BGP Configuration Using Peer Templates

The BGP Configuration Using Peer Templates feature introduces a new mechanism that groups distinct neighbor configurations for Border Gateway Protocol (BGP) neighbors that share common policies. This type of policy configuration has been traditionally configured with BGP peer groups. However, peer groups have certain limitations because peer group configuration is bound to update grouping and specific session characteristics. Configuration templates provide an alternative to peer group configuration and overcome some of the limitations of peer groups.

Feature History for the BGP Configuration Using Peer Templates Feature

Release	Modification
12.0(24)S	This feature was introduced.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This feature was integrated into Cisco IOS Release 12.2(27)SBC.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for BGP Configuration Using Peer Templates

BGP must be configured in your network.

Restrictions for BGP Configuration Using Peer Templates

The following restrictions apply to the BGP Configuration Using Peer Templates feature for this release:

- A peer policy template can inherit up to seven policy templates, and no more than eight policy templates can be applied to a BGP neighbor or peer.
- A peer session template can directly inherit only one session template. However, each inherited session template can also inherit one session template. So, a BGP neighbor can directly inherit only one session template and indirectly inherit up to seven additional session templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. In other words, a BGP neighbor can be configured only to belong to a peer group or to inherit policies from peer templates.

Information About BGP Configuration Using Peer Templates

To configure this feature, you must understand the following concepts:

- Peer Groups and BGP Update Generation, page 2
- BGP Dynamic Update Peer-Groups, page 3
- Peer Templates, page 3
- Inheritance in Peer Templates, page 4
- Peer Session Templates, page 4
- Peer Policy Templates, page 6

Peer Groups and BGP Update Generation

In previous versions of Cisco IOS software, BGP update messages were grouped based on peer group configurations. This method of grouping neighbors for BGP update message generation reduced the amount of system processing resources needed to scan the routing table. This method, however, had the following limitations:

- All neighbors that shared the same peer group configuration also had to share the same outbound routing policies.
- All neighbors had to belong to the same peer group and address family. Neighbors configured in different address-families could not belong to different peer groups.

These limitations existed to balance optimal update generation and replication against peer group configuration. These limitations also caused the network operator to configure smaller peer groups, which reduced the efficiency of update message generation and limited the scalability of neighbor configuration.

BGP Dynamic Update Peer-Groups

The introduction of the BGP Dynamic Update Peer-Groups feature separates peer-group configuration from update group generation. The BGP Dynamic Update Peer-Groups feature introduced an algorithm that dynamically calculates BGP update-group membership based on outbound routing policies. Optimal BGP update message generation occurs automatically and independently. BGP neighbor configuration is no longer restricted by outbound routing policies, and update-groups can belong to different address families.

Even though BGP update message generation has been separated from peer group configuration, peer group configuration still retains the following limitations:

- A neighbor can belong only to one peer group.
- Neighbors that belong to different address-families cannot belong to the same peer group.
- Different sets of policies cannot be grouped and applied to a neighbor.

To address the limitations of peer groups, the BGP Configuration Using Peer Templates feature was introduced along with the BGP Dynamic Update Peer-Groups feature. The BGP Configuration Using Peer Templates feature introduces a new mechanism called the peer template. For more information about the BGP Dynamic Update Peer-Groups feature, refer to the *BGP Dynamic Update Peer-Groups* document.

Peer Templates

A peer template is a configuration pattern that can be applied to neighbors that share common policies. Peer templates are reusable and support inheritance, which allows the network operator to group and apply distinct neighbor configurations for BGP neighbors that share common policies. Peer templates also allow the network operator to define very complex configuration patterns through the capability of a peer template to inherit a configuration from another peer template.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and Network Layer Reachability Information (NLRI) configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address-families and NLRI configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. With the configuration of the BGP Configuration Using Peer Templates feature and the support of the BGP Dynamic Update Peer-Groups feature, the network operator no longer needs to configure peer groups in BGP and can benefit from improved configuration flexibility and faster convergence.

Note Peer groups are still supported. The configuration of the BGP Configuration Using Peer Templates feature does not conflict with or restrict peer group configuration. However, a BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

Inheritance in Peer Templates

The inheritance capability is a key component of peer template operation. Inheritance in a peer template is similar to node and tree structures commonly found in general computing, for example, file and directory trees. A peer template can directly or indirectly inherit the configuration from another peer template. The directly inherited peer template represents the tree in the structure. The indirectly inherited peer template represents a node in the tree. Because each node also supports inheritance, branches can be created that apply the configurations of all indirectly inherited peer templates within a chain back to the directly inherited peer template or the source of the tree. This structure eliminates the need to repeat configuration statements that are commonly reapplied to groups of neighbors because common configuration statements can be applied once and then indirectly inherited by peer templates that are applied to neighbor groups with common configurations. Configuration statements that are duplicated separately within a node and a tree are filtered out at the source of the tree by the directly inherited template. A directly inherited template will overwrite any indirectly inherited statements that are duplicated in the directly inherited template.

Inheritance expands the scalability and flexibility of neighbor configuration by allowing you to chain together peer templates configurations to create simple configurations that inherit common configuration statements or complex configurations that apply very specific configuration statements along with common inherited configurations. Specific details about configuring inheritance in peer session templates and peer policy templates are provided in the following sections.

Peer Session Templates

Peer session templates are used to group and apply the configuration of general session commands to groups of neighbors that share common session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template. The following general session commands are supported by peer session templates:

- **description**
- **disable-connected-check**
- **ebgp-multihop**
- **exit peer-session**
- **inherit peer-session**
- **local-as**
- **password**
- **remote-as**
- **shutdown**
- **timers**
- **translate-update**
- **update-source**
- **version**

The BGP Configuration Using Peer Templates feature introduces peer session configuration mode and the following peer session commands:

- **exit peer-session**—This command is used to enter session-template configuration mode and to create a peer session template.
- **inherit peer-policy**—This command is used to configure a peer session template to inherit a configuration from another peer session template. A peer session template can directly inherit only one peer session template. However, each inherited session template can also contain one inherited template.
- **template peer-policy**—This command is used to exit session-template configuration mode and enter router configuration mode.
- **neighbor inherit peer-session**—This command is used to send locally configured peer session templates to the specified neighbor. If the locally configured peer session template is configured to inherit configurations from other peer session templates, the specified neighbor will also indirectly inherit these configurations from the other peer session templates. This command sends only general session configuration information.
- **show ip bgp template peer-session**—This command is used to display locally configured peer session templates. The output can be filtered to display a specific peer session template. By default, it will display all peer session templates.

