



Telephony Gateway Registration Protocol on Cisco IOS Gateways

The Telephony Gateway Registration Protocol (TGREP) on Cisco IOS Gateways feature allows gateways to send routing information to call-control agents, such as Cisco SIP Proxy Servers. TGREP is an auxiliary protocol for Telephony Routing over IP (TRIP), which is a protocol for interdomain exchange of telephone routing information.

Feature Specifications for the Telephone Routing over IP for Cisco IOS Gateways

Feature History

Release	Modification
12.3(1)	This feature was introduced.

Supported Platforms

For platforms supported in Cisco IOS Release 12.3.(1), consult Cisco Feature Navigator.

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

- [Prerequisites for TGREP on Cisco IOS Gateways, page 2](#)
- [Restrictions for TGREP on Cisco IOS Gateways, page 2](#)
- [Information About TGREP on Cisco IOS Gateways, page 2](#)
- [How to Configure TGREP on Cisco IOS Gateways, page 8](#)
- [Configuration Example for TGREP on IOS Gateways, page 39](#)
- [Where to Go Next, page 43](#)
- [Additional References, page 44](#)
- [Command Reference, page 45](#)
- [Glossary, page 107](#)

Prerequisites for TGREP on Cisco IOS Gateways

- Your gateway must have a voice image that is configurable for SIP.
- Establish a working IP network.
For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*.
- Configure VoIP.
For more information about configuring VoIP, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Restrictions for TGREP on Cisco IOS Gateways

TGREP can be used to advertise reachable routes on SIP networks only.

Information About TGREP on Cisco IOS Gateways

To configure TGREP on Cisco IOS gateways, you must understand the following concepts:

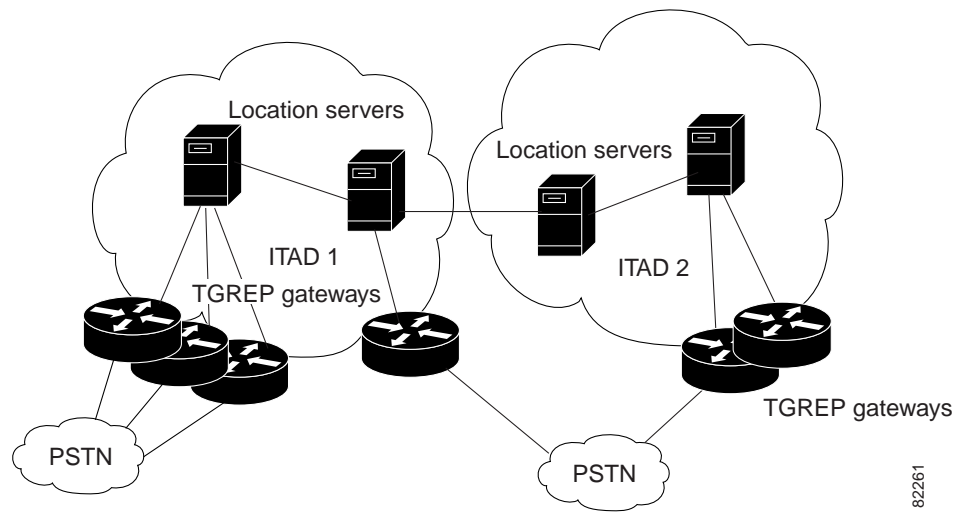
- [TRIP and TGREP, page 2](#)
- [Address Family Hierarchy, page 6](#)
- [Trunk Group Resources, page 7](#)
- [Destination Pattern Conversion, page 7](#)

TRIP and TGREP

Telephony Routing over IP (TRIP) is a protocol for interdomain exchange of telephone routing information. TRIP can be used for the discovery of reachable routes and gateway capabilities from gateways in a Voice over IP (VoIP) network. With TRIP, gateways can provide dynamic routing information to associated call-control agents, such as gatekeepers, routing servers, or proxy servers. On these call-control agents, the element that communicates TRIP information is the *location server*. TRIP allows the gateway to export routes to the location server and also to provide resource information so the location server can perform load balancing.

Location servers exchange routing information to build a graph of the IP Telephony Administrative Domain (ITAD). The ITAD consists of gateways, location servers, and other network elements under the control of a single administrative authority. TRIP works between elements of the ITAD, with the location servers exchanging information between each other to prevent routing loops, exchange attributes necessary to enforce policies, and select routes based on path or gateway characteristics. A sample topology is shown in [Figure 1](#).

Figure 1 Sample TRIP Topology



TRIP is modeled after Border Gateway Protocol 4 (BGP-4) and enhanced with some link state features, as in Open Shortest Path First (OSPF) protocol, Intermediate System-to-Intermediate System (IS-IS), and Server Cache Synchronization Protocol (SCSP). TRIP uses BGP's interdomain transport mechanism, BGP's peer communication, BGP's finite state machine, and formats and attributes similar to those of BGP. Unlike BGP, TRIP permits generic intradomain location server topologies, which simplifies configuration and increases scalability in contrast to BGP's full mesh requirement of internal BGP speakers. TRIP uses an intradomain flooding mechanism similar to that used in OSPF, IS-IS, and SCSP. Intradomain flooding keeps the telephony routing information synchronized between TRIP devices.

TRIP Operation

A TRIP database exchange begins when two peer location servers form a transport protocol connection between one another. They exchange messages to open and confirm the connection parameters, and negotiate the capabilities of each location server as well as the type of information to be advertised over this connection.

Keep-alive messages are sent periodically to ensure that adjacent peers are operational. Notification messages are sent in response to errors or special conditions. If a connection encounters an error condition, a Notification message is sent and the connection is closed.

Once the peer connection has been established, the initial data flow is a dump of all routes relevant to the new peer. Incremental updates are sent as the TRIP routing tables change. TRIP does not require periodic refresh of the routes. Therefore, a location server must retain the current version of all routing entries.

If a particular ITAD has multiple location servers and is providing transit service for other ITADs, then care must be taken to ensure a consistent view of routing within the ITAD. When synchronized, the TRIP routing tables of all internal peers are identical.

TRIP routes are advertised between a pair of location servers in UPDATE messages. The destination addresses and other attributes such as path or egress gateway are included in the UPDATE message.

TGREP in the TRIP Network

TGREP on Cisco IOS gateways works with TRIP and has similar procedures for session establishment. TGREP shares the same format for messages and a subset of attributes with TRIP. TGREP allows the gateway to convey a set of telephone routes to a location server, which resides on a Cisco SIP proxy server. The gateway only sends information because it is only interested in advertising its reachability and is not interested in learning about the reachability of other gateways and other domains. The telephone routes are represented by telephone number prefixes along with attributes that can express resource availability. With TGREP, the routing tables are exchanged once and only change when updates are sent.

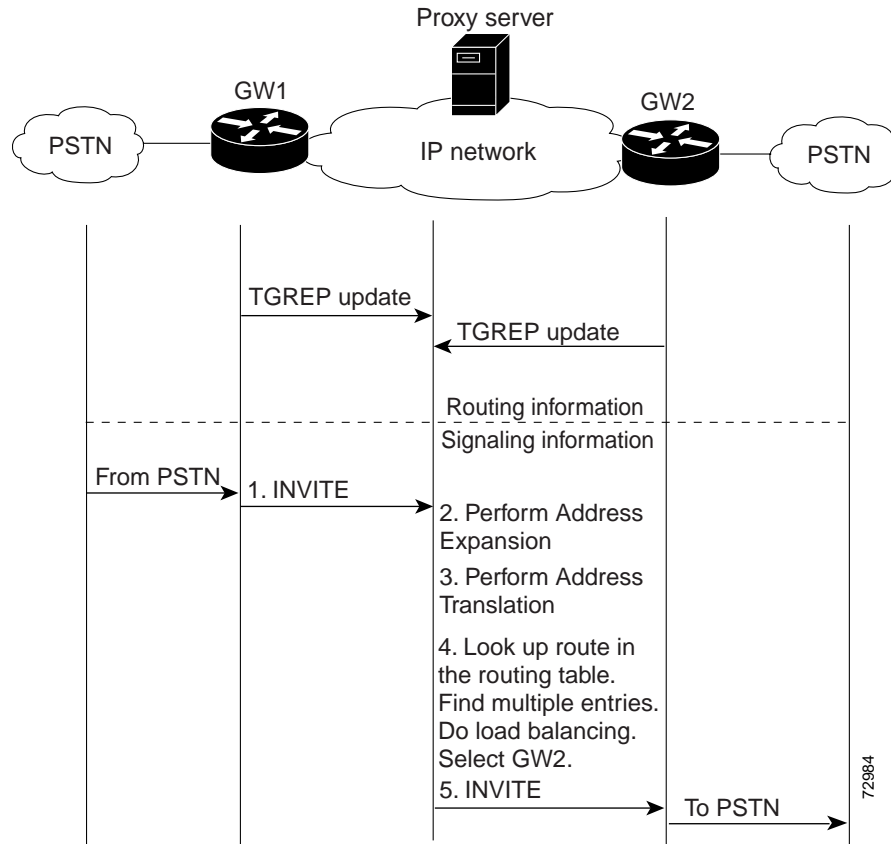
TGREP Operation

The TGREP gateway establishes a peer session with a location server within its ITAD. Once the peer session has been established, the gateway sends update messages to the TRIP location server with the gateway's reachability. The gateway also sends any attributes associated with the routes. If the gateway's reachability changes at any point in time, the gateway generates update messages with the change.

Keep-alive messages are periodically exchanged over the peer session between the TGREP gateway and the TRIP location server.

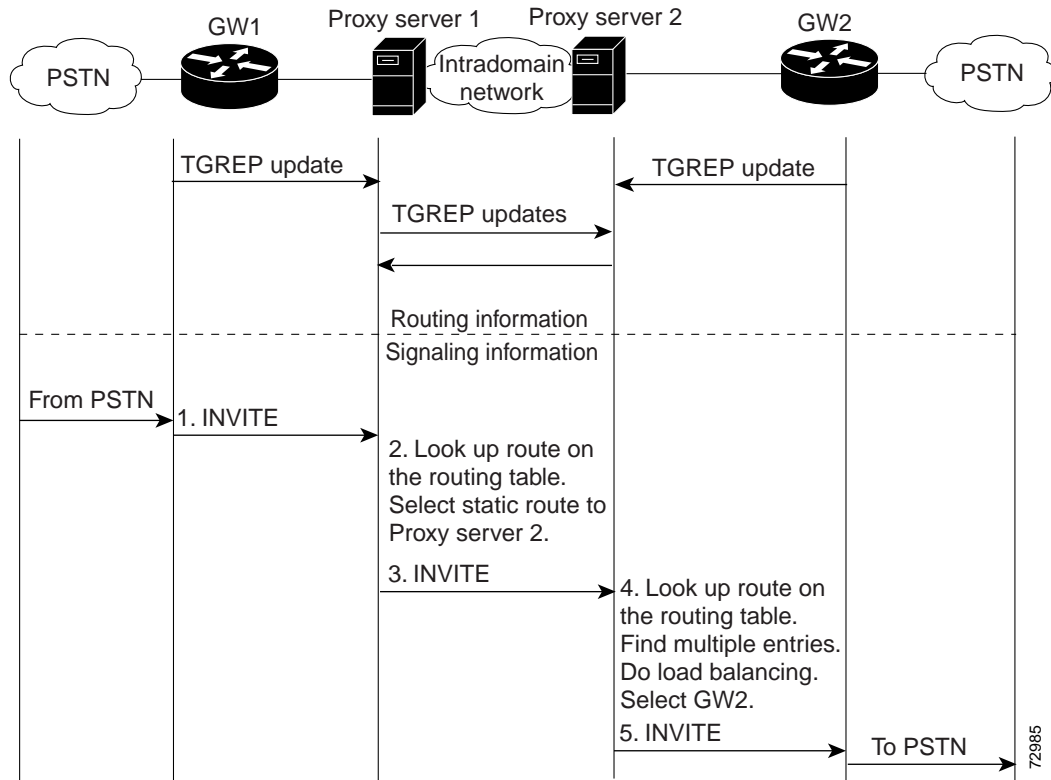
For simple intradomain calls, the gateway sends the route information to the Cisco SIP proxy server. The Cisco SIP Proxy Server performs functions such as address expansion and translation. The Cisco SIP Proxy Server finds an appropriate path and passes the call to the egress gateway, as shown in [Figure 2](#).

Figure 2 Intradomain Calls Using TGREP on IOS Gateways



Calls can also be routed between call-control agents within a domain, as shown in Figure 3. TRIP location servers can dynamically exchange routes.

Figure 3 Calls Routed Between Call-Control Agents Using TGREP on IOS Gateways

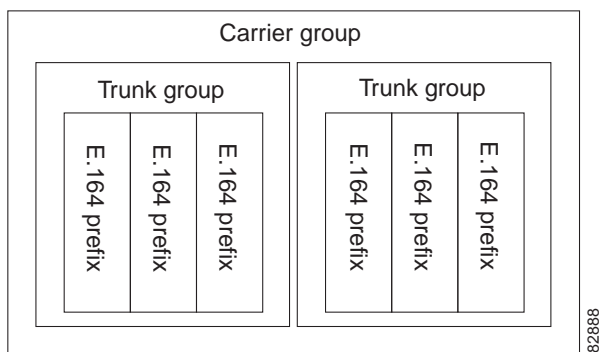


Address Family Hierarchy

In typical telephony environments, the E.164 address family is used to process a call. With TRIP, additional addressing is performed with carrier and trunk groups. This level of granularity provides additional flexibility in managing gateway resources, reduces potential update traffic between the gateway and the call control agent, and provides a framework for a scalable architecture. Voice over IP gateways using TGREP can interconnect with network facilities from different carriers and can advertise the carrier information in addition to the telephony destinations, to the call control agent. The address families can be ordered into a hierarchy, as follows:

- The prefixes used in E.164 addressing point to specific telephony destinations.
- Trunk groups can terminate calls to several telephony destinations, the information for which is provisioned on the gateway.
- Trunks of the same carrier may be grouped based on geographical considerations or on the basis of different grades of service that are offered by the carrier to its customers.

