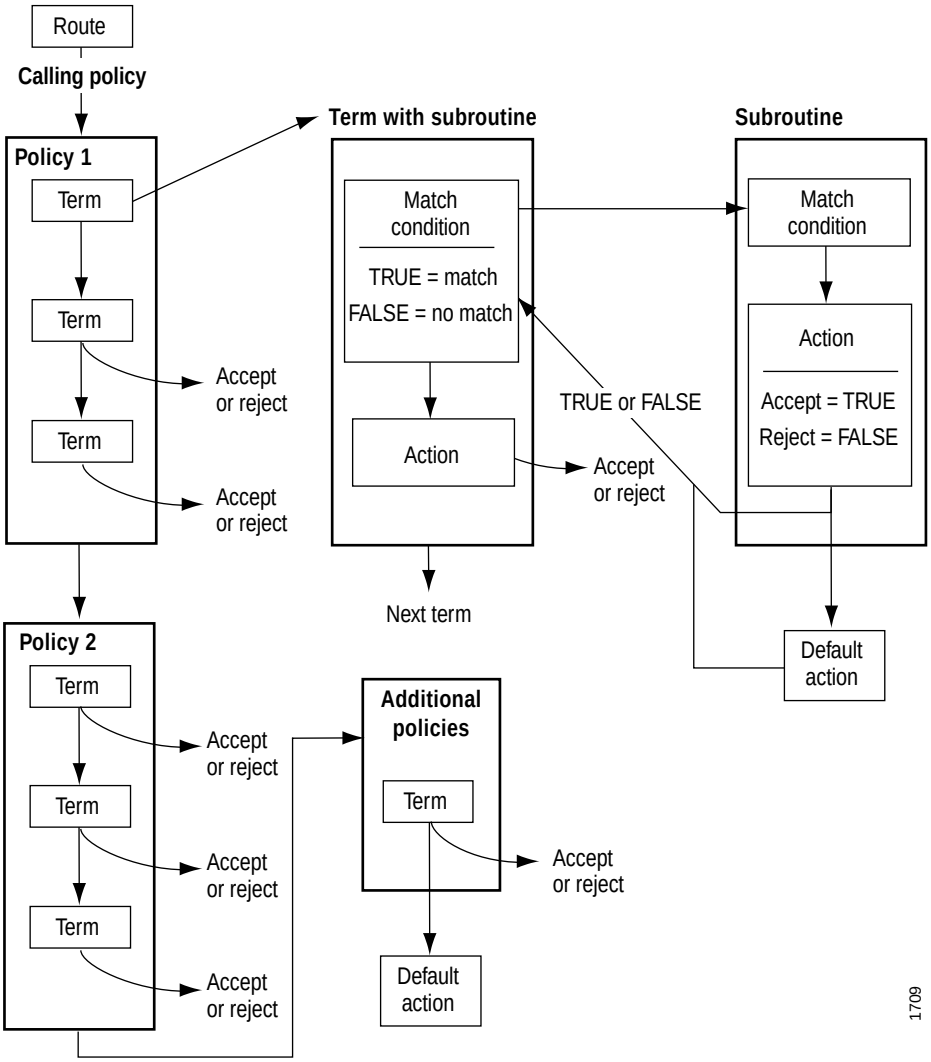
   b. The calling policy’s subroutine match condition is evaluated. During this part of the evaluation, TRUE equals a match and FALSE equals no match. If the subroutine returns TRUE to the calling policy, then the evaluation of the calling policy continues. If the subroutine returns FALSE to the calling policy, then the evaluation of the current term ends and the next term is evaluated.

2. The route is evaluated against the second term in the first routing policy. For information about how the subsequent terms and policies are evaluated, see “How a Routing Policy Chain Is Evaluated” on page 31.

NOTE: If a term defines multiple match conditions, including a subroutine, and a route does not match a condition specified before the subroutine, the evaluation of the term ends and the subroutine is not called and evaluated. In this situation, an action specified in the subroutine that manipulates a route’s characteristics is not implemented.

NOTE: If you specify a policy chain as a subroutine, the entire chain acts as a single subroutine. As with other chains, the action specified by the default policy is taken only when the entire chain does not accept or reject a route.

Figure 10: Routing Policy Subroutine Evaluation



Routing Policy Tests

After you have created a routing policy, you can use the test policy command to ensure that the policy produces the results that you expect before applying the policy in a live environment. This command determines whether the routes specified in your routing policy are accepted or rejected. The default action of the test policy command is accept.

NOTE: The default policy of the test policy command accepts all routes from all protocols. Test output can be misleading when you are evaluating protocol-specific conditions.

For example, if you define a policy for BGP that accepts routes of a specified prefix and apply it to BGP as an export policy, the BGP routes that match the prefix are advertised to the BGP peers. However, if you test the same policy using the test policy command, the test output might indicate that non-BGP routes have been accepted.

Supported Standards and Drafts

The JUNOS software supports the following standards and drafts:

RFC 1997, *BGP Communities A ttribute*

RFC 2439, *BGP Route Flap Damping*