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template.

Note If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

This behavior allows a BGP neighbor to directly inherit only one session template and indirectly inherit up to seven additional peer session templates. This allows you to apply up to a maximum of eight peer session configurations to a neighbor: the configuration from the directly inherited peer session template and the configurations from up to seven indirectly inherited peer session templates. Inherited peer session configurations are evaluated first and applied starting with the last node in the branch and ending with the directly applied peer session template configuration at the of the source of the tree. The directly applied peer session template will have priority over inherited peer session template configurations. Any configuration statements that are duplicated in inherited peer session templates will be overwritten by the directly applied peer session template. So, if a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template. The following examples illustrate the use of this feature.

In the following example, the general session command **remote-as 1** is applied in the peer session template named SESSION-TEMPLATE-ONE:

```
template peer-session SESSION-TEMPLATE-ONe
 remote-as 1
 exit peer-session
```

Another peer session template named SESSION-TEMPLATE-TWO is configured to inherit the configuration from SESSION-TEMPLATE-ONE and apply the **remote-as 2** command.

```
template peer-session SESSION-TEMPLATE-TWO
 remote-as 2
 inherit peer-session SESSION-TEMPLATE-ONE
 exit peer-session
```

The template named “session-template-one” is indirectly inherited by the template named “peer-session-two”, and the **remote-as** command is applied in each template but with a different autonomous system number. The **remote-as 1** command statement is evaluated first. But, since the **remote-as 2** command statement was directly applied in “session-template-two”, the command in “session-template-two” will have priority over the command from “session-template-one”, and the **remote-as 2** command will be applied instead of the **remote-as 1** command.

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

Peer Policy Templates

Peer policy templates are used to group and apply the configuration of commands that are applied within specific address-families and NLRI configuration mode. Peer policy templates are created and configured in peer policy configuration mode. BGP policy commands that are configured for specific address-families or NLRI configuration modes are configured in a peer policy template. The following BGP policy commands are supported by peer policy templates:

- **advertisement-interval**
- **allowas-in**
- **as-override**
- **capability**
- **default-originate**
- **distribute-list**
- **dmzlink-bw**
- **exit-peer-policy**
- **filter-list**
- **inherit peer-policy**
- **maximum-prefix**
- **next-hop-self**
- **next-hop-unchanged**
- **prefix-list**
- **remove-private-as**
- **route-map**
- **route-reflector-client**
- **send-community**
- **send-label**

- **soft-reconfiguration**
- **unsuppress-map**
- **weight**

The BGP Configuration Using Peer Templates feature introduces peer policy configuration mode and the following peer policy commands:

- **debug ip bgp groups**—This command is used to enter policy-template configuration mode and to create a peer policy template.
- **Glossary**—This command is used to configure a peer policy template to inherit a configuration from another peer policy template. A peer policy template can indirectly inherit up to seven peer policy template configurations.
- **Glossary**—This command is used to exit policy-template configuration mode and enter router configuration mode.
- **neighbor inherit peer-policy**—This command is used to send locally configured peer policy templates to the specified neighbor. If the peer policy template is configured to inherit configurations from other peer policy templates, the specified neighbor will also indirectly inherit these configurations from the other peer policy templates. This command sends BGP policy configuration information that is applied for specific address-families and NLRI configuration modes. This command does not support general session information.
- **show ip bgp template peer-policy**—This command is used to display locally configured peer policy templates. The output can be filtered to display a specific peer policy template. By default, it will display all peer policy templates.

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address-families and NLRI configuration modes. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.

Like peer session templates, a peer policy template supports inheritance. However, there are minor differences. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group. Inherited peer policy templates are configured with sequence numbers like route-maps. An inherited peer policy template, like a route-map, is evaluated starting with the inherit statement with the lowest sequence number and ending with the highest sequence number. However, there is a difference; a peer policy template will not fall through like a route-map. Every sequence is evaluated, and if a BGP policy command is reapplied with a different value, it will overwrite any previous value from a lower sequence number.

The following peer policy template configuration examples illustrate this behavior. The examples show the configuration of three peer policy templates named ONE, TWO, and THREE. Each template applies the **weight** command with a different value, and each subsequent template has a higher sequence number than the previous template.

Peer policy template POLICY-TEMPLATE-ONE is configured with the sequence number 10. It is processed first because it has the lowest sequence number. It applies a weight of 1000.

```
template peer-policy POLICY-TEMPLATE-ONE 10
 weight 1000
 exit peer-policy
```

Peer policy template POLICY-TEMPLATE-TWO is processed second because it has the next highest sequence number. It inherits the configuration from peer policy template POLICY-TEMPLATE-ONE and applies a weight of 1000. However, this template is also configured to apply a weight of 2000. So the value of 2000 will overwrite the value inherited from peer policy template POLICY-TEMPLATE-ONE.

```
template peer-policy POLICY-TEMPLATE-TWO 20
 inherit peer-policy POLICY-TEMPLATE-ONE
 weight 2000
 exit peer-policy
```

Finally, peer policy template POLICY-TEMPLATE-THREE inherits the configurations from peer policy template ONE and TWO. But because peer policy template POLICY-TEMPLATE-THREE is also configured to apply a weight, the weight of 3000 overwrites the weight value of 2000 that was inherited from peer policy template POLICY-TEMPLATE-TWO.

```
Template peer-policy POLICY-TEMPLATE-THREE 30
 inherit peer-policy POLICY-TEMPLATE-TWO
 weight 3000
 exit peer-policy
```

The directly applied peer policy template and the inherit statement with the highest sequence number will always have priority and be applied last. Commands that are reapplied in subsequent peer templates will always overwrite the previous values. This behavior is designed to allow you to apply common policy configurations to large neighbor groups and specific policy configurations only to certain neighbors and neighbor groups without duplicating individual policy configuration commands.