Advertisement, reporting, and capacity can be performed on each level of this hierarchy. An illustration of the address family hierarchy is shown in [Figure 4](#).

Figure 4 Address Family Hierarchy

The E.164 address family is used if the telephony network is a public telephony network. Decimal and pentadecimal options can be used to advertise private dial plans. For example if a company wants to use TRIP in within their enterprise telephony network using 5-digit extensions, then the gateway would advertise the beginning digits of their private numbers as a decimal address family. These calls cannot be sent out of the company's private telephony network because they are not E.164-compliant.

E.164 addresses cannot be more than 15 digits long and can be any of the following types of numbers:

- A geographic number, where the country code is 1 to 3 digits and the national number is 15-n (where n is 1 to 3). Individual countries can partition the national number into a subscriber number and an area code subject to the 15-n total length maximum.
- A global number with a 3 digit country code for global services, and a global subscriber number with up to 12 digits.
- An international public telecommunication number with a country code of 3 digit, an ID code of 1 to 4 digits, and a subscriber number of 12-x digits (where x=1 to 4).

Trunk Group Resources

Using trunk groups simplifies the task of configuring dial peers and also enables the dynamic selection of interfaces as needed in the gateway. If you have trunk groups configured under the dial peer, all trunk group resources are aggregated for the total circuits (TC) and available circuits (AC) attributes. If the trunk group is shut down and multiple trunk groups are configured, the TC and AC attributes are adjusted.

Destination Pattern Conversion

When the destination pattern is configured on the dial peer, TGREP converts the destination pattern to a TRIP prefix. The TRIP prefix contains only E.164 digits. If symbol or alphabetic characters are used, there is information loss when the conversion happens.

For example, a destination pattern "123..." means that any 6 digits starting with "123" can be terminated by this dial peer. However, TGREP advertises the prefix as "123", so the information about the three additional wildcards is lost. Similarly, if destination pattern is "123T", then TGREP advertises the prefix "123" only. In this case, the information about terminating calls starting with "123" but with any length is lost. When using a location server, the server routes the call to this gateway as long as the DNIS of the call starts with "123", the length of the DNIS does not matter.

Some destination patterns may not qualify as TRIP prefixes and cannot be advertised. [Table 1](#) shows some examples.

Table 1 Examples of Destination Pattern to TRIP Prefix Conversion

Destination pattern	TRIP prefix
123...	123
1234T	1234
12(3)%	12
12(3)+	123
12[3-4]	123 124
123	123
12(3)%4	Invalid for TRIP
12(3)+4	Invalid for TRIP
123*	Invalid for TRIP
.T	Invalid for TRIP

Note that for destination pattern "*" means the actual character "*" and "%" has the same meaning as "*" in regular expressions

How to Configure TGREP on Cisco IOS Gateways

This section contains the following procedures. Each procedure is identified as either required or optional.

- [Configuring a Local ITAD and TRIP Neighbor Connections, page 8](#) (Required)
- [Configuring the Global Carrier ID, page 11](#) (Optional)
- [Configuring the Trunk Groups, page 12](#) (Required)
- [Configuring a POTS Dial Peer for TGREP, page 14](#) (Required)
- [Configuring Voice Call Parameters, page 17](#) (Optional)
- [Verifying the TGREP Configuration on Cisco IOS Gateways, page 21](#) (Optional)
- [DETAILED STEPSTroubleshooting TGREP Configuration on Cisco IOS Gateways, page 24](#) (Optional)

Configuring a Local ITAD and TRIP Neighbor Connections

Configure the TRIP connections to start the TRIP processes, configure neighbors in the Internet Telephony Administrative Domain (ITAD), and specify address families.

SUMMARY STEPS

1. **enable**
2. **configure {terminal | memory | network}**

3. **tgrep local-itad** *string*
4. **neighbor** *ip-address*
5. **advertise** {**e164** | **decimal** | **penta-decimal** | **trunk-group** | **carrier**}[**csr**][**ac**][**tc**][**trunk-group** | **carrier**]
6. **address-family** {**e164** | **decimal** | **penta-decimal**}
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	tgrep local-itad <i>itad_number</i> Example: Router(config)# tgrep local-itad 1234	Enters TGREP configuration mode and define an ITAD. <ul style="list-style-type: none"> • <i>itad_number</i>—ITAD number associated with the gateway. The value can be from 1 to $2^{32}-1$.
Step 4	neighbor <i>ip-address</i> Example: Router(config-tgrep)# neighbor 192.168.255.255	Creates a TRIP session with the device at the specified IP address. <ul style="list-style-type: none"> • <i>ip-address</i>—IP address of a peer device with which TGREP information will be exchanged.

	Command or Action	Purpose
Step 5	<pre>advertise {e164 decimal penta-decimal trunk-group carrier} [csr] [ac] [tc] [trunk-group carrier]</pre> <p>Example: Router(config-tgrep)#advertise e164 csr ac tc trunk-group</p>	<p>Turns on the reporting for a particular attribute for any address family.</p> <ul style="list-style-type: none"> • <i>e164</i>—E.164 address family • <i>decimal</i>—Decimal address family • <i>penta-decimal</i>—Penta-decimal address family • <i>trunk-group</i>—Trunk group address family • <i>carrier</i>—Carrier code address family • <i>csr</i>—Call success rate • <i>ac</i>—Available circuits • <i>tc</i>—Total circuits <p>If you specify e164, decimal or penta-decimal for the address family, you can specify whether the related carrier or trunk-group parameters are advertised. If you specify carrier or trunk-group for the address family, you can specify that the related address family prefix is advertised. If you specify carrier or trunk-group for the address family, you cannot specify carrier or trunk-group attributes for advertising.</p> <p>When the no version of this command is used, it turns off the advertisement of that particular address family altogether.</p> <p>The default is to advertise none of the attributes for address families.</p>
Step 6	<pre>address-family {e164 decimal penta-decimal}</pre> <p>Example: Router(config-tgrep)# address-family e164</p>	<p>(Optional) Sets the global address family to be used by all of the dial peers. Address families configured on locally on dial peers override this global configuration.</p> <ul style="list-style-type: none"> • <i>e164</i>—E.164 address family. • <i>decimal</i>—Digital address family • <i>penta-decimal</i>—Pentadecimal address family
Step 7	<pre>exit</pre> <p>Example: Router(config-tgrep)# exit</p>	Exits TGREP configuration mode.
Step 8	<pre>end</pre> <p>Example: Router(config)# end</p>	Exits to privileged EXEC mode.

What to Do Next

Proceed to the [“Configuring the Global Carrier ID”](#) section.

Configuring the Global Carrier ID

The carrier ID for the trunk groups can be set at the global level. Carrier IDs configured locally on the trunk group supersede this global setting.

SUMMARY STEPS

1. **enable**
2. **configure** { **terminal** | **memory** | **network** }
3. **carrier-id** *string* [**cic**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	carrier-id <i>string</i> [cic] Example: Router(config)# carrier-id 101 cic	(Optional) The carrier ID for the trunk groups can be set at the global level. Carrier IDs configured locally on the trunk group supersede this global setting. <ul style="list-style-type: none"> • <i>string</i>—Identifier for the carrier ID. Must be 4-digit numeric carrier identification code to be advertised as a TRIP carrier family but can be alphanumeric if used otherwise. • cic—Specifies that the carrier ID is a circuit identification code. To advertise the carrier as a TRIP carrier family, the cic keyword must be used. When cic is used, only numeric values can be accepted for the <i>string</i> value. If cic is not used, the <i>string</i> value can be alphanumeric but is not advertised to TRIP location servers.
Step 4	end Example: Router(config)# end	Exits to privileged EXEC mode.

What to Do Next

Proceed to the [“Configuring the Trunk Groups”](#) section.

Configuring the Trunk Groups

For complete information about trunk group configuration, see the “Configuring Network Side ISDN PRI Signaling, Trunking, and Switching” chapter in the *Cisco IOS Dial Technologies Configuration Guide*:

SUMMARY STEPS

1. **enable**
2. **configure** { **terminal** | **memory** | **network** }
3. **trunk group** *group-number*
4. **tgrep advertise** { **trunk-group** | **carrier** } [**csr**] [**ac**] [**tc**] [**prefix**] [**disable**]
5. **carrier-id** *string* [**cic**]
6. **capacity update interval** *seconds*
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.
Step 3	<code>trunk group group-number</code> Example: Router(config)# trunk group 10	Defines the trunk group globally. <ul style="list-style-type: none">• <i>group-number</i>—A value from 0 to 23 that identifies the trunk group.

Command or Action	Purpose
<p>Step 4</p> <pre> tgrep advertise {trunk-group carrier} [csr] [ac] [tc] [disable] Example: Router(config-trunkgroup)# tgrep advertise trunk-group csr ac tc </pre>	<p>Turns on the advertisement of this trunk group.</p> <ul style="list-style-type: none"> • trunk-group—Specifies advertisement for the trunk group family. • carrier—Specifies advertisement for the carrier defined under this trunk group. • csr—Call success rate. • ac—Available circuits. • tc—Total circuits. • disable—Disables advertisement on the trunk group. <p>Note Use tgrep advertise disable to turn off advertisement for this trunk group. When the no tgrep advertise command is used, the global setting takes effect.</p> <ul style="list-style-type: none"> • When only tgrep advertise is entered, the trunk group is advertised without any other attribute. This command sets the attributes to be advertised for the trunk group or carrier or to disable advertisement of the this trunk group or carrier. <p>Note This command overrides the attributes set for advertisement using the global advertise (tgrep) command.</p>
<p>Step 5</p> <pre> carrier-id string [cic] Example: Router(config-trunkgroup)# carrier-id 101 cic </pre>	<p>(Optional) The carrier ID can be set locally for the trunk group. Carrier IDs configured locally on the trunk group supersede the global setting.</p> <ul style="list-style-type: none"> • string—Identifier for the carrier ID. Must be 4-digit numeric carrier identification code to be advertised as a TRIP carrier family but can be alphanumeric if used otherwise. • cic—Specifies that the carrier ID is a circuit identification code. To advertise the carrier as a TRIP carrier family, the cic keyword must be used. When cic is used, only numeric values can be accepted for the <i>string</i> value. If cic is not used, the <i>string</i> value can be alphanumeric but is not advertised to TRIP location servers.
<p>Step 6</p> <pre> capacity update interval seconds Example: Router(config-trunkgroup)# capacity update interval 50 </pre>	<p>(Optional) Changes the capacity update for this trunk group.</p> <ul style="list-style-type: none"> • seconds—Interval, in seconds, between the sending of periodic capacity updates. This can be a number in the range 10 to 1000. The default value is 25 seconds.

	Command or Action	Purpose
Step 7	<code>exit</code> Example: <code>Router(config-trunkgroup)# exit</code>	Exits trunk group configuration mode.
Step 8	<code>end</code> Example: <code>Router(config)# end</code>	Exits to privileged EXEC mode.

What to Do Next

Proceed to the [“Configuring a POTS Dial Peer for TGREP”](#) section.

Configuring a POTS Dial Peer for TGREP

To configure a POTS dial peer for TGREP, you must do the following:

- Identify the dial peer by assigning it a unique tag number.
- Define its destination telephone number or range of telephone numbers.
- Associate it with a voice port or trunk group through which calls are established.

Under most circumstances, the default values for the remaining dial peer configuration commands are sufficient to establish connections.

For complete information about configuring dial peers, see the “Configuring Dial Plans, Dial Peers, and Digit Manipulation” chapter in the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure** { **terminal** | **memory** | **network** }
3. **dial-peer voice** *number* **pots**
4. **destination-pattern** *string* [**T**]
5. **port** *location*
6. **tgrep address family** { **e164** | **decimal** | **penta-decimal** }
7. **tgrep advertise** [**csr**] [**ac**] [**tc**] [**carrier** | **trunk-group**] [**disable**]
8. **capacity update interval** *seconds*
9. **exit**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.
Step 3	<code>dial-peer voice number pots</code> Example: Router(config)# dial-peer voice 10 pots	Enters dial-peer configuration mode and defines a local dial peer that connects to a POTS. <ul style="list-style-type: none"> <code>interface.number</code>—Digits that define a particular dial peer. Valid entries are from 1 to 2147483647.
Step 4	<code>destination-pattern string</code> Example: Router(config-dial-peer)# destination-pattern 408555....	Matches dialed digits to a telephony device. <ul style="list-style-type: none"> <code>string</code>—A series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the numbers 0 through 9 and the letters A through D. <p>See the “Destination Pattern Conversion” section on page 7 for more information about valid destination pattern entries for TGREP.</p>
Step 5	<code>port location</code> Example: Router(config-dial-peer)# port 1/0/0	Maps the dial peer to a specific logical interface. <ul style="list-style-type: none"> The port command syntax is platform-specific. For more information about the syntax of this command, see the chapter “Configuring Voice Ports” in the <i>Cisco IOS Voice, Video, and Fax Configuration Guide</i>.
Step 6	<code>tgrep address family {e164 decimal penta-decimal}</code> Example: Router(config-dial-peer)# tgrep address family e164	Sets the address family to be used by the dial peer. <ul style="list-style-type: none"> Configuring the address family on this dial peer overrides the global address family configuration: <code>e164</code>—E.164 address family. <code>decimal</code>—Digital address family <code>penta-decimal</code>—Pentadecimal address family

	Command or Action	Purpose
Step 7	<pre>tgrep advertise [csr] [ac] [tc] [carrier trunk-group] [disable]</pre> <p>Example: Router(config-dial-peer)# tgrep advertise csr ac tc carrier</p>	<p>Turns on the prefix advertisement or any particular attribute of this dial peer.</p> <ul style="list-style-type: none"> • csr—Call success rate. • ac—Available circuits. • tc—Total circuits. • trunk-group—Specifies advertisement for the trunk group address family attributes. • carrier—Specifies advertisement for the carrier address family attributes. • disable—Disables advertisement on the trunk group. <p>Note Use tgrep advertise disable to turn off advertisement for this dial peer. When the no tgrep advertise command is used, the global setting takes effect.</p> <ul style="list-style-type: none"> • When only tgrep advertise is entered, the dial peer is advertised without any other attribute. This command sets the attributes to be advertised for the trunk group or carrier or to disable advertisement of the this trunk group or carrier. <p>Note This command overrides the attributes set for advertisement using the global advertise (tgrep) command.</p>
Step 8	<pre>capacity update interval seconds</pre> <p>Example: Router(config-dial-peer)# capacity update interval 50</p>	<p>(Optional) Changes the capacity update for prefixes related to this dial peer.</p> <ul style="list-style-type: none"> • <i>seconds</i>—Interval, in seconds, between the sending of periodic capacity updates. This can be a number in the range 10 to 1000. The default value is 25 seconds.
Step 9	<pre>exit</pre> <p>Example: Router(config-dial-peer)# exit</p>	Exits dial peer configuration mode.
Step 10	<pre>end</pre> <p>Example: Router(config)# end</p>	Exits to privileged EXEC mode.

What to Do Next

Proceed to the [“Configuring Voice Call Parameters”](#) section.

Configuring Voice Call Parameters

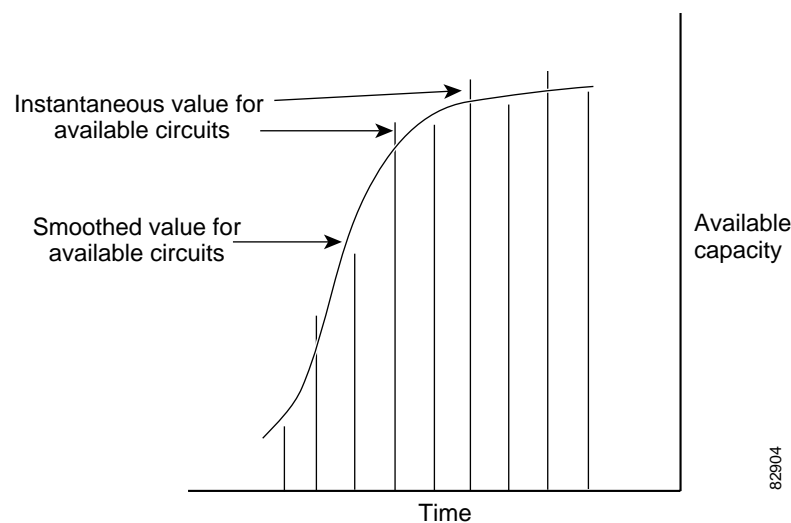
Different options for configuring voice call parameters can be used depending on what kind of data is to be sent.

Because the available circuit (AC) attribute of a destination is very dynamic, reporting of this attribute should be handled carefully. AC should be reported as frequently as possible so that the location server has current information about the resources. However, the location server should not be overwhelmed with too many updates.

A smoothing algorithm is applied to the quantity of AC reports. The algorithm eliminates reporting of noise. The degree of smoothing can be configured with the **voice call capacity mir stw** command. This command sets the smoothing transition time for weight (STW). STW is the time it takes for the current smoothed value of AC to come halfway between the current smoothed value and the current instantaneous value of AC. Lower STW values speed the smoothed value of AC as it approaches the instantaneous value of AC. When STW is set to 0, the smoothed value is always equal to the instantaneous value of AC.

Instantaneous and smoothed values of AC are shown in [Figure 5](#). Note that the instantaneous values, which are a sample at a particular moment in time, can spike, while the smoothed value takes the sample averaged over time.

Figure 5 Instantaneous and Smoothed Values for AC



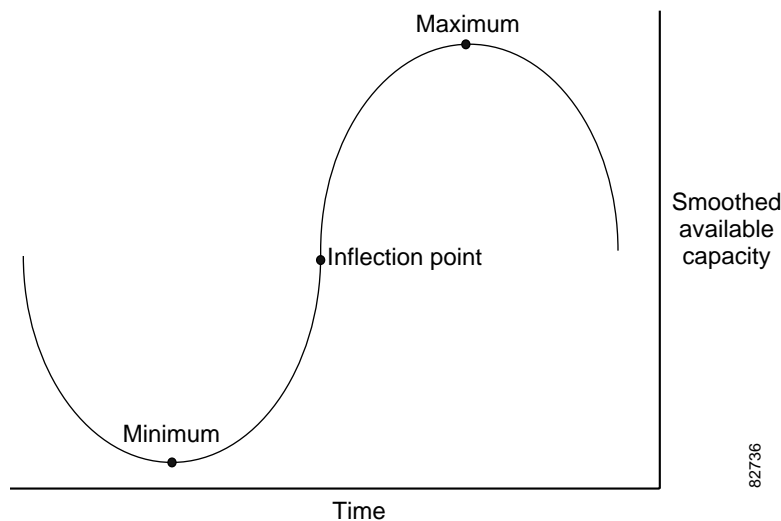
For the reporting interval, a periodic timer called the capacity update timer handles periodic updates of AC and can be configured using the **voice call capacity timer interval** command. For example, if AC has changed since the last reporting, the AC is again reported when the capacity update timer expires. In addition, AC is reported when any of the following events happens:

- Absolute percent change is above a threshold. This can be configured with the **voice call trigger hwm percent-change** command.
- Value of AC falls below a threshold, called the *low water mark*. This can be configured with the **voice call trigger hwm lwm** command.

- Value of AC goes above a threshold, called the *high water mark*. This can be configured with the **voice call trigger hwm hwm** command.
- The smoothed curve of AC has a maximum or a minimum. This represents a change in the direction of the call rate. This can be configured with the **voice call capacity reporting maxima** command. This reporting is off by default.
- The smoothed curve of AC has an inflection point. This represents the rate of call rate has changed sign. This can be configured with the **voice call capacity reporting inflection** command. This reporting is off by default.

Maximum, minimum and inflection points are illustrated in [Figure 6](#).

Figure 6 Maximum, Minimum, and Inflection Points for Available Capacity



All of the AC reporting, called the *interesting point of AC*, will be done if the specified event happens within the *minimum interval between reporting* (**mir**) time since last reporting. The **mir** parameter, configured in the **voice call capacity mir mir** command, sets the amount of time used for the interval to control the number of interesting points that are reported so not to overwhelm the location server with too many AC updates.

SUMMARY STEPS

1. **enable**
2. **configure {terminal | memory | network}**
3. **voice call {carrier | trunkgroup | prefix} capacity {mir | stw} value**
4. **voice call {carrier | trunkgroup | prefix} capacity reporting {maxima | inflection}**
5. **voice call {carrier | trunkgroup | prefix} capacity timer interval value**
6. **voice call {carrier | trunkgroup | prefix} csr data-points value**
7. **voice call {carrier | trunkgroup | prefix} csr recording interval value**
8. **voice call {carrier | trunkgroup | prefix} csr reporting interval value**
9. **voice call {carrier | trunkgroup | prefix} trigger {percent-change | lwm | hwm} value**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p>Example: Router> enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<pre>configure terminal</pre> <p>Example: Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>voice call {carrier trunkgroup prefix} capacity {mir stw} value</pre> <p>Example: Router(config)# voice call prefix capacity mir 25 </p>	<p>(Optional) Sets the minimum interval between reporting (mir) and smoothing time for weight (stw).</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • mir—Minimum interval between reporting • stw—Smoothing transition time for weight is the time it takes for current smoothed value of AC to come half way between the current smoothed value and the current instantaneous value of AC. Lower stw values speed the smoothed value of AC as it approaches the instantaneous value of AC. When stw is set to 0, the smoothed value will be always equal to the instantaneous value of AC. • value—If mir is selected, the value can be from 1 to 3600 seconds with a default of 10. This value cannot be set higher than the time configured for the capacity update interval. <p>If stw is selected, the value can be from 0 to 60 seconds with a default of 10.</p>
Step 4	<pre>voice call {carrier trunkgroup prefix} capacity reporting {maxima inflection}</pre> <p>Example: Router(config)# voice call trunkgroup capacity reporting inflection </p>	<p>(Optional) Turns on the reporting of maxima (first derivative) or inflection (second derivative) points in available circuits.</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • maxima—Maxima (first derivative) point in available capacity • inflection—Inflection (second derivative) point in available capacity

	Command or Action	Purpose
Step 5	<pre>voice call {carrier trunkgroup prefix} capacity timer interval value</pre> <p>Example: Router(config)# voice call trunkgroup capacity timer interval 100</p>	<p>(Optional) Sets the periodic reporting capacity from the carrier, trunk group, or prefix database.</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • <i>value</i>—10 to 3600 seconds with the default at 25
Step 6	<pre>voice call {carrier trunkgroup prefix} csr data-points value</pre> <p>Example: Router(config)# voice call carrier csr data-points 15</p>	<p>(Optional) Turns on the number of call success rate (CSR) history data points:</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • <i>value</i>—10 to 50 data points with the default at 30
Step 7	<pre>voice call {carrier trunkgroup prefix} csr recording interval value</pre> <p>Example: Router(config)# voice call trunkgroup csr recording interval 45</p>	<p>(Optional) Turns on the call success rate (CSR) recording interval.</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • <i>value</i>—10 to 1000 minutes with the default at 60.
Step 8	<pre>voice call {carrier trunkgroup prefix} csr reporting interval value</pre> <p>Example: Router(config)# voice call prefix csr reporting interval 60</p>	<p>(Optional) Turns on the call success rate (CSR) reporting interval.</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • <i>value</i>—10 to 10000 seconds with the default at 25

Command or Action	Purpose
<p>Step 9</p> <pre>voice call {carrier trunkgroup prefix} trigger {percent-change lwm hwm} value</pre> <p>Example: Router(config)# voice call carrier trigger lwm 25</p>	<p>(Optional) Sets the value for percentage change, low-water mark (lwm) or high-water mark (hwm) of available circuits in a carrier, trunk group, or prefix database:</p> <ul style="list-style-type: none"> • carrier—Carrier code address family • trunk-group—Trunk group address family • prefix—E.164 prefix • percent-change—Trigger when the percent change is reached. • lwm—Trigger when the low water mark is reached. • hwm—Trigger when the high water mark is reached. • <i>value</i>—If percent-change is selected, value can be 0 to 100 percent with a default of 30. If set to 0, this trigger will be turned off. <p>If lwm is selected, value can be 0 to 30 percent with a default of 10. If set to 0, this trigger will be turned off.</p> <p>If hwm is selected, value can be 50 to 100 percent with a default of 80. If set to 100, this trigger will be turned off.</p>
<p>Step 10</p> <pre>end</pre> <p>Example: Router(config)# end</p>	<p>Exits to privileged EXEC mode.</p>

What to Do Next

If, after you have performed the preceding tasks, the output from the optional show commands does not appear to be accurate, proceed to the [“DETAILED STEPSTroubleshooting TGREP Configuration on Cisco IOS Gateways”](#) task.

Verifying the TGREP Configuration on Cisco IOS Gateways

To verify that the TGREP on Cisco IOS Gateways feature is working, perform the following optional steps:

- [Monitor TGREP Status, page 21](#)
- [Clear and Reset TGREP Attributes, page 23](#)

Monitor TGREP Status

To display information about TGREP, perform these tasks:

SUMMARY STEPS

1. **enable**

2. **show tgrep neighbors** [* |*ip_address*]
3. **show voice eddri prefix** [*prefix_number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show tgrep neighbors [* <i>ip_address</i>] Example: Router# show tgrep neighbors *	Displays configured TRIP neighbors. <ul style="list-style-type: none"> The * keyword displays all neighbors, or an individual neighbor can be specified by its IP address.
Step 3	show voice eddri prefix [<i>prefix_number</i>] Example: Router# show voice eddri prefix 4	Displays applicable prefixes. <ul style="list-style-type: none"> If no prefix is specified, all configured prefixes appear. The Event Dispatcher and Data Repository Interface (EDDRI) notifies TGREP when an attribute changes on some subsystems. EDDRI interacts with the dial peer subsystem, the trunk group subsystems, CCAPI subsystem and the CRM subsystem to notify changes in particular attributes. EDDRI is responsible for creating the prefix database.