Peer policy templates support only policy configuration commands. BGP policy configuration commands that are configured only for specific address families or NLRI configuration modes are configured with peer policy templates.

How to Configure BGP Using Peer Templates

This section contains the following procedures:

Configuring Peer Session Templates

This task creates and configures a peer session template.

Peer Session Templates Simplify BGP Configuration

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from peer session templates. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Note The commands in Step 5 and 6 are optional and could be replaced with any supported general session commands.

Restrictions

The following restrictions apply to the peer session templates:

- A peer session template can directly inherit only one session template, and each inherited session template can also contain one indirectly inherited session template. So, a neighbor or neighbor group can be configured with only one directly applied peer session template and seven additional indirectly inherited peer session templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to only inherit policies from peer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **template peer-session** *session-template-name*
5. **remote-as** *as-number*
6. **timers** *keepalive-interval hold-time*
7. **exit peer-session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session INTERNAL-BGP`	Enters session-template configuration mode and creates a peer session template.
Step 5	**remote-as** *as-number* **Example:** `Router(config-router-stmp)# remote-as 202`	Configures peering with a remote neighbor in the specified autonomous system.
Step 6	**timers** *keepalive-interval hold-time* **Example:** `Router(config-router-stmp)# timers 30 300`	Configures BGP keepalive and hold timers. • The hold time must be at least twice the keepalive time.
Step 7	**exit peer-session** **Example:** `Router(config-router-stmp)# exit peer-session`	Exits session-template configuration mode and enters router configuration mode.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

Configuring Peer Session Template Inheritance with the inherit peer-session Command

This task configures peer session template inheritance with the **inherit peer-session** command. It creates and configures a peer session template and allows it to inherit a configuration from another peer session template.

Note The commands in Steps 5 and 6 are optional and could be replaced with any supported general session commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **template peer-session** *session-template-name*
5. **description** *text-string*
6. **update-source** *interface-type interface-number*
7. **inherit peer-session** *session-template-name*
8. **exit peer-session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** as-*number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session CORE1`	Enter session-template configuration mode and creates a peer session template.

	Command or Action	Purpose
Step 5	**description** *text-string* **Example:** `Router(config-router-stmp)# description CORE-123`	Configures a description. • The text string can be up to 80 characters.
Step 6	**update-source** *interface-type interface-number* **Example:** `Router(config-router-stmp)# update-source loopback 1`	Configures a router to select a specific source or interface to receive routing table updates. • The example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface.
Step 7	**inherit peer-session** *session-template-name* **Example:** `Router(config-router-stmp)# inherit peer-session INTERNAL-BGP`	Configures this peer session template to inherit the configuration of another peer session template. • The example configures this peer session template to inherit the configuration from INTERNAL-BGP. This template can be applied to a neighbor, and the configuration INTERNAL-BGP will be applied indirectly. No additional peer session templates can be directly applied. However, the directly inherited template can contain up to seven indirectly inherited peer session templates.
Step 8	**exit peer-session** **Example:** `Router(config-router-stmp)# exit peer-session`	Exits session-template configuration mode and enters router configuration mode.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

Configuring Peer Session Template Inheritance with the neighbor inherit peer-session Command

This task configures a router to send a peer session template to a neighbor to inherit the configuration from the specified peer session template with the **neighbor inherit peer-session** command. Use the following steps to send a peer session template configuration to a neighbor to inherit:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **neighbor** *ip-address* **inherit peer-session** *session-template-name*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 172.16.0.1 remote-as 202`	Configures a peering session with the specified neighbor. • The explicit remote-as statement is required for the neighbor inherit statement in Step 5 to work. If a peering is not configured, the specified neighbor in Step 5 will not accept the session template.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **inherit peer-session** *session-template-name* **Example:** `Router(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1`	Sends a peer session template to a neighbor so that the neighbor can inherit the configuration. • The example configures a router to send the peer session template named CORE1 to the 172.16.0.1 neighbor to inherit. This template can be applied to a neighbor, and if another peer session template is indirectly inherited in CORE1, the indirectly inherited configuration will also be applied. No additional peer session templates can be directly applied. However, the directly inherited template can also inherit up to seven additional indirectly inherited peer session templates.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

Verifying Peer Session Templates

This task verifies the configuration of peer session templates.

SUMMARY STEPS

1. **show ip bgp template peer-session** *session-template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip bgp template peer-session** [session-*template-name*] **Example:** `Router# show ip bgp template peer-session`	Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the *peer-session-name* argument. This command also supports all standard output modifiers.

Configuring Peer Policy Templates

This task creates and configures a peer policy template.

Peer Policy Templates Simplify and Improve the Flexibility of BGP Configuration

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address-families and NLRI configuration modes. Peer policy templates can be configured in address family and NLRI configuration modes. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. Once the router is in peer policy template mode, you can configure any address family-specific command. The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, very specific and very complex BGP policies can also be created.

Note The commands in Steps 5 through 8 are optional and could be replaced with any supported BGP policy configuration commands.

Restrictions

The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to only inherit policies from peer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **template peer-policy** *policy-template-name*
5. **send-community** [**both** | **extended** | **standard**]
6. **maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**]
7. **weight** *weight-value*
8. **prefix-list** *prefix-list-name* {**in** | **out**}
9. **exit peer-policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy GLOBAL`	Enter policy-template configuration mode and creates a peer policy template.
Step 5	**send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-ptmp)# send community`	Configures the router to send the community attribute or the extended community attribute or both the community and extended community attribute.
Step 6	**maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**] **Example:** `Router(config-router-ptmp)# maximum-prefix 10000`	Configures the maximum number of prefixes that a neighbor will accept from this peer.
Step 7	**weight** *weight-value* **Example:** `Router(config-router-ptmp)# weight 300`	Sets the default weight for routes that are sent from this neighbor.
Step 8	**prefix-list** *prefix-list-name* {**in** | **out**} **Example:** `Router(config-router-ptmp)# prefix-list NO-MARTIAN in`	Filters prefixes that are received by the router or sent from the router. • The prefix-list in the example filters inbound internal addresses.
Step 9	**exit peer-policy** **Example:** `Router(config-router-ptmp)# exit peer-policy`	Exits policy-template configuration mode and enters router configuration mode.