The following example displays all neighbors:

```
Router# show tgrep neighbors *

There are 1 nbrs configured

----- NBR:16.1.1.202-----
TIMERS:
  Keepalive : Timer Stopped
  Hold Timer : Timer Stopped
  Connect Retry : Running, time remaining in ms, 20698

SYNC IN PROGRESS
STATE: TRIPS_IDLE
QUEUES:
  writeQ : 0
  sec_writeQ : 0
  readQ : 0

SOCKET FDs:
prim socket -1, sec socket -1
tgrep_update_version : 0

LAST RESET: USER_INITIATED

Router#
Router#!!!! Trip Connection is setup here...
----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
```

```

0xFFFFFFFF98 0xFFFFFFFF9A 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

Option Paramater #1
Param Type: Capability
Length 8
    Cap Code :Send Receive Capability
    Cap Len  :4
        Send Rec Cap: RCV ONLY MODE
-->All route types supported

----- OPEN DUMP ENDS -----

```

Step 1 The following example displays applicable prefixes:

```

Router# show voice eddri prefix 4

prefix 4 address family decimal
advertise flag 0x27 ac 24 tc 24 capacity timer 25 sec
AC_avg 24, FD_avg 0, SD_avg 0
succ_curr 0 tot_curr 0
succ_report 0 tot_report 0
changed 0 replacement position 0
trunk group castg2
dial peer tag 1001

```

Clear and Reset TGREP Attributes

To clear TGREP attributes, use the following commands:

SUMMARY STEPS

1. **enable**
2. **clear tgrep counters [csr | ac] [{dial-peer | trunk-group | carrier} {value | *}][*]**
3. **clear tgrep neighbor {ip_address | *} }**

DETAILED STEPTroubleshooting TGREP Configuration on Cisco IOS Gateways

	Command or Action	Purpose
Step 1	<code>enable</code> Example: <code>Router> enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	<code>clear tgrep counters [csr ac] [{dial-peer trunk-group carrier} {value *}] [*]</code> Example: <code>Router# clear tgrep counters csr dial-peer *</code>	Clears the counters for call success rate (csr) and available circuits (ac) for all (*) or for a particular dial peer or trunk group.
Step 3	<code>clear tgrep neighbor {ip_address *}</code> Example: <code>Router# clear tgrep neighbor 170.10.10.255</code>	Resets the specified neighbor or all (*) neighbors.

To debug TGREP activity, events, and operations, use these commands in privileged EXEC mode. Use the no version of each command to turn off debugging:

SUMMARY STEPS

1. `enable`
2. `configure {terminal | memory | network}`
3. `debug tgrep error`
4. `debug tgrep events`
5. `debug tgrep fsm`
6. `debug tgrep io`
7. `debug tgrep msgdump`
8. `debug tgrep msgs`
9. `debug tgrep timer-event`
10. `debug tgrep timers`
11. `debug tgrep tripr`
12. `debug voip eddri {event | timers | prefix | all}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.
Step 3	<code>debug tgrep error</code> Example: Router# debug tgrep errors	Shows any errors in functioning.
Step 4	<code>debug tgrep events</code> Example: Router# debug tgrep events	Shows main events occurring throughout the subsystem.
Step 5	<code>debug tgrep fsm</code> Example: Router# debug tgrep fsm	Shows Finite State Machine (FSM) activity.
Step 6	<code>debug tgrep io</code> Example: Router# debug tgrep io	Shows detailed socket level activities.
Step 7	<code>debug tgrep msgdump</code> Example: Router# debug tgrep msgdump	Shows the dump of the details of TGREP messages.
Step 8	<code>debug tgrep messages</code> Example: Router# debug tgrep messages	Shows the movement of TGREP messages.
Step 9	<code>debug tgrep timer-events</code> Example: Router# debug tgrep timer-events	Shows events that are related to the timer.
Step 10	<code>debug tgrep timers</code> Example: Router# debug tgrep timers	Shows timer activity.

	Command or Action	Purpose
Step 11	<code>debug tgrep tripr</code> Example: Router# <code>debug tgrep tripr</code>	Enables debugs from the TRIP Reporter.
Step 12	<code>debug voip eddri {event timers prefix all}</code> Example: Router# <code>debug voip eddri all</code>	Enables debugs from the EDDRI.

The following example shows any errors in functioning:

```
Router# debug tgrep errors

Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on

----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

Option Paramater #1
Param Type: Capability
Length 8
      Cap Code :Send Receive Capability
      Cap Len  :4
      Send Rec Cap: RCV ONLY MODE
-->All route types supported

----- OPEN DUMP ENDS -----
We already have connection with such itad/tripid combo in progress

NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Error: Active connection to the nbr failed NBR:16.1.1.203
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
```

The following example shows main events occurring throughout the subsystem:

```
Router# debug tgrep events

tgrep-gw-1-02#Received a TGREP_UPD_TIMER timeout
```

```

The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time

```

The following example shows FSM activity:

```

Router# debug tgrep fsm

Generic routes combined : 0x61FA38B4, 13 bytes
+++++
 0x0 0x2 0x0 0x9 0x0
 0x5 0x0 0x0 0x0 0x3
 0x6D 0x63 0x69
-----
+++++
NEXT HOP SERVER : 0x61FA38C1, 10 bytes
+++++
 0x0 0x3 0x0 0x6 0x0
 0x0 0x4 0xFFFFFD2 0x0 0x0
-----
+++++
AD RD PATH : 0x61FA38CB, 10 bytes
+++++Getting a major event 4 on I/O
Received a TRIP_IO_WRITEQ_BOOLEAN event 313
The peer connection check for fd 1 is success
Writing some pending stuff first NBR:14.1.1.210
Moving ahead with more reading rc = 4
-->Starting regular write for nbr NBR:14.1.1.210
The queuesize before we start is 1
Selected primary socket for NBR:14.1.1.210
The peer connection check for fd 1 is success
Dequeued 1 message (left 0) for NBR:14.1.1.210 for writing to socket
A socket has gulped all that we fed it NBR:14.1.1.210 -- 92 bytes
Dequeued 0 message (left 0) for NBR:14.1.1.210 for writing to socket
Wrote out the whole socket buffer or Q in 2 attempts NBR:14.1.1.210 rc 4 was
NBR:14.1.1.210 Starting keepalive timer after writing something
Getting a major event 512 on I/O
Received an event on a socket for some nbr
Received Mask event of 0x1 for fd 1
Looking for fd match on nbr NBR:14.1.1.210
Recieved READ_EVENT for for nbr NBR:14.1.1.210
Read 3 bytes from that network for nbr NBR:14.1.1.210
+++++
This is what we READ : 0x63E79090, 3 bytes
+++++
 0x0 0x3 0x4
-----
NBR:14.1.1.210 Re-starting hold timer after a message is read
tmsg malloc total memory allocated is 95
Allocated another buffer for TRIP message

```

```

TRIP Messages Read so far 1
+++++
  Enqueing this tmsg : 0x691D09DC, 3 bytes
+++++
  0x0 0x3 0x4
-----

Enqueuing a message into the ReadQ of nbr: NBR:14.1.1.210
Read -1 bytes from that network for nbr NBR+++++
  0x0 0x4 0x0 0x6 0x2
  0x1 0x0 0x0 0x4 0xFFFFFD2
-----

+++++
  AD RD PATH : 0x61FA38D5, 10 bytes
+++++
  0x0 0x5 0x0 0x6 0x2
  0x1 0x0 0x0 0x4 0xFFFFFD2
-----

+++++
  LOCAL PREF : 0x61FA38DF, 8 bytes
+++++
  0x0 0x7 0x0 0x4 0x0
  0x0 0x0 0x5
-----

+++++
  Available Ckts : 0x61FA38E7, 8 bytes
+++++
  0x0 0xF 0x0 0x4 0x0
  0x0 0x0 0x17
-----

+++++
  TOTAL CIRCUITS : 0x61FA38EF, 8 bytes
+++++
  0x0 0x10 0x0 0x4 0x0
  0x0 0x0 0x17
-----

+++++
  CALL SUCCESS RATE : 0x61FA38F7, 12 bytes
+++++
  0x0 0x11 0x0
tgrep-gw-1-02#
tgrep-gw-1-02#und al:14.1.1.210
Getting a major event 512 on I/O
Errors : Process socket event has an invalid fd to work on
l 0x8 0x0
  0x0 0x0 0x78 0x0 0x0
  0x0 0x7F
-----

+++++
  PREFIX_ATTRIBUTE : 0x61FA3903, 64 bytes
+++++
  0x0 0x12 0x0 0x3C 0x0
  0x4 0x31 0x31 0x32 0x38
  0x0 0x4 0x31 0x31 0x32
  0x37 0x0 0x4 0x31 0x31
  0x32 0x36 0x0 0x4 0x31
  0x31 0x32 0x35 0x0 0x4
  0x31 0x31 0x32 0x34 0x0
  0x4 0x31 0x31 0x32 0x33
  0x0 0x4 0x31 0x31 0x32
  0x32 0x0 0x5 0x39 0x39
  0x39 0x39 0x39 0x0 0x9
  0x31 0x32 0x33 0x34 0x35
  0x36

```

The following example shows detailed socket level activities:

```

Router# debug tgrep io

Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.202
Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.203
A socket has gulped all that we fed it NBR:16.1.1.202 -- 5 bytes
Closing all the fds for NBR:16.1.1.202
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet

Going to initiate a connect to 16.1.1.202
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.202 fd 1
Received Mask event of 0x1 for fd 1
Recieved WRITE_EVENT for for nbr NBR:16.1.1.202
Only Active Open Succeeded
Post connect succeeded for the nbr NBR:16.1.1.202, fd 1
A socket has gulped all that we fed it NBR:16.1.1.202 -- 29 bytes
Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 29 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on

----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

Option Paramater #1
Param Type: Capability
Length 8
          Cap Code :Send Receive Capability
          Cap Len  :4
          Send Rec Cap: RCV ONLY MODE
-->All route types supported

----- OPEN DUMP ENDS -----
Doing fd reassignment for nbr NBR:16.1.1.202
Moving ahead with more reading rc = 4
A socket has gulped all that we fed it NBR:16.1.1.202 -- 3 bytes

```

```

Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Moving ahead with more reading rc = 4
A socket has gulped all that we fed it NBR:16.1.1.202 -- 598 bytes
Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 15 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2
Received WRITE_EVENT for for nbr NBR:16.1.1.203
The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2
Recieved WRITE_EVENT for for nbr NBR:16.1.1.203
The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2
Recieved WRITE_EVENT for for nbr NBR:16.1.1.203
The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet

```

```
Errors : Process socket event has an invalid fd to work on
```

```
Router#
Router#clear trip nei *
Router#Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.202
Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.203
A socket has gulped all that we fed it NBR:16.1.1.202 -- 5 bytes
Closing all the fds for NBR:16.1.1.202
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
```

The following example shows the dump of the details of TGREP messages:

```
Router# debug tgrep msgdump

tgrep-gw-1-02#Received an KEEPALIVE NBR:14.1.1.210
+++++
TMSG datagramstart : 0x69188648, 150 bytes
+++++
0x0 0xFFFFF96 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0x9 0x0 0x5 0x0 0x0
0x0 0x3 0x6D 0x63 0x69
0x0 0x3 0x0 0x6 0x0
0x0 0x4 0xFFFFFD2 0x0 0x0
0x0 0x4 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x5 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x7 0x0 0x4 0x0
0x0 0x0 0x5 0x0 0xF
0x0 0x4 0x0 0x0 0x0
0x16 0x0 0x10 0x0 0x4
0x0 0x0 0x0 0x17 0x0
0x11 0x0 0x8 0x0 0x0
0x0 0x74 0x0 0x0 0x0
0x7B 0x0 0x12 0x0 0x3C
0x0 0x4 0x31 0x31 0x32
0x38 0x0 0x4 0x31 0x31
0x32 0x37 0x0 0x4 0x31
0x31 0x32 0x36 0x0 0x4
0x31 0x31 0x32 0x35 0x0
0x4 0x31 0x31 0x32 0x34
0x0 0x4 0x31 0x31 0x32
0x33 0x0 0x4 0x31 0x31
0x32 0x32 0x0 0x5 0x39
0x39 0x39 0x39 0x39 0x0
0x9 0x31 0x32 0x33 0x34
0x35 0x36 0x37 0x38 0x39

-----
Received an KEEPALIVE NBR:14.1.1.210
+++++
TMSG datagramstart : 0x691B0CA0, 92 bytes
+++++
0x0 0x5C 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0xF 0x0 0x3 0x0 0x0
0x0 0x9 0x31 0x32 0x33
0x34 0x35 0x36 0x37 0x38
```

```

0x39 0x0 0x3 0x0 0x6
0x0 0x0 0x4 0xFFFFFD2 0x0
0x0 0x0 0x4 0x0 0x6
0x2 0x1 0x0 0x0 0x4
0xFFFFFD2 0x0 0x5 0x0 0x6
0x2 0x1 0x0 0x0 0x4
0xFFFFFD2 0x0 0x7 0x0 0x4
0x0 0x0 0x0 0x5 0x0
0xF 0x0 0x4 0x0 0x0
0x0 0x17 0x0 0x10 0x0
0x4 0x0 0x0 0x0 0x17
0x0 0x11 0x0 0x8 0x0
0x0 0x0 0x75 0x0 0x0
0x0 0x78
-----
+++++
TMSG datagramstart : 0x691885EC, 150 bytes
+++++
0x0 0xFFFFF96 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0x9 0x0 0x5 0x0 0x0
0x0 0x3 0x6D 0x63 0x69
0x0 0x3 0x0 0x6 0x0
0x0 0x4 0xFFFFFD2 0x0 0x0
0x0 0x4 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x5 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x7 0x0 0x4 0x0
0x0 0x0 0x5 0x0 0xF
0x0 0x4 0x0 0x0 0x0
0x16 0x0 0x10 0x0 0x4
0x0 0x0 0x0 0x17 0x0
0x11 0x0 0x8 0x0 0x0
0x0 0x75 0x0 0x0 0x0
0x7C 0x0 0x12 0x0 0x3C
0x0 0x4 0x31 0x31 0x32
0x38 0x0 0x4 0x31 0x31
0x32 0x37 0x0 0x4 0x31
0x31 0x32 0x36 0x0 0x4
0x31 0x31 0x32 0x35 0x0
0x4 0x31 0x31 0x32 0x34
0x0 0x4 0x31 0x31 0x32
0x33 0x0 0x4 0x31 0x31
0x32 0x32 0x0 0x5 0x39
0x39 0x39 0x39 0x39 0x0
0x9 0x31 0x32 0x33 0x34
0x35 0x36 0x37 0x38 0x39
-----
Received an KEEPALIVE NBR:14.1.1.210
Received an KEEPALIVE NBR:14.1.1.210

```

The following example shows the movement of TGREP messages:

```
Router# debug tgrep messages
```

```
tgrep-gw(config-tgrep)#Received an OPEN NBR:14.1.1.210
```

```

----- OPEN DUMP BEGINS -----
0x1 0x0 0x0 0xFFFFFB4 0x0
0x0 0x0 0x19 0x0 0x0
0x45 0x67 0x0 0x0

```



```

Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

Router#clear trip nei *
Router#Entering trip_reset_nbr_timers to reset timers
Starting the CONNECT timer for nbr NBR:16.1.1.202 for value of 30 seconds
Stopping hold timer and keepalive timer while resetting NBR:16.1.1.202
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
IO_CONNECT TIMER for nbr NBR:16.1.1.202 has expired
NBR:16.1.1.202 -Restarting the connect timer
NBR:16.1.1.202 starting the holder timer after post connect with large value

----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFF4B4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

Option Paramater #1

```



```

The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
NBR:16.1.1.202 Re-starting hold timer after a message is read
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

NBR:16.1.1.202 Received a Keepalive timer time out
NBR:16.1.1.202 Starting keepalive timer after writing something
NBR:16.1.1.202 Re-starting hold timer after a message is read
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
clear trip nei *
Router#Entering trip_reset_nbr_timers to reset timers
Starting the CONNECT timer for nbr NBR:16.1.1.202 for value of 30 seconds
Stopping hold timer and keepalive timer while resetting NBR:16.1.1.202
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

```

The following example shows timer activity:

```

Router# debug tgrep timers

tgrep-gw-1-02#Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
clReceived a TGREP_UPD_TIMER timeout

```

```

The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

tgrep-gw-1-02#Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

tgrep-gw-1-02#und aReceived a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

```

The following example enables debugs from the TRIP Reporter:

```
Router# debug tgrep trip
```

```

20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x2prefix 1128 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 0 total 0
20:51:11:
20:51:11: -----
20:51:11: attrib 0x4002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 1128
20:51:11: ac: 22
20:51:11:
20:51:11: =====
20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x27prefix 123456789 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 117 total 120
20:51:11: tg mci cc mci
20:51:11: tripr_build_triprtr_prefix_destination_ev tg mci cic 0 carrier mci
20:51:11:
20:51:11: -----
20:51:11: attrib 0x1C002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 123456789
20:51:11: csr: tot 120 succ 117
20:51:11: ac: 22tc: 23
20:51:11:
20:51:11: =====
20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x27prefix 99999 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 0 total 0
20:51:11: tg mci cc mci
20:51:11: tripr_build_triprtr_prefix_destination_ev tg mci cic 0 carrier mci
20:51:11:
20:51:11: -----
20:51:11: attrib 0x1C002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 99999

```

```

20:51:11: csr: tot 0 succ 0
20:51:11: ac: 22tc: 23
20:51:11:
20:51:11: =====

```

The following example enables debugs from the EDDRI:

```
Router# debug voip eddri all
```

```

21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:53: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:53: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:53: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:53: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:53: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:53: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:53: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:53: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:53: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:53: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:53: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:54: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:54: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:54: eddri_interesting_ac_pt: percent trigger diff 4
21:00:54: eddri_interesting_ac_pt: Interesting Point
21:00:54: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:54: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:54: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:54: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:54: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:54: eddri_interesting_ac_pt: percent trigger diff 4
21:00:54: eddri_interesting_ac_pt: Interesting Point
21:00:54: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

```

```

21:00:54: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:54: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:54: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:54: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:54: eddri_interesting_ac_pt: percent trigger diff 4
21:00:54: eddri_interesting_ac_pt: Interesting Point
21:00:54: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:54: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:54: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:54: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:54: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:54: eddri_interesting_ac_pt: percent trigger diff 4
21:00:54: eddri_interesting_ac_pt: Interesting Point
21:00:54: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1

21:00:54: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:58: eddri_send_crm_or_tg_event_to_clients : reason 0x8 subsys 3

21:00:58: eddri_send_crm_or_tg_event_to_clients : reason 0x8 subsys 1

21:00:58: eddri_send_crm_or_tg_event_to_clients attr 0x7F ev_id 2 qid 0x64209230 reason
0x8
eddri_dequeue_event : dequeue event

```

Configuration Example for TGREP on IOS Gateways

This configuration example shows TGREP enabled on a Cisco AS5300 gateway:

```

version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname Router
!
!
username all
spe 2/0 2/23
  firmware location feature_card_flash
!
!
resource-pool disable
clock timezone GMT 0
!
ip subnet-zero
ip host CALLGEN-SECURITY-V2 10.96.8.90 10.78.0.0
ip host dirt 192.168.254.254
!

```



Note

The global carrier ID is set here. Carrier IDs configured locally on the trunk group supersede this global setting.

```

carrier-id 33
!
!

```

**Note**

The trunk groups are defined here. These trunk groups are defined as TGREP carriers and are advertised to TRIP location servers, as the **cic** keyword is used in each trunk group. Note that in the **trunk group att**, the capacity update interval for that particular carrier ID has been set.

```

trunk group att
  carrier-id 444 cic
  capacity carrier update interval 90
!
!
trunk group mci
  carrier-id 16 cic
!
!
trunk group 404
  carrier-id 2 cic
!
!
trunk group 101
  capacity trunk-group update interval 90
!
isdn switch-type primary-ni
!
!

```

**Note**

The capacity timer interval has been set globally for trunk groups.

```

voice call trunkgroup capacity timer interval 100
!
voice service pots
!
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type modem
mta receive maximum-recipients 0
!
controller E1 0
  clock source line primary
  pri-group timeslots 1-31
!
controller E1 1
  ds0-group 1 timeslots 1-2 type e&m-fgb dtmf dnis
  ds0-group 2 timeslots 3-5 type e&m-fgb dtmf dnis
  ds0-group 3 timeslots 6-10 type e&m-fgb dtmf dnis
  cas-custom 1
  trunk-group att
  cas-custom 2
  trunk-group mci
  cas-custom 3
!
controller E1 2
  pri-group timeslots 1-24
!
controller E1 3

```

```
shutdown
clock source line secondary 3
ds0-group 0 timeslots 1-15,17-24 type e&m-immediate-start
cas-custom 0
!
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
!
interface Ethernet0
 ip address 10.8.56.13 255.255.0.0
 ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 fair-queue 64 256 32
 ip rsvp bandwidth 1000 1000
!
interface Serial0:15
 no ip address
 trunk-group 101
 isdn switch-type primary-ni
 isdn incoming-voice modem
 no cdp enable
!
interface Serial2:15
 no ip address
 shutdown
 isdn switch-type primary-ni
 isdn protocol-emulate network
 no isdn T309-enable
 no cdp enable
!
interface FastEthernet0
 ip address 172.16.140.117 255.255.0.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.8.0.1
ip route 172.16.0.0 255.0.0.0 172.16.140.1
no ip http server
ip pim bidir-enable
!
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
radius-server host 10.7.157.1 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key cisco
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
radius-server vsa send authentication
call rsvp-sync
!
voice-port 0:D
!
voice-port 1:1
 compand-type a-law
!
```

```

voice-port 1:2
  compand-type a-law
!
voice-port 1:3
  compand-type a-law
!
voice-port 2:D
!
voice-port 3:0
  compand-type a-law
!
!
mgcp profile default
!

```

**Note**

The local ITAD is configured in the following steps. Note that each type of address family is defined using separate advertise commands. If an address family is not defined, it is not advertised.

```

tgrep local-itad 1
  neighbor 2.3.4.5
  nexthop 1.1.1 advertise e164 csr ac tc trunk-group
  advertise decimal csr ac tc carrier
  advertise penta-decimal csr ac tc carrier
  advertise trunk-group ac tc
  advertise carrier csr tc
  address-family e164
!
tgrep 1
!
dial-peer cor custom
!
!
!
dial-peer voice 1 voip
  destination-pattern T
  session target ipv4:1.8.56.12
!

```

**Note**

The POTS dial peers are configured here. Note that for dial peers **101** and **105**, the address families are defined and override the global setting.

```

dial-peer voice 101 pots
  trunkgroup 101
  trunkgroup mci
  shutdown
  tgrep address-family decimal
  destination-pattern 510888....
  no digit-strip
  direct-inward-dial
  forward-digits all
!
dial-peer voice 102 pots
  trunk-group-label source 101
  direct-inward-dial
!
dial-peer voice 103 pots
  trunkgroup 101
  trunkgroup mci
  shutdown
  destination-pattern 567

```

```
!  
dial-peer voice 104 pots  
  trunkgroup 404  
  shutdown  
  destination-pattern 76[5-7]  
!  
dial-peer voice 105 pots  
  trunkgroup 101  
  tgrep address-family penta-decimal  
  destination-pattern 766  
!  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
line 1 24  
  no flush-at-activation  
  modem Host  
  no modem ibc  
line aux 0  
line vty 0 4  
  password lab  
!  
scheduler interval 1000  
end
```

Where to Go Next

If you now want to further configure TRIP on Cisco IOS gateways, refer to Cisco.com and choose the following path: **Cisco > Service & Support > Technical Assistance Center > Technologies > ??**

Additional References

For additional information related to TGREP on Cisco IOS gateways, refer to the following references:

Related Documents

Related Topic	Document Title
VoIP configuration tasks	<i>Cisco IOS Voice, Video, and Fax Configuration Guide</i> , Cisco IOS Release 12.2
Additional VoIP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS Voice, Video, and Fax Command Reference</i> , Cisco IOS Release 12.2
Cisco SIP Proxy Server configuration and administration tasks	<i>Cisco SIP Proxy Server Administrator Guide</i>

Standards

Standards ¹	Title
IETF draft	<i>Usage of TRIP in Gateways for Exporting Phone Routes</i>

1. Not all supported standards are listed.

MIBs

MIBs ¹	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

1. Not all supported MIBs are listed.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register>

RFCs

RFCs ¹	Title
RFC 3219	<i>Telephony Routing over IP (TRIP)</i>

1. Not all supported RFCs are listed.

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, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3(1) command reference publications.

- [address-family](#)
- [advertise \(tgrep\)](#)
- [capacity update interval \(dial peer\)](#)
- [capacity update interval \(trunk group\)](#)
- [carrier-id \(global\)](#)
- [carrier-id \(trunk group\)](#)
- [debug tgrep error](#)
- [debug tgrep events](#)
- [debug tgrep fsm](#)
- [debug tgrep io](#)
- [debug tgrep messages](#)
- [debug tgrep msgdump](#)
- [debug tgrep timer-event](#)
- [debug tgrep timers](#)
- [debug tgrep tripr](#)
- [debug voip eddri](#)
- [neighbor](#)
- [show tgrep neighbors](#)
- [show voice eddri prefix](#)

- **tgrep address-family**
- **tgrep advertise (dial peer)**
- **tgrep advertise (trunk group)**
- **tgrep local-itad**
- **voice call capacity mir**
- **voice call capacity stw**
- **voice call capacity reporting**
- **voice call capacity timer interval**
- **voice call csr data-points**
- **voice call csr recording interval**
- **voice call csr reporting interval**
- **voice call trigger hwm**
- **voice call trigger lwm**
- **voice call trigger percent-change**

address-family

To set the global address family to be used on all dial peers, use the **address-family** command in TGREP configuration mode. To change back to the default address family, use the **no** form of this command.

address family { **e164** | **decimal** | **penta-decimal** }

no address family { **e164** | **decimal** | **penta-decimal** }

Syntax Description		
	e164	E.164 address family.
	decimal	Digital address family
	penta-decimal	Pentadecimal address family

Defaults E.164 address family

Command Modes TGREP configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines The E. 164 address family is used if the telephony network is a public telephony network. Decimal and pentadecimal options can be used to advertise private dial plans. For example if a company wants to use TRIP in within their enterprise telephony network using 5-digit extensions, then the gateway would advertise the beginning digits of their private numbers as a decimal address family. These calls cannot be sent out of the company's private telephony network because they are not E.164-compliant.

The pentadecimal family allows numbers 0 through 9 and alphabetic characters A through E and can be used in countries where letters are also carried in the called number.

Examples The following example shows that the address family for ITAD 1234 is set for E.164 addresses:

```
Router(config)# tgrep local-itad 1234
Router(config-tgrep)# address family e164
```

Related Commands	Command	Description
	tgrep local-itad	Enters TGREP configuration mode and defines an ITAD.

advertise (tgrep)

To turn on reporting for a specified address family, use the **advertise** command in TGREP configuration mode. To turn off reporting for a specified address family, use the **no** form of this command.

```
advertise {e164 | decimal | penta-decimal }[csr][ac][tc][trunk-group | carrier]
```

```
advertise {trunk-group | carrier}[csr][ac][tc]
```

```
no advertise {e164 | decimal | penta-decimal | trunk-group | carrier}
```

Syntax Description

e164	E.164 address family.
decimal	Decimal address family
penta-decimal	Penta-decimal address family (what is this?)
trunk-group	Trunk group address family
carrier	Carrier code address family
csr	Call success rate
ac	Available circuits
tc	Total circuits

Defaults

No attributes for address families are advertised.

Command Modes

TGREP configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

If you specify **e164**, **decimal** or **penta-decimal** for the address family, you can stipulate whether the related **carrier** or **trunk-group** parameters are advertised. If you stipulate **carrier** or **trunk-group** for the address family, you can stipulate that the related address family prefix is advertised. If you stipulate **carrier** or **trunk-group** for the address family, you cannot stipulate **carrier** or **trunk-group** attributes for advertising.

When the **no** version of this command is used, it turns off the advertisement of that particular address family altogether.