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template with the **inherit peer-policy** or **neighbor inherit peer-policy** command.

Configuring Peer Policy Template Inheritance with the inherit peer-policy Command

This task configures peer policy template inheritance with the **inherit peer-policy** command. It creates and configure a peer policy template and allows it to inherit a configuration from another peer policy template.

Note The commands in Steps 5 and 6 are optional and could be replaced with any supported BGP policy configuration commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **template peer-policy** *policy-template-name*
5. **route-map** *map-name* {**in** | **out**}
6. **filter-list** *as-path-list* {**in** | **out**}
7. **inherit peer-policy** *policy-template-name sequence-number*
8. **exit peer-policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** as-*number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy NETWORK-A`	Enter policy-template configuration mode and creates a peer policy template.
Step 5	**route-map** *map-name* {**in** | **out**} **Example:** `Router(config-router-ptmp)# route-map SET-COMMUNITY in`	Applies the specified route-map to inbound or outbound routes.
Step 6	**filter-list** *as-path-list* {**in** | **out**} **Example:** `Router(config-router-ptmp)# filter-list 202 in`	Creates a filter-list and applies it to inbound or outbound routes.
Step 7	**inherit peer-policy** *policy-template-name sequence-number* **Example:** `Router(config-router-ptmp)# inherit peer-policy GLOBAL 10`	Configures the peer policy template to inherit the configuration of another peer policy template. • The *sequence-number* argument sets the order in which the peer policy template is evaluated. Like a route-map sequence number, the lowest sequence number is evaluated first. • The example configures this peer policy template to inherit the configuration from GLOBAL. If the template created in these steps is applied to a neighbor, the configuration GLOBAL will also be inherited and applied indirectly. Up to six additional peer policy templates can be indirectly inherited from GLOBAL for a total of eight directly applied and indirectly inherited peer policy templates. • This template in the example will be evaluated first if no other templates are configured with a lower sequence number.
Step 8	**exit peer-policy** **Example:** `Router(config-router-ptmp)# exit peer-policy`	Exits policy-template configuration mode and enters router configuration mode.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-policy** or **neighbor inherit peer-policy** command.

Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command

This task configures a router to send a peer policy template to a neighbor to inherit with the **neighbor inherit peer-policy** command. Use the following steps to send a peer policy template configuration to a neighbor to inherit:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **address-family** {**ipv4** | **ipv6** | **vpnv4** [**multicast** | **unicast** | **vrf** *vrf-name*]}
6. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 202`	Configures a peering session with the specified neighbor. • The explicit remote-as statement is required for the neighbor inherit statement in Step 5 to work. If a peering is not configured, the specified neighbor in Step 5 will not accept the session template.
Step 5	**address-family** {**ipv4** | **ipv6** |**vpnv4** [**multicast** | **unicast** | **vrf** *vrf-name*]} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure a neighbor to accept address family-specific command configurations.

	Command or Action	Purpose
Step 6	**neighbor** *ip-address* **inherit peer-policy** *session-template-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 inherit peer-policy GLOBAL`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration. • The example configures a router to send the peer policy template named GLOBAL to the 10.0.0.1 neighbor to inherit. This template can be applied to a neighbor, and if another peer policy template is indirectly inherited from GLOBAL, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from GLOBAL.
Step 7	**exit** **Example:** `Router(config-router)# exit`	Exits address family configuration mode and enters router configuration mode.

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template with the **inherit peer-policy** or **neighbor inherit peer-policy** command.

Verifying Peer Policy Templates

This task verifies the configuration of peer policy templates.

SUMMARY STEPS

1. **show ip bgp template peer-policy** [*policy-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip bgp template peer-policy** [*policy-template-name*] **Example:** `Router# show ip bgp template peer-policy`	Displays locally configured peer policy templates. • The output can be filtered to display a single peer policy template with the *peer-policy-name* argument. This command also supports all standard output modifiers.

Configuration Examples for BGP Configuration Using Peer Templates

This section contains the following peer session and peer policy template configuration and verification examples:

Peer Session Template Configuration Examples

The following example creates a peer session template named INTERNAL-BGP in session-template configuration mode:

```
Router(config)# router bgp 101
Router(config-router)# template peer-session INTERNAL-BGP
Router(config-router-stmp)# remote-as 202
Router(config-router-stmp)# timers 30 300
Router(config-router-stmp)# exit-peer-session
Router(config-router)#
```

The following example creates a peer session template named CORE1. This example inherits the configuration of the peer session template named INTERNAL-BGP.

```
Router(config-router)# template peer-session CORE1
Router(config-router-stmp)# description CORE-123
Router(config-router-stmp)# update-source loopback 1
Router(config-router-stmp)# inherit peer-session INTERNAL-BGP
Router(config-router-stmp)# exit-peer-session
Router(config-router)#
```

The following example configures the 172.16.0.1 neighbor to inherit the CORE1 peer session template. The 172.16.0.1 neighbor will aso indirectly inherit the configuration from the peer session template named INTERNAL-BGP. The explicit remote-as statement is required for the neighbor inherit statement to work. If a peering is not configured, the specified neighbor will not accept the session template.

```
Router(config)# router bgp 101
Router(config-router)# neighbor 172.16.0.1 remote-as 202
Router(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1
Router(config-router)#
```