Examples

The following example shows that the E.164 address family with call success rate, available circuits, total circuits, and trunk group attributes is being advertised for ITAD 1234:

```
Router(config)# tgrep local-itad 1234
Router(config-tgrep)# advertise e164 csr ac tc trunk-group
```

Related Commands

Command	Description
tgrep local-itad	Enters TGREP configuration mode and defines an ITAD.

capacity update interval (dial peer)

To change the capacity update for prefixes associated with this dial peer, use the **capacity update interval** command in dial peer configuration mode. To return to the default, use the **no** form of this command.

capacity update interval *seconds*

no capacity update interval *seconds*

Syntax Description	<i>seconds</i>	Interval, in seconds, between the sending of periodic capacity updates. This can be a number in the range 10 to 1000. The default value is 25 seconds.
---------------------------	----------------	--

Defaults	25 seconds
-----------------	------------

Command Modes	Dial peer configuration
----------------------	-------------------------

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines	The update interval should be set depending on the number of updates that are sent. Updates are sent more often when more calls are coming in, which can lead to data getting out of sync. If the interval is too short for the amount of updates, the location server can be overwhelmed. If this dial peer gets too much traffic, set the <i>seconds</i> argument to a higher value.
-------------------------	--

Examples	The following example shows that POTS dial peer 10 is having the capacity update occur every 35 seconds:
-----------------	--

```
Router(config)# dial-peer voice 10 pots
Router(config-dial-peer)# capacity update interval 35
```

Related Commands	Command	Description
	dial-peer voice	Enters dial-peer configuration mode and specifies the method of voice-related encapsulation.

capacity update interval (trunk group)

To change the capacity update for carriers or trunk groups, use the **capacity update interval** command in trunk group configuration mode. To return to the default, use the **no** form of this command.

capacity {carrier | trunk-group} update interval *seconds*

no capacity {carrier | trunk-group} update interval *seconds*

Syntax Description		
	carrier	Carrier capacity.
	trunk-group	Trunk group capacity.
	<i>seconds</i>	Interval, in seconds, between the sending of periodic capacity updates. This can be a number in the range 10 to 1000. The default value is 25 seconds.

Defaults 25 seconds

Command Modes Trunk group configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines The update interval should be set depending on the number of updates that are sent. Updates are sent more often when more calls are coming in, which can lead to data getting out of sync. If the interval is too short for the amount of updates, the location server can be overwhelmed. If this trunk group or carrier group gets too much traffic, set the *seconds* argument to a higher value.

Examples The following example sets the capacity update for trunk group 101 to occur every 45 seconds:

```
Router(config)# trunk group 101
Router(config-trunkgroup)# capacity trunk-group update interval 45
```

Related Commands	Command	Description
	trunk group	Defines the trunk group and enters trunk group configuration mode.

carrier-id (global)

To set the carrier ID for trunk groups when a local carrier ID is not configured, use the **carrier-id** command in global configuration mode. To disable the carrier ID, use the **no** form of this command.

carrier-id *name* [**cic**]

no carrier-id *name*[**cic**]

Syntax Description

<i>name</i>	Identifier for the carrier ID. Must be 4-digit numeric carrier identification code to be advertised as a TRIP carrier family but can be alphanumeric if used otherwise.
cic	Specifies that the carrier ID is a circuit identification code(CIC).

Defaults

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

To advertise the carrier as a TRIP carrier family, the **cic** keyword must be used. When **cic** is used, only numeric values can be accepted for the *name* value. If **cic** is not used, the *name* value can be alphanumeric but is not advertised to TRIP location servers.

Examples

The following example shows a carrier ID using the circuit identification code:

```
Router(config)# carrier-id 1234 cic
```

Related Commands

Command	Description
carrier-id (trunk group)	Configures the carrier ID locally on the trunk group.

carrier-id (trunk group)

To specify the carrier associated with a trunk group, use the **carrier-id** command in trunk group configuration mode. To delete the source carrier ID, use the **no** form of this command.

carrier-id *name* [**cic**]

no carrier-id *name* [**cic**]

Syntax Description	<i>name</i>	Specifies the ID of the carrier to use for the call. Valid carrier IDs contain a maximum of 127 alphanumeric characters. To be advertised as a TRIP carrier family, this must be set to a 4-digit numeric carrier identification code.
	cic	Specifies that the carrier ID is a circuit identification code.

Defaults No default behavior or values

Command Modes Trunk group configuration

Command History	Release	Modification
	12.2(11)T	This command was introduced.
	12.3(1)	The cic keyword was added.

Usage Guidelines In a network, calls are routed over incoming trunk groups and outgoing trunk groups. The *name* arguments identifies the carrier that handles the calls for a specific trunk group. In some cases, the same trunk group may be used to carry both incoming calls and outgoing calls.

The carrier ID configured locally on the trunk group supersedes the globally configured carrier ID.

To advertise the carrier as a TRIP carrier family, the **cic** keyword must be used. When **cic** is used, only numeric values can be accepted for the *name* value. If **cic** is not used, the *name* value can be alphanumeric but is not advertised to TRIP location servers.

Examples The following example indicates that carrier “alpha1” carries calls for trunk group 5:

```
Router(config)# trunk group 5
Router(config-trunk-group)# carrier-id alpha1
```

The following example shows that the carrier with circuit identification code 1234 carries calls for trunk group 101. This trunk group can carry TRIP advertisements:

```
Router(config)# trunk group 101
Router(config-trunk-group)# carrier-id 1234 cic
```

■ carrier-id (trunk group)

Related Commands	Command	Description
	carrier-id (global)	Configures the carrier ID globally for all trunk groups.
	translation-profile (trunk group)	Associates a translation profile with a trunk group.
	trunk group	Initiates the definition of a trunk group.

debug tgrep error

To turn on debugging for any TGREP errors, use the **debug tgrep error** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep error

no debug tgrep error

Defaults

Debug is not enabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

There is always a performance penalty when using debug commands.

The “We already have connection with such itad/tripid combo in progress” message appears when an error occurs where two location servers with the same ITAD and TripID initiate a TRIP connection to the gateway. When the second OPEN message arrives at the gateway, the **debug trip error** command displays the message.

Examples

The following example shows output from the **debug tgrep error** command:

```
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
```

After the errors are reported, the open dump begins. The ITAD is identified in the dump.

```
----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

      Version      :1
      Hold Time    :180
      My ITAD      :1112
      TRIP ID      :101161129

      Option Paramater #1
      Param Type: Capability
      Length 8
           Cap Code :Send Receive Capability
           Cap Len  :4
```

```

                                Send Rec Cap: RCV ONLY MODE
-->All route types supported

----- OPEN DUMP ENDS -----

```

The “We already have connection with such itad/tripid combo in progress” message appears when an error occurs where two location servers with the same ITAD and TripID initiate a TRIP connection to the gateway.

We already have connection with such itad/tripid combo in progress

```

NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Error: Active connection to the nbr failed NBR:16.1.1.203
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Errors : Process socket event has an invalid fd to work on
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on

```

Related Commands

Command	Description
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep events

To turn on debugging for main events occurring throughout the subsystem, use the **debug tgrep events** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep events

no debug tgrep events

Defaults

Debug is not enabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

There is always a performance penalty when using debug commands.

Examples

The following example shows output from the **debug tgrep events** command:

```
tgrep-gw-1-02#Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
```

[Table 2](#) describes the significant fields shown in the display.

Table 2 *debug tgrep events Field Descriptions*

Field	Description
Received a TGREP_UPD_TIMER timeout	This event shows that a TREP update timer timeout event occurred.
The bulkSyncQ size is 0 at this time	This event indicates the size of bulk sync queue.
The tgrepQ size is 0 at this time	This event indicates the size of TGREP queue.

Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.

■ debug tgrep events

Command	Description
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep fsm

To turn on debugging for Finite State Machine (FSM) events, use the **debug tgrep fsm** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep fsm

no debug tgrep fsm

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

Examples The following example shows output from the **debug tgrep fsm** command:

```
Generic routes combined : 0x61FA38B4, 13 bytes
+++++
0x0 0x2 0x0 0x9 0x0
0x5 0x0 0x0 0x0 0x3
0x6D 0x63 0x69
-----
+++++
NEXT HOP SERVER : 0x61FA38C1, 10 bytes
+++++
0x0 0x3 0x0 0x6 0x0
0x0 0x4 0xFFFFFD2 0x0 0x0
-----
+++++
AD RD PATH : 0x61FA38CB, 10 bytes
+++++Getting a major event 4 on I/O
```

Here, a write event occurs. Note how the finite state machine details each step of the writing process.

```
Received a TRIP_IO_WRITEQ_BOOLEAN event 313
The peer connection check for fd 1 is success
Writing some pending stuff first NBR:14.1.1.210
Moving ahead with more reading rc = 4
-->Starting regular write for nbr NBR:14.1.1.210
The queuesize before we start is 1
Selected primary socket for NBR:14.1.1.210
The peer connection check for fd 1 is success
Dequeued 1 message (left 0) for NBR:14.1.1.210 for writing to socket
A socket has gulped all that we fed it NBR:14.1.1.210 -- 92 bytes
Dequeued 0 message (left 0) for NBR:14.1.1.210 for writing to socket
Wrote out the whole socket buffer or Q in 2 attempts NBR:14.1.1.210 rc 4 was
NBR:14.1.1.210 Starting keepalive timer after writing something
```

```
Getting a major event 512 on I/O
Received an event on a socket for some nbr
Received Mask event of 0x1 for fd 1
Looking for fd match on nbr NBR:14.1.1.210
```

Now a read event occurs. After this event, the total number of TRIP messages read is displayed.

```
Recieved READ_EVENT for for nbr NBR:14.1.1.210
Read 3 bytes from that network for nbr NBR:14.1.1.210
+++++
  This is what we READ : 0x63E79090, 3 bytes
+++++
  0x0 0x3 0x4
-----
NBR:14.1.1.210 Re-starting hold timer after a message is read
tmsg malloc total memory allocated is 95
Allocated another buffer for TRIP message
TRIP Messages Read so far 1
+++++
  Enqueing this tmsg : 0x691D09DC, 3 bytes
+++++
  0x0 0x3 0x4
-----
Enqueuing a message into the ReadQ of nbr: NBR:14.1.1.210
Read -1 bytes from that network for nbr NBR+++++
  0x0 0x4 0x0 0x6 0x2
  0x1 0x0 0x0 0x4 0xFFFFFD2
-----
```

Statistics for available circuits, total circuits, and call success rate are displayed.

```
+++++
  AD RD PATH : 0x61FA38D5, 10 bytes
+++++
  0x0 0x5 0x0 0x6 0x2
  0x1 0x0 0x0 0x4 0xFFFFFD2
-----
+++++
  LOCAL PREF : 0x61FA38DF, 8 bytes
+++++
  0x0 0x7 0x0 0x4 0x0
  0x0 0x0 0x5
-----
+++++
  Available Ckts : 0x61FA38E7, 8 bytes
+++++
  0x0 0xF 0x0 0x4 0x0
  0x0 0x0 0x17
-----
+++++
  TOTAL CIRCUITS : 0x61FA38EF, 8 bytes
+++++
  0x0 0x10 0x0 0x4 0x0
  0x0 0x0 0x17
-----
+++++
  CALL SUCCESS RATE : 0x61FA38F7, 12 bytes
+++++
  0x0 0x11 0x0
tgrep-gw-1-02#
tgrep-gw-1-02#und al:14.1.1.210
Getting a major event 512 on I/O
Errors : Process socket event has an invalid fd to work on
1 0x8 0x0
```

```

0x0 0x0 0x78 0x0 0x0
0x0 0x7F
-----
+++++
PREFIX_ATTRIBUTE : 0x61FA3903, 64 bytes
+++++

```

The prefix is shown here in hex format.

```

0x0 0x12 0x0 0x3C 0x0
0x4 0x31 0x31 0x32 0x38
0x0 0x4 0x31 0x31 0x32
0x37 0x0 0x4 0x31 0x31
0x32 0x36 0x0 0x4 0x31
0x31 0x32 0x35 0x0 0x4
0x31 0x31 0x32 0x34 0x0
0x4 0x31 0x31 0x32 0x33
0x0 0x4 0x31 0x31 0x32
0x32 0x0 0x5 0x39 0x39
0x39 0x39 0x39 0x0 0x9
0x31 0x32 0x33 0x34 0x35
0x36

```

Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep io

To turn on debugging for detailed socket-level activities, use the **debug tgrep io** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep io

no debug tgrep io

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

Examples The following example shows output from the **debug tgrep io** command:

```

Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.202
Dispatching a TRIP_EV_NBR_IO_ASYNC_RESET to I/O for NBR:16.1.1.203
A socket has gulped all that we fed it NBR:16.1.1.202 -- 5 bytes
Closing all the fds for NBR:16.1.1.202
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
NBR:16.1.1.202 is not eligible to write, no non(-1) fd yet
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet

```

At this point, the connection is initiated.

```

Going to initiate a connect to 16.1.1.202
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.202 fd 1
Received Mask event of 0x1 for fd 1
Recieved WRITE_EVENT for for nbr NBR:16.1.1.202
Only Active Open Succeeded
Post connect succeeded for the nbr NBR:16.1.1.202, fd 1
A socket has gulped all that we fed it NBR:16.1.1.202 -- 29 bytes
Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202

```

Errors begin to appear here.

```
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 29 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
```

After the errors are detected, a dump occurs. The ITAD and TRIP ID are displayed.

```
----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

Option Paramater #1
Param Type: Capability
Length 8
          Cap Code :Send Receive Capability
          Cap Len  :4
                Send Rec Cap: RCV ONLY MODE
-->All route types supported

----- OPEN DUMP ENDS -----
Doing fd reassignment for nbr NBR:16.1.1.202
Moving ahead with more reading rc = 4
A socket has gulped all that we fed it NBR:16.1.1.202 -- 3 bytes
Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Moving ahead with more reading rc = 4
A socket has gulped all that we fed it NBR:16.1.1.202 -- 598 bytes
Wrote out the whole socket buffer or Q in 2 attempts NBR:16.1.1.202 rc 4 was
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 15 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2
Recieved WRITE_EVENT for for nbr NBR:16.1.1.203
```

```

The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2

```

Errors continue to occur. Note that the router still attempts to write, but the connection is not active.

```

Recieved WRITE_EVENT for for nbr NBR:16.1.1.203
The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on
Received Mask event of 0x1 for fd 1
Recieved READ_EVENT for for nbr NBR:16.1.1.202
Read 3 bytes from that network for nbr NBR:16.1.1.202
Read -1 bytes from that network for nbr NBR:16.1.1.202
Errors : Process socket event has an invalid fd to work on
Going to initiate a connect to 16.1.1.203
Called a socket_connect with errno 11, confirmation later
Initiated a Async connect call for nbr NBR:16.1.1.203 fd 2
Received Mask event of 0x1 for fd 2
Recieved WRITE_EVENT for for nbr NBR:16.1.1.203
The Active connect never succeeded, no passive yet, resetting NBR:16.1.1.203
Error: Active connection to the nbr failed NBR:16.1.1.203
A Socket error has caused a write failure NBR:16.1.1.203 errno 13
Closing all the fds for NBR:16.1.1.203
Post connect succeeded for the nbr NBR:16.1.1.203, fd -1
Moving ahead with more reading rc = 4
NBR:16.1.1.203 is not eligible to write, no non(-1) fd yet
Errors : Process socket event has an invalid fd to work on

```

Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep messages

To turn on debugging for movement of TGREP messages, use the **debug tgrep messages** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep messages

no debug tgrep messages

Defaults

Debug is not enabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

There is always a performance penalty when using debug commands.