Peer Session Template Verification Examples

The **show ip bgp template peer-session** command is used to verify the configuration of local peer session templates. The following example shows the peer session templates named INTERNAL-BGP and CORE1. The output also shows that INTERNAL-BGP is inherited by CORE1.

```
Router# show ip bgp template peer-session

Template:INTERNAL-BGP, index:1
Local policies:0x21, Inherited polices:0x0
 *Inherited by Template CORE1, index= 2
Locally configured session commands:
 remote-as 202
 timers 30 300
Inherited session commands:

Template:CORE1, index:2
Local policies:0x180, Inherited polices:0x21
This template inherits:
  INTERNAL-BGP index:1 flags:0x0
Locally configured session commands:
 update-source loopback 1
```

```
 description CORE-123
Inherited session commands:
 remote-as 202
 timers 30 300
```

The **show running-config** command can be used to verify the local configuration of peer session templates and peer session templates that are configured to be sent to neighbors. The following ouput is filtered to show only the relevant portion of the configuration. The following output shows that the router is configured to send the peer session template named CORE1 to the 172.16.0.1 neighbor:

```
Router# show running-config | begin BGP

 template peer-session INTERNAL-BGP
  remote-as 202
  timers 30 300
 exit-peer-session
 !
 template peer-session CORE1
  description CORE-123
  update-source Loopback1
  inherit peer-session INTERNAL-BGP
 exit-peer-session
 !
 no synchronization
 bgp log-neighbor-changes
 neighbor 172.16.0.1 inherit peer-session CORE1
 no auto-summary
!
ip default-gateway 192.168.0.1
ip classless
ip route 192.168.0.0 255.255.0.0 192.168.0.1
ip route 172.16.1.1 255.255.255.255 192.168.0.1
!
```

Peer Policy Template Configuration Examples

The following example creates a peer policy template named GLOBAL in policy-template configuration mode:

```
Router(config)# router bgp 101
Router(config-router)# template peer-policy GLOBAL
Router(config-router-ptmp)# send-community
Router(config-router-ptmp)# weight 1000
Router(config-router-ptmp)# maximum-prefix 10000
Router(config-router-ptmp)# prefix-list no-martian in
Router(config-router-ptmp)# exit-peer-policy
Router(config-router)#
```

The following example creates a peer policy template named PRIMARY-IN in policy-template configuration mode:

```
Router(config-router)# template peer-policy PRIMARY-IN
Router(config-router-ptmp)# prefix-list ALLOW-PRIMARY-A in
Router(config-router-ptmp)# route-map SET-LOCAL in
Router(config-router-ptmp)# weight 2345
Router(config-router-ptmp)# default-originate
Router(config-router-ptmp)# exit-peer-policy
Router(config-router)#
```

The following example creates a peer policy template named CUSTOMER-A. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
Router(config-router)# template peer-policy CUSTOMER-A
Router(config-router-ptmp)# route-map SET-COMMUNITY in
Router(config-router-ptmp)# filter-list 20 in
Router(config-router-ptmp)# inherit peer-policy PRIMARY-IN 20
Router(config-router-ptmp)# inherit peer-policy GLOBAL 10
Router(config-router-ptmp)# exit-peer-policy
Router(config-router)#
```

The following example configures the 10.0.0.1 neighbor in address family mode to inherit the peer policy template name CUSTOMER-A. The 10.0.0.1 neighbor will also indirectly inherit the peer policy templates named PRIMARY-IN and GLOBAL.

```
Router(config)# router bgp 101
Router(config-router)# neighbor 10.0.0.1 remote-as 202
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)# neighbor 10.0.0.1 inherit peer-policy CUSTOMER-A
Router(config-router-af)# exit
```

Peer Policy Template Verification Examples

The **show ip bgp template peer-policy** command is used to verify the configuration of local peer policy templates. The following example shows the peer policy templates named GLOBAL, PRIMARY-IN, and CUSTOMER-A. The output also shows that GLOBAL and PRIMARY-IN are inherited by CUSTOMER-A.

```
Router# show ip bgp template peer-policy

Template:GLOBAL, index:1.
Local policies:0x88840, Inherited polices:0x0
 *Inherited by Template CUSTOMER-A, index:3
Locally configured policies:
  prefix-list no-martian in
  weight 1000
  send-community
  maximum-prefix 10000
Inherited policies:

Template:PRIMARY-IN, index:2.
Local policies:0xC41, Inherited polices:0x0
 *Inherited by Template CUSTOMER-A, index:3
Locally configured policies:
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  default-originate route-map none
  weight 2345
Inherited policies:

Template:CUSTOMER-A, index:3.
Local policies:0x5, Inherited polices:0x88C40
This template inherits:
  PRIMARY-IN, index:2, seq_no:20, flags:0x5
  GLOBAL, index:1, seq_no:10, flags:0xC45
Locally configured policies:
  filter-list 20 in
  route-map SET-COMMUNITY in
Inherited policies:
  prefix-list ALLOW-PRIMARY-A in
  default-originate route-map none
```

```
  weight 2345
  send-community
  maximum-prefix 10000
```

The **show running-config** command can be used to verify the local configuration of peer policy templates and peer policy templates that are configured to be sent to neighbors. The following ouput is filtered to show only the relevant portion of the configuration. The following output shows that the router is configured to send the peer policy template named CUSTOMER-A to the 10.0.0.1 neighbor.