Examples

The following example shows output from the **debug tgrep messages** command:

```
tgrep-gw(config-tgrep)#Received an OPEN NBR:14.1.1.210
```

```
----- OPEN DUMP BEGINS -----
0x1 0x0 0x0 0xFFFFFFFFB4 0x0
0x0 0x0 0x19 0x0 0x0
0x45 0x67 0x0 0x0

      Version      :1
      Hold Time    :180
      My ITAD      :25
      TRIP ID      :17767

      No optional parameters -- hence all route types supported.
      Send-Recv capability in effect

----- OPEN DUMP ENDS -----
```

After the dump occurs, the TRGREP messages are displayed. In this case, keepalive messages are being received by this gateway.

```
Enqueued a Keepalive for NBR:14.1.1.210
Received an KEEPALIVE NBR:14.1.1.210
Received Keepalive for NBR:14.1.1.210
Received an KEEPALIVE NBR:14.1.1.210
```

Related Commands	Command	Description
	debug tgrep error	Turns on debugging for any errors in functioning.
	debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
	debug tgrep fsm	Turns on debugging for FSM activity.
	debug tgrep io	Turns on debugging for detailed socket level activities.
	debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
	debug tgrep timer-event	Turns on debugging for events that are related to the timer.
	debug tgrep timers	Turns on debugging for timer activity.
	debug tgrep tripr	Turns on debugging for the TRIP Reporter.
	debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep msgdump

To turn on debugging for the dump of the details of TGREP messages, use the **debug tgrep msgdump** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep msgdump

no debug tgrep msgdump

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

Examples The following example shows output from the **debug tgrep msgdump** command:

```

tgrep-gw-1-02#Received an KEEPALIVE NBR:14.1.1.210
+++++
MSG datagramstart : 0x69188648, 150 bytes
+++++
0x0 0xFFFFF96 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0x9 0x0 0x5 0x0 0x0
0x0 0x3 0x6D 0x63 0x69
0x0 0x3 0x0 0x6 0x0
0x0 0x4 0xFFFFFD2 0x0 0x0
0x0 0x4 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x5 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFD2
0x0 0x7 0x0 0x4 0x0
0x0 0x0 0x5 0x0 0xF
0x0 0x4 0x0 0x0 0x0
0x16 0x0 0x10 0x0 0x4
0x0 0x0 0x0 0x17 0x0
0x11 0x0 0x8 0x0 0x0
0x0 0x74 0x0 0x0 0x0
0x7B 0x0 0x12 0x0 0x3C
0x0 0x4 0x31 0x31 0x32
0x38 0x0 0x4 0x31 0x31
0x32 0x37 0x0 0x4 0x31
0x31 0x32 0x36 0x0 0x4
0x31 0x31 0x32 0x35 0x0
0x4 0x31 0x31 0x32 0x34
0x0 0x4 0x31 0x31 0x32
0x33 0x0 0x4 0x31 0x31

```

```
debug tgrep msgdump
```

```
0x32 0x32 0x0 0x5 0x39
0x39 0x39 0x39 0x39 0x0
0x9 0x31 0x32 0x33 0x34
0x35 0x36 0x37 0x38 0x39
```

After each event occurs, a dump of the message appears. The entire dump of each keepalive is being displayed.

```
-----
Received an KEEPALIVE NBR:14.1.1.210
+++++
TMSG datagramstart : 0x691B0CA0, 92 bytes
+++++
0x0 0x5C 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0xF 0x0 0x3 0x0 0x0
0x0 0x9 0x31 0x32 0x33
0x34 0x35 0x36 0x37 0x38
0x39 0x0 0x3 0x0 0x6
0x0 0x0 0x4 0xFFFFFFFFD2 0x0
0x0 0x0 0x4 0x0 0x6
0x2 0x1 0x0 0x0 0x4
0xFFFFFFFFD2 0x0 0x5 0x0 0x6
0x2 0x1 0x0 0x0 0x4
0xFFFFFFFFD2 0x0 0x7 0x0 0x4
0x0 0x0 0x0 0x5 0x0
0xF 0x0 0x4 0x0 0x0
0x0 0x17 0x0 0x10 0x0
0x4 0x0 0x0 0x0 0x17
0x0 0x11 0x0 0x8 0x0
0x0 0x0 0x75 0x0 0x0
0x0 0x78
-----
+++++
TMSG datagramstart : 0x691885EC, 150 bytes
+++++
0x0 0xFFFFFFFF96 0x2 0x0 0x1
0x0 0x0 0x0 0x2 0x0
0x9 0x0 0x5 0x0 0x0
0x0 0x3 0x6D 0x63 0x69
0x0 0x3 0x0 0x6 0x0
0x0 0x4 0xFFFFFFFFD2 0x0 0x0
0x0 0x4 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFFFFD2
0x0 0x5 0x0 0x6 0x2
0x1 0x0 0x0 0x4 0xFFFFFFFFD2
0x0 0x7 0x0 0x4 0x0
0x0 0x0 0x5 0x0 0xF
0x0 0x4 0x0 0x0 0x0
0x16 0x0 0x10 0x0 0x4
0x0 0x0 0x0 0x17 0x0
0x11 0x0 0x8 0x0 0x0
0x0 0x75 0x0 0x0 0x0
0x7C 0x0 0x12 0x0 0x3C
0x0 0x4 0x31 0x31 0x32
0x38 0x0 0x4 0x31 0x31
0x32 0x37 0x0 0x4 0x31
0x31 0x32 0x36 0x0 0x4
0x31 0x31 0x32 0x35 0x0
0x4 0x31 0x31 0x32 0x34
0x0 0x4 0x31 0x31 0x32
0x33 0x0 0x4 0x31 0x31
0x32 0x32 0x0 0x5 0x39
0x39 0x39 0x39 0x39 0x0
```

```
0x9 0x31 0x32 0x33 0x34
0x35 0x36 0x37 0x38 0x39
```

```
-----
Received an KEEPALIVE NBR:14.1.1.210
Received an KEEPALIVE NBR:14.1.1.210
```

Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep timer-event

To turn on debugging for events that are related to the timer, use the **debug tgrep timer-event** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep timer-event

no debug tgrep timer-event

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

Examples The following example shows output from the **debug tgrep timer-event** command:

```
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
```

The TRIP timer registers timeouts until the next event occurs. Here, the timers are reset.

```
Entering trip_reset_nbr_timers to reset timers
Starting the CONNECT timer for nbr NBR:16.1.1.202 for value of 30 seconds
Stopping hold timer and keepalive timer while resetting NBR:16.1.1.202
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
```

Timeouts are again reported until the next event.

```
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
```

Here, the TRIP neighbor is cleared, which causes the timer to reset.

```
Router#clear trip nei *
Router#Entering trip_reset_nbr_timers to reset timers
Starting the CONNECT timer for nbr NBR:16.1.1.202 for value of 30 seconds
Stopping hold timer and keepalive timer while resetting NBR:16.1.1.202
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry

Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 3 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
IO_CONNECT_TIMER for nbr NBR:16.1.1.202 has expired
NBR:16.1.1.202 -Restarting the connect timer
NBR:16.1.1.202 starting the holder timer after post connect with large value
```

```
----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3
```

```
Version :1
```


Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timers	Turns on debugging for timer activity.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep timers

To turn on debugging for detailed socket level activities, use the **debug tgrep timers** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep timers

no debug tgrep timers

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

Examples The following example shows output from the **debug tgrep timers** command:

```
tgrep-gw-1-02#Received a TGREP_UPD_TIMER timeout
The bulkSyncQ size is 0 at this time
The tgrepQ size is 0 at this time
Restarting the router UPD timer after expiry
```

[Table 4](#) describes the significant fields in the display.

Table 3 *debug tgrep timers Field Descriptions*

Field	Description
Received a TGREP_UPD_TIMER timeout	This indicates that a timeout was received.
The bulkSyncQ size is 0 at this time	This indicates the size of the bulk sync queue.
The tgrepQ size is 0 at this time	This indicates the size of the TGREP queue.
Restarting the router UPD timer after expiry	This indicates that the timer has been reset.

Related Commands	Command	Description
	debug tgrep error	Turns on debugging for any errors in functioning.
	debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
	debug tgrep fsm	Turns on debugging for FSM activity.
	debug tgrep io	Turns on debugging for detailed socket level activities.

Command	Description
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep tripr	Turns on debugging for the TRIP Reporter.
debug voip eddri	Turns on debugging for the EDDRI.

debug tgrep tripr

To turn on debugging from the TRIP Reporter, use the **debug tgrep tripr** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug tgrep tripr

no debug tgrep tripr

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

A watched queue is used to inform the TRIP Reporter (TRIPR) process about changes in any of the interesting attributes of dial peer that potentially could trigger TRIP update. A dial peer attribute change manifest into a prefix attribute change and is deposited into the watched queue of TRIPR by the Event Dispatcher. The trunk group system also does the same.

Examples The following example shows output from the **debug tgrep tripr** command:

```
20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x27prefix 1128 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 0 total 0
20:51:11:
20:51:11: -----
20:51:11: attrib 0x4002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 1128
20:51:11: ac: 22
20:51:11:
20:51:11: =====
20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x27prefix 123456789 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 117 total 120
20:51:11: tg mci cc mci
20:51:11: tripr_build_triprtr_prefix_destination_ev tg mci cic 0 carrier mci
20:51:11:
20:51:11: -----
20:51:11: attrib 0x1C002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 123456789
20:51:11: csr: tot 120 succ 117
20:51:11: ac: 22tc: 23
```

```

20:51:11:
20:51:11: =====
20:51:11: tripr_build_triprtr_prefix_destination_ev : got the ev id 1 reason 64 num_prefix
1 advertise 0x27prefix 99999 addrFam 4
20:51:11: tripr_build_triprtr_prefix_destination_ev ac 22 tc 23 ac_avg 22
20:51:11: tripr_build_triprtr_prefix_destination_ev csr success 0 total 0
20:51:11:   tg mci cc mci
20:51:11: tripr_build_triprtr_prefix_destination_ev tg mci cic 0 carrier mci
20:51:11:
20:51:11: -----
20:51:11: attrib 0x1C002
20:51:11: ***** REACHABLE ROUTE *****
20:51:11: TRIP_AF_E164 99999
20:51:11:   csr: tot 0 succ 0
20:51:11:   ac: 22tc: 23
20:51:11:
20:51:11: =====

```

Table 4 describes the significant fields in the display.

Table 4 *debug tgrep tripr Field Descriptions*

Field	Description
ev id	This field can contain the following entries: <ul style="list-style-type: none"> • 1 - Prefix regular event • 2 - Trunk group regular event • 3 - Carrier regular event • 4 - Prefix sync event • 5 - Trunk group sync event • 6 - Carrier sync event • 7 - Null sync event
reason: (for a prefix family event)	This field can contain the following entries: <ul style="list-style-type: none"> • 1 - Prefix down • 2 - Prefix up • 4 - Prefix trunk group attribute changed • 8 - Prefix available circuits changed • 16- Prefix total circuits changed • 32 - Prefix CSR changed • 64 - Prefix AC interesting point • 128 - Prefix carrier attributes changed • 256 - Prefix stop advertise configured • 512 - Prefix start advertise configured

Table 4 *debug tgrep tripr* Field Descriptions

Field	Description
reason: (for a trunk group family event)	This field can contain the following entries: <ul style="list-style-type: none"> • 1 - Trunk group down • 2 - Trunk group up • 4 - Trunk group prefix attribute changed • 8 - Trunk group available circuits changed • 16 - Trunk group total circuits changed • 32 - Trunk group CSR changed • 64 - Trunk group AC interesting point • 128 - Trunk group stop advertise configured • 256 - Trunk group start advertise configured
reason: (for a carrier family event)	This field can contain the following entries: <ul style="list-style-type: none"> • 1 - Carrier down • 2 - Carrier up • 4 - Carrier prefix attribute changed • 8 - Carrier available circuits changed • 16 - Carrier total circuits changed • 32 - Carrier CSR changed • 64 - Carrier AC interesting point • 128 - Carrier stop advertise configured • 256 - Carrier start advertise configured

Related Commands

Command	Description
debug tgrep error	Turns on debugging for any errors in functioning.
debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
debug tgrep fsm	Turns on debugging for FSM activity.
debug tgrep io	Turns on debugging for detailed socket level activities.
debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
debug tgrep messages	Turns on debugging for the movement of TGREP messages.
debug tgrep timer-event	Turns on debugging for events that are related to the timer.
debug tgrep timers	Turns on debugging for timer activity.
debug voip eddri	Turns on debugging for the EDDRI.

debug voip eddri

To turn on debugging for the event dispatcher and data repository interface (EDDRI), use the **debug voip eddri** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

```
debug voip eddri {event | timers | prefix | all}
```

```
no debug voip eddri {event | timers | prefix | all}
```

Syntax Description		
	event	Turns on debugging for EDDRI events.
	timers	Turns on debugging for EDDRI timers.
	prefix	Turns on debugging for the prefix database.
	all	Turns on debugging all EDDRI activities.