```
Router# show running-config | begin bgp

router bgp 101
 template peer-policy GLOBAL
  prefix-list no-martian in
  weight 1000
  maximum-prefix 10000
  send-community
 exit-peer-policy
 !
 template peer-policy PRIMARY-IN
  route-map SET-LOCAL in
  prefix-list ALLOW-PRIMARY-A in
  default-originate
  weight 2345
 exit-peer-policy
 !
 template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
 exit-peer-policy
 !
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 202
 neighbor 10.0.0.1 inherit peer-policy CUSTOMER-A
 no auto-summary
!
ip default-gateway 192.168.0.1
ip classless
ip route 192.168.0.0 255.255.0.0 192.168.0.1
ip route 172.16.1.1 255.255.255.255 192.168.0.1
!
```

Where to Go Next

For information about the BGP Dynamic Update Peer-Groups feature, refer to the *BGP Dynamic Update Peer-Groups* feature module.

Additional References

The following sections provide references related to the BGP Configuration Using Peer Templates feature.

Related Documents

Related Topic	Document Title
BGP commands	Cisco IOS Release 12.0 *Network Protocols Command Reference, Part 1* *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2 *Cisco IOS IP Command Reference, Volume 2 of 4: Routing Protocols*, Release 12.3T
BGP configuration tasks	Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1* Cisco IOS Release 12.0 *Cisco IOS IP Configuration Guide*
BGP dynamic update group information	*BGP Dynamic Update Peer-Groups* Cisco IOS Release 12.3(4)T.

Standards

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

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	TAC Home Page: http://www.cisco.com/public/support/tac/home.shtml BGP Support Page: http://www.cisco.com/cgi-bin/Support/browse/psp_view.pl?p=Internetworking:BGP

Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

Peer Session Commands

- **exit peer-session**
- **inherit peer-session**
- **show ip bgp template peer-session**

Peer Policy Commands

- **inherit peer-policy**
- **template peer-policy**
- **show ip bgp template peer-policy**

Debug Command

- **debug ip bgp groups**

exit peer-session

To exit session-template configuration mode and enter router configuration mode, use the **exit peer-session** command in session-template configuration mode.

exit peer-session

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Session-template configuration

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

This command is used to exit session-template configuration mode.

Examples

In the following example, the router is configured to exit session-template configuration mode and enter router configuration mode:

```
Router(config-router-stmp)# exit-peer-session
Router(config-router)#
```

Related Commands

Command	Description
template peer-session	Creates a peer session template and enters session-template configuration mode.

inherit peer-session

To configure a peer session template to inherit the configuration from another peer session template, use the **inherit peer-session** command in session-template configuration mode. To remove an inherit statement from a peer session template, use the **no** form of this command.

inherit peer-session *template-name*

no inherit peer-session *template-name*

Syntax Description

template-name	Name of the peer session template to inherit.

Defaults

No inherit statements are configured.

Command Modes

Session-template configuration

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

The **inherit peer-session** command is used to configure a peer session template to inherit the configuration of another peer session template. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template. However, each indirectly inherited session template can also contain an indirectly inherited template. So, a peer can directly inherit only one peer session template and indirectly inherit up to seven additional indirectly inherited peer session templates, allowing you to apply up to a maximum of eight inherited peer session configurations.

Note If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

Indirectly inherited peer session templates are evaluated first, and the directly applied (locally configured) peer session template is evaluated last. If a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template. In other words, an overlapping statement from a local configuration will override the statement from the inherited configuration.

Examples

In the following example, a peer session template named CORE1 is created. This example inherits the configuration of the peer session template named INTERNAL-BGP.

```
Router(config-router)# template peer-session CORE1
Router(config-router-stmp)# description CORE-123
Router(config-router-stmp)# update-source loopback 1
Router(config-router-stmp)# inherit peer-session INTERNAL-BGP
Router(config-router-stmp)# exit-peer-session
Router(config-router)#
```

Related Commands

Command	Description
exit peer-session	Exits session-template configuration mode and enters router configuration mode.
neighbor inherit peer-session	Configures a router to send a peer session template to a neighbor so that the neighbor can inherit the configuration.
show ip bgp template peer-session	Displays locally configured peer session templates.
template peer-session	Creates a peer session template and enters session-template configuration mode.

show ip bgp template peer-session

To display peer policy template configurations, use the **show ip bgp template peer-session** command in user EXEC mode.

show ip bgp template peer-session [*session-template-name*]

Syntax Description

session-template-name	(Optional) Name of a locally configured peer session template.

Defaults

If a peer session template is not specified with the *session-template-name* argument, all peer session templates will be displayed.

Command Modes

User EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

This command is used to display locally configured peer session templates. The output can be filtered to display a single peer session template with the *peer-session-name* argument. This command also supports all standard output modifiers.

Examples

The **show ip bgp template peer-session** command is used to verify the configuration of local peer session templates. The following example shows the peer session templates named INTERNAL-BGP and CORE1. The output also shows that INTERNAL-BGP is inherited by CORE1.

```
Router# show ip bgp template peer-session

Template:INTERNAL-BGP, index:1
Local policies:0x21, Inherited polices:0x0
 *Inherited by Template CORE1, index= 2
Locally configured session commands:
 remote-as 202
 timers 30 300
Inherited session commands:

Template:CORE1, index:2
Local policies:0x180, Inherited polices:0x21
This template inherits:
  INTERNAL-BGP index:1 flags:0x0
Locally configured session commands:
 update-source loopback 1
 description CORE-123
Inherited session commands:
```

```
remote-as 202
timers 30 300
```

Table 1 describes the significant fields shown in the display.

Table 1 show ip bgp template peer-session Field Descriptions

Field	Description
Template:	Name of the peer template.
index:	The sequence number in which the displayed template is processed.
Local policies:	Displays the hexadecimal value of locally configured policies.
Inherited polices:	Displays the hexadecimal value of inherited policies. The 0x0 value is displayed when no templates are inherited.
Locally configured session commands:	Displays a list of commands that are locally configured in a peer template.
Inherited session commands:	Displays a list of commands that are inherited from a peer session template.

Related Commands

Command	Description
inherit peer-session	Configures a peer session template to inherit the configuration from another peer session template.
template peer-session	Creates a peer session template and enters session-template configuration mode.

inherit peer-policy

To configure a peer policy template to inherit the configuration from another peer policy template, use the **inherit peer-policy** command in policy-template configuration mode. To remove an inherit statement from a peer policy template, use the **no** form of this command.

inherit peer-policy *policy-template sequence-number*

no inherit peer-policy *policy-template sequence-number*

Syntax Description

peer-policy	Name of the peer policy template to be inherited.
sequence-number	Sequence number that sets the order in which the peer policy template is evaluated. Like a route-map sequence number, the lowest sequence number is evaluated first.

Defaults

No inherit statements are configured.

Command Modes

Policy-template configuration

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

The **inherit peer-policy** command is used to configure a peer policy template to inherit the configuration of another peer policy template. Peer policy templates support inheritance and a peer can directly and indirectly inherit up to seven peer policy templates. Inherited peer policy templates are configured with sequence numbers like route maps. An inherited peer policy template, like a route map, is evaluated starting with the inherit statement with the lowest sequence number. However, peer policy templates do not fall through. Every sequence is evaluated. If a BGP policy command is reapplied with a different value, it will overwrite any previous value from a lower sequence number.

Note A Border Gateway Protocol (BGP) routing process cannot be configured to be a member of a peer group and to use peer templates for group configurations. You must use one method or the other. We recommend peer templates because they provide improved performance and scalability.

Examples

In the following example, a peer policy template named CUSTOMER-A is created. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
Router(config-router)# template peer-policy CUSTOMER-A
Router(config-router-ptmp)# route-map SET-COMMUNITY in
Router(config-router-ptmp)# filter-list 20 in
Router(config-router-ptmp)# inherit peer-policy PRIMARY-IN 20
Router(config-router-ptmp)# inherit peer-policy GLOBAL 10
Router(config-router-ptmp)# exit-peer-policy
Router(config-router)#
```

Related Commands

Command	Description
exit peer-policy	Exits policy-template configuration mode and enters router configuration mode.
neighbor inherit peer-policy	Configures a router to send a peer policy template to a neighbor so that the neighbor can inherit the configuration.
show ip bgp template peer-policy	Displays locally configured peer policy templates.
template peer-policy	Creates a peer policy template and enters policy-template configuration mode.

template peer-policy

To create a peer policy template and enter policy-template configuration mode, use the **template peer-policy** command in router configuration mode. To remove a peer policy template, use the **no** form of this command.

template peer-policy *policy-template-name*

no template peer-policy *policy-template-name*

Syntax Description

policy-template-name	Name or tag for the peer policy template.

Defaults

Removing a peer policy template by using the **no** form of this command removes all policy configurations inside of the template.

Command Modes

Router configuration

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

Peer policy templates are used to group and apply the configuration of commands that are applied within specific address-families and NLRI configuration mode. Peer policy templates are created and configured in peer policy configuration mode. BGP policy commands that are configured for specific address-families or NLRI configuration modes are configured in a peer policy template. The following BGP policy commands are supported by peer policy templates:

- **advertisement-interval**
- **allowas-in**
- **as-override**
- **capability**
- **default-originate**
- **distribute-list**
- **dmzlink-bw**
- **exit-peer-policy**
- **filter-list**
- **inherit peer-policy**
- **maximum-prefix**

- **next-hop-self**
- **next-hop-unchanged**
- **prefix-list**
- **remove-private-as**
- **route-map**
- **route-reflector-client**
- **send-community**
- **send-label**
- **soft-reconfiguration**
- **unsuppress-map**
- **weight**

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address-families and NLRI configuration modes. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.

Peer policy templates support direct and indirect inheritance from up to eight peer policy templates. Inherited peer policy templates are configured with sequence numbers like route-maps. An inherited peer policy template, like a route-map, is evaluated starting with the inherit statement with the lowest sequence number and ending with the highest sequence number. However, there is a difference; a peer policy template will not fall through like a route-map. Every sequence is evaluated, and if a BGP policy command is reapplied with different value, it will overwrite any previous value from a lower sequence number.

Peer policy templates support only general policy commands. BGP policy configuration commands that are configured only for specific address families or NLRI configuration modes are configured with peer policy templates.

Note A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from only peer templates.

Examples

The following example creates a peer policy template named CUSTOMER-A. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
Router(config-router)# template peer-policy CUSTOMER-A
Router(config-router-ptmp)# route-map SET-COMMUNITY in
Router(config-router-ptmp)# filter-list 20 in
Router(config-router-ptmp)# inherit peer-policy PRIMARY-IN 20
Router(config-router-ptmp)# inherit peer-policy GLOBAL 10
Router(config-router-ptmp)# exit-peer-policy
Router(config-router)#
```

Related Commands

Command	Description
advertisement-interval	Sets the minimum interval between the sending of BGP routing updates.
allowas-in	Configures PE routers to allow readvertisement of all prefixes containing duplicate autonomous system numbers.
as-override	Configures a PE router to override the ASN of a site with the ASN of a provider.
capability orf prefix-list	Configures outbound route filtering and advertises the capability to send and receive ORF updates to the neighbor routers.
default-originate	Originates a default route to the local router.
distribute-list	Distributes BGP neighbor information as specified in an access list.
dmzlink-bw	Advertises the bandwidth of links that are used to exit an autonomous system.
exit peer-policy	Exits policy-template configuration mode and enters router configuration mode.
filter-list	Sets up a BGP filter.
inherit peer-policy	Configures a peer policy template to inherit the configuration from another peer policy template.
maximum-prefix	Controls how many prefixes can be received from a neighbor.
neighbor inherit peer-policy	Configures a router to send a peer policy template to a neighbor so that the neighbor can inherit the configuration.
neighbor send-label	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.
next-hop-self	Disables next-hop processing of BGP updates on the router.
next-hop-unchanged	Propagates the next- hop unchanged for iBGP paths to this router.
prefix-list	Specifies a prefix list, a CLNS filter set, or a CLNS filter expression to be used to filter BGP advertisements.
remove-private-as	Removes the private autonomous system number from outbound routing updates.
route-map	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.
route-reflector-client	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
send-community	Specifies that the BGP community attribute should be sent to the specified neighbor.
show ip bgp template peer-policy	Displays locally configured peer policy templates.
show ip bgp template peer-session	Displays locally configured peer session templates.
soft-reconfiguration	Configures the Cisco IOS software to start storing updates.
template peer-session	Creates a peer session template and enters session-template configuration mode.
unsuppress-map	Selectively unsuppresses surpressed routes.
weight	Assigns a weight to a neighbor connection.

show ip bgp template peer-policy

To display locally configured peer policy templates, use the **show ip bgp template peer-policy** command in user or privileged EXEC modes.

show ip bgp template peer-policy [*policy-template-name*]

Syntax Description

policy-template-name	(Optional) Name of a locally configured peer policy template.

Defaults

If a peer policy template is not specified with the *policy-template-name* argument, all peer policy templates will be displayed.

Command Modes

User or Privileged EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

This command is used to display locally configured peer policy templates. The output can be filtered to display a single peer policy template with the *policy-template-name* argument. This command also supports all standard output modifiers.

Examples

The **show ip bgp template peer-policy** command is used to verify the configuration of local peer policy templates. The following sample output shows the peer policy templates named GLOBAL, PRIMARY-IN, and CUSTOMER-A. The output also shows that GLOBAL and PRIMARY-IN are inherited by CUSTOMER-A.

```
Router# show ip bgp template peer-policy

Template:GLOBAL, index:1.
Local policies:0x88840, Inherited polices:0x0
 *Inherited by Template CUSTOMER-A, index:3
Locally configured policies:
  prefix-list no-martian in
  weight 1000
  send-community
  maximum-prefix 10000
Inherited policies:

Template:PRIMARY-IN, index:2.
Local policies:0xC41, Inherited polices:0x0
 *Inherited by Template CUSTOMER-A, index:3
Locally configured policies:
  prefix-list ALLOW-PRIMARY-A in
```

```
  route-map SET-LOCAL in
  default-originate route-map none
  weight 2345
Inherited policies:

Template:CUSTOMER-A, index:3.
Local policies:0x5, Inherited polices:0x88C40
This template inherits:
  PRIMARY-IN, index:2, seq_no:20, flags:0x5
  GLOBAL, index:1, seq_no:10, flags:0xC45
Locally configured policies:
  filter-list 20 in
  route-map SET-COMMUNITY in
Inherited policies:
  prefix-list ALLOW-PRIMARY-A in
  default-originate route-map none
  weight 2345
  send-community
  maximum-prefix 10000
```

Table 2 describes the significant fields shown in the display.

Table 2 show ip bgp template peer-policy Field Descriptions

Field	Description
Template:	Name of the peer template.
index:	The sequence number in which the displayed template is processed.
Local policies:	Displays the hexadecimal value of locally configured policies.
Inherited polices:	Displays the hexadecimal value of inherited policies. The 0x0 value is displayed when no templates are inherited.
Locally configured policies:	Displays a list of commands that are locally configured in a peer policy template.
Inherited policies:	Displays a list of commands that are inherited from a peer template.

Related Commands

Command	Description
inherit peer-policy	Configures a peer policy template to inherit the configuration from another peer policy template.
template peer-policy	Creates a peer policy template and enters policy-template configuration mode.

debug ip bgp groups

To display information related to the processing of Border Gateway Protocol (BGP) update-groups, use the **debug ip bgp update** privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp groups [*index-group* | *ip-address*]

no debug ip bgp groups

Syntax Description

index-group	(Optional) Specifies that update-group debugging information for the corresponding index number will be displayed. The range of update-group index numbers is from 1 to 4294967295.
ip-address	(Optional) Specifies that update-group debugging information for a single peer will be displayed.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

The output of this command displays information about update-group calculations and the addition and removal of update-group members. Information about peer-groups, peer-policy, and peer-session templates will also be displayed in the output of this command as neighbor configurations change.

Note The output of this command can be very verbose. This command should not be deployed in a production network unless you are troubleshooting a problem.

When a change to outbound policy occurs, the router automatically recalculates update-group memberships and applies the changes by triggering an outbound soft reset after a 3-minute timer expires. This behavior is designed to provide the network operator with time to change the configuration if a mistake is made. You can manually enable an outbound soft reset before the timer expires by entering the **clear ip bgp** *ip-address* **soft out** command.

Examples

The following sample output from the **debug ip bgp groups** command shows that peering has been established with neighbor 10.4.9.8 and update-group calculations are occurring for this member:

```
Router# debug ip bgp groups

5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 1 fl0
5w4d: BGP-DYN(0): Created update-group(0) flags 0x0 cap 0x0 from neighbor 10.4.0
5w4d: BGP-DYN(0): Adding  neighbor 10.4.9.8 flags 0x0 cap 0x0, to update-group 0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

The following sample output from the **debug ip bgp groups** command shows the recalculation of update-groups after the **clear ip bgp groups** command was issued:

```
Router# debug ip bgp groups

5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.5 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.8 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.21 flags 0x0 cap 0x0 and updgrp 1 f0
5w4d: BGP-DYN(0): Update-group 1 flags 0x0 cap 0x0 policies same as 10.4.9.21 f0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

Table 3 describes the significant fields shown in the display.

Table 3 debug ip bgp groups Field Descriptions

Field	Description
%BGP-5-ADJCHANGE:	A BGP neighbor has come Up or gone Down. The IP address of the neighbor is specified in the output string.
BGP-DYN(0):	This line is displayed when a neighbor adjacency is established. The BGP dynamic update group algorithm analyzes the policies of the new neighbor and then adds the neighbor to the appropriate BGP update group.

Related Commands

Command	Description
clear ip bgp	Resets a BGP connection or session.
clear ip bgp update-group	Clears BGP update-group member sessions.
show ip bgp replication	Displays BGP update-group replication statistics.
show ip bgp update-group	Displays information about BGP update-groups.

Glossary

dynamic peer-group—Another name for an update-group.

peer group—A group of neighbors, calculated at configuration time, that share the same outbound policies and can be replicated in the same updates.

peer policy template—Peer policy templates are used to used group and apply the configuration of commands that are applied within specific address-families and NLRI configuration modes.

peer session template—Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and NLRI configuration modes.

peer template—A configuration pattern that can be applied to neighbors that share common policies. Peer templates are reusable and support inheritance, allowing the network operator to group and apply distinct neighbor configurations for BGP neighbors that share common policies, and define very complex configuration patterns through the ability of a peer template to inherit configurations from other peer templates. There are two types of peer templates: “policy templates” and “session templates.”

update-group—Like a peer-group, an update-group is a group of neighbors, calculated at configuration time, that share the same outbound policies and can be replicated in the same updates. However, update-groups are calculated dynamically and are not restricted by outbound policies.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.