Defaults Debug is not enabled

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines There is always a performance penalty when using debug commands.

The event dispatcher and data repository interface (EDDRI) notifies TGREP when an attribute changes on some subsystems. EDDRI interacts with the dial peer subsystem, the trunk group subsystems, call control API (CCAPI) subsystem and the customer relationship management (CRM) subsystem to notify changes in particular attributes. EDDRI is responsible for creating the prefix database.

Examples The following example shows output from the **debug voip eddri** command:

```
21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1
```

With the send prefix event the available circuits value and the triggers for reporting are updated.

```
21:00:53: eddri_send_prefix_event_to_clients attr 0xFF ev_id 1 qid 0x64209230 reason 0x40
eddri_dequeue_event : dequeue event

21:00:53: eddri_interesting_ac_pt : tc 23 IAC 22 lwm 5 hwm 50 pct_trigger 2 oneMinusW 933
21:00:53: eddri_interesting_ac_pt: old AC_curr 23 FD_curr 0 SD_curr 0
21:00:53: eddri_interesting_ac_pt: new AC_curr 22 FD_curr -5 SD_curr -5
21:00:53: eddri_interesting_ac_pt: percent trigger diff 4
21:00:53: eddri_interesting_ac_pt: Interesting Point
21:00:53: eddri_send_prefix_event_to_clients : reason 0x40 num_prefix 1
```

Related Commands	Command	Description
	debug tgrep error	Turns on debugging for any errors in functioning.
	debug tgrep events	Turns on debugging for main events occurring throughout the subsystem.
	debug tgrep fsm	Turns on debugging for FSM activity.
	debug tgrep io	Turns on debugging for detailed socket level activities.
	debug tgrep msgdump	Turns on debugging for the dump of the details of TGREP messages.
	debug tgrep messages	Turns on debugging for the movement of TGREP messages.
	debug tgrep timer-event	Turns on debugging for events that are related to the timer.
	debug tgrep timers	Turns on debugging for timer activity.
	debug tgrep tripr	Turns on debugging for the TRIP Reporter.
	show voice eddri prefix	Shows applicable prefixes for the EDDRI.

neighbor

To create a TGREP session with another device, use the **neighbor** command in TGREP configuration mode. To disable a TRIP connection, use the **no** form of this command.

neighbor *ip_address*

no neighbor *ip_address*

Syntax	Description
<i>ip_address</i>	IP address of a peer device with which TGREP information will be exchanged.

Defaults	Description
	No neighboring devices are defined

Command Modes	Description
	TGREP configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Examples The following example shows that the gateway with the IP address 192.116.56.10 is defined as a neighbor for ITAD 1234:

```
Router(config)# tgrep local-itad 1234
Router(config-tgrep)# neighbor 192.116.56.10
```

Related Commands	Command	Description
	tgrep local-itad	Enters TGREP configuration mode and defines an ITAD.

show tgrep neighbors

To show configured TGREP neighbors, use the **show tgrep neighbors** command in privileged EXEC mode.

```
show tgrep neighbors [* | ip_address]
```

Syntax Description		
	*	Displays all neighbors
	<i>ip_address</i>	Individual neighbor can be specified by its IP address

Defaults No default behavior or values

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Examples The following example shows output for the **show tgrep neighbors** command:

```
There are 1 nbrs configured

----- NBR:16.1.1.202-----
TIMERS:
  Keepalive : Timer Stopped
  Hold Timer : Timer Stopped
  Connect Retry : Running, time remaining in ms, 20698

SYNC IN PROGRESS
STATE: TRIPS_IDLE
QUEUES:
  writeQ : 0
  sec_writeQ : 0
  readQ : 0

SOCKET FDs:
prim socket -1, sec socket -1
tgrep_update_version : 0

LAST RESET: USER_INITIATED

Router#
Router#!!!! Trip Connection is setup here...
----- OPEN DUMP BEGINS -----
0x1 0xFFFFFFFF 0x0 0xFFFFFFFFB4 0x0
0x0 0x4 0x58 0x6 0x7
0xFFFFFFFF98 0xFFFFFFFFA9 0x0 0xC 0x0
0x1 0x0 0x8 0x0 0x2
0x0 0x4 0x0 0x0 0x0
0x3
```

```

Version      :1
Hold Time    :180
My ITAD      :1112
TRIP ID      :101161129

      Option Paramater #1
      Param Type: Capability
      Length 8
          Cap Code :Send Receive Capability
          Cap Len  :4
              Send Rec Cap: RCV ONLY MODE
-->All route types supported

```

```
----- OPEN DUMP ENDS -----
```

Related Commands

Command	Description
neighbor	Creates a TGREP session with another device.

show voice eddri prefix

To show applicable prefixes for the event dispatcher and data repository interface (EDDRI), use the **show voice eddri prefix** command in privileged EXEC mode.

show voice eddri prefix [*prefix_number*]

Syntax Description	all	Displays all neighbors
	<i>prefix_number</i>	(Optional) Specified EDDRI prefix

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines If no prefix is specified, all configured prefixes appear.

The event dispatcher and data repository interface (EDDRI) notifies TGREP when an attribute changes on some subsystems. EDDRI interacts with the dial peer subsystem, the trunk group subsystems, call control API (CCAPI) subsystem and the customer relationship management (CRM) subsystem to notify changes in particular attributes. EDDRI is responsible for creating the prefix database.

Examples The following example shows output for the **show voice eddri prefix** command:

```
prefix 4 address family decimal
advertise flag 0x27 ac 24 tc 24 capacity timer 25 sec
AC_avg 24, FD_avg 0, SD_avg 0
succ_curr 0 tot_curr 0
succ_report 0 tot_report 0
changed 0 replacement position 0
trunk group castg2
dial peer tag 1001
```

Related Commands	Command	Description
	debug voip eddri	Turns on debugging for the EDDRI.

tgrep address-family

To set the address family to be used on a local dial peer, use the **tgrep address-family** command in dial peer configuration mode. To return to the global setting, use the **no** form of this command.

tgrep address family {e164 | decimal | penta-decimal}

no tgrep address family {e164 | decimal | penta-decimal}

Syntax Description	e164	E.164 address family.
	decimal	Decimal address family
	penta-decimal	Penta-decimal address family

Defaults No default behavior or values.

Command Modes Dial peer configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines The E. 164 address family is used if the telephony network is a public telephony network. Decimal and pentadecimal options can be used to advertise private dial plans. For example if a company wants to use TRIP in within their enterprise telephony network using 5-digit extensions, then the gateway would advertise the beginning digits of their private numbers as a decimal address family. These calls cannot be sent out of the company's private telephony network because they are not E.164-compliant.

The pentadecimal family allows numbers 0 through 9 and alphabetic characters A through E and can be used in countries where letters are also carried in the called number.

Examples The following example shows that POTS dial peer 10 has the address family set for E.164 addresses:

```
Router(config)# dial-peer voice pots 10
Router(config-dial-peer)# tgrep address family e164
```

Related Commands	Command	Description
	dial-peer voice	Enters dial-peer configuration mode and specifies the method of voice-related encapsulation.

tgrep advertise (dial peer)

To set the attributes for advertisement of the prefix on this dial peer or to disable advertisement on this dial peer altogether, use the **tgrep advertise** command in dial peer configuration mode. To return to using the global setting, use the **no** form of this command.

```
tgrep advertise [csr] [ac] [tc] [carrier | trunk-group] [disable]
```

```
no tgrep advertise [csr] [ac] [tc] [carrier | trunk-group] [disable]
```

Syntax Description

csr	Call success rate
ac	Available circuits
tc	Total circuits
carrier	Carrier code address family
trunk-group	Trunk group address family
disable	Disables advertisement of this dial peer

Defaults

Prefix advertisement is not sent.

Command Modes

Dial peer configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

When only **tgrep advertise** is entered, the dial peer is advertised without any other attribute.

When **no tgrep advertise** is used on the dial peer, the dial peer inherits the attributes set in the global **advertise** command.

When the global **no advertise** command is used, it forbids advertisement of that particular address family altogether. The **tgrep advertise** command has no effect until the advertisement of the address family is enabled globally.

Examples

The following example shows a TGREP advertisement that sends call success rate, available circuits, total circuits, and carrier address family attribute information:

```
Router(config)# dial-peer voice pots 10
Router(config-dial-peer)# tgrep advertise csr ac tc carrier
```

Related Commands

Command	Description
dial-peer voice	Enters dial-peer configuration mode and specifies the method of voice-related encapsulation.

tgrep advertise (trunk group)

To turn on the advertisement of this trunk group for resource availability and other carrier information, use the **tgrep advertise** command in trunk group configuration mode. To turn off local trunk group advertisement and use the global setting, use the **no** form of this command.

tgrep advertise [csr] [ac] [tc] [disable]

no tgrep advertise [csr] [ac] [tc] [disable]

Syntax Description

csr	Call success rate.
ac	Available circuits.
tc	Total circuits.
disable	Disables advertisement on the trunk group.

Defaults

Trunk group advertisement is not sent

Command Modes

Trunk group configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

When only **tgrep advertise** is entered, the trunk group is advertised without any other attribute. When **no tgrep advertise** is used, the trunk group uses the global setting configured with the **advertise** command in TGREP configuration mode. To turn off advertisement of this trunk group, the **disable** keyword should be used.

There is a subtle difference between the **no** form of this command and the **no** form of the global **advertise** command. When **no tgrep advertise** is used on the trunk group, the trunk group inherits the attributes set in the global **advertise** command.

When the global **no advertise** command is used, it forbids advertisement of that particular address family altogether. The **tgrep advertise** command has no effect until the advertisement of the address family is enabled globally.

When the **carrier** keyword is used, the carrier defined under the trunk group assumes the configuration. Because multiple trunk groups can have the same carrier defined, the same configuration will show up under all trunk groups that have the same carrier defined. When the **no tgrep advertise carrier** command is used to revert to the global carrier configuration for the carrier under this trunk group, the same will happen to all the trunk groups who have the same carrier defined under them.



Note

This command overrides the attributes set for advertisement using the global **advertise (tgrep)** command.

Examples

The following example shows that trunk group 101 has been configured to send a TGREP advertisement that sends call success rate, available circuits, total circuits, and prefix attribute information:

```
Router(config)# trunk group 101  
Router(config-dial-peer)# tgrep advertise csr ac tc carrier
```

Related Commands

Command	Description
trunk group	Defines the trunk group and enters trunk group configuration mode.

tgrep local-itad

To enable TGREP on the gateway and enter TGREP configuration mode, use the **tgrep local-itad** command in global configuration mode. To disable TRIP on the gateway, use the **no** form of this command.

tgrep local-itad *itad_number*

no tgrep local-itad *itad_number*

Syntax Description	<i>itad_number</i>	ITAD number associated with the gateway. The value can be from 1 to 4294967295.
---------------------------	--------------------	---

Defaults	TGREP is not enabled on the gateway.
-----------------	--------------------------------------

Command Modes	Global configuration
----------------------	----------------------

Command History	Release	Modification
	12.3(1)	This command was introduced.

Examples	The following example shows TGREP being enabled for ITAD number 1234:
-----------------	---

```
Router (config)# tgrep local-itad 1234
```

Related Commands	Command	Description
	address-family	Sets the global address family to be used on all dial peers.
	advertise (tgrep)	Turns on reporting for a specified address family.
	neighbor	Creates a TGREP session with another device.

voice call capacity mir

To set the value for the minimum interval between reporting (MIR), use the **voice call capacity mir** command in global configuration mode. To turn off these attributes, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **capacity mir** *seconds*

no voice call { **carrier** | **trunk-group** | **prefix** } **capacity mir**

Syntax Description		
	carrier	Carrier code address family
	trunk-group	Trunk group address family
	prefix	E.164 prefix
	<i>value</i>	Minimum interval, in seconds, with a range of 1 to 3600 seconds and a default of 10. This value cannot be set higher than the time configured for the capacity update interval .

Defaults 10 seconds.

Command Modes Global configuration.

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines Because the available circuit (AC) attribute of a destination is very dynamic, reporting of this attribute should be handled carefully. AC should be reported as frequently as possible so that the location server has better information about the resources. However, the location server should not be overwhelmed with too many updates.

All of the AC reporting, called the *interesting point of AC*, is performed when the specified event happens within the *minimum interval between reporting* (MIR) time since last reporting. This command sets the amount of time used for the interval to control the number of interesting points that are reported so not to overwhelm the location server with too many AC updates.

The *seconds* argument cannot be set higher than the time configured for the **capacity update interval**.

Examples The following example shows the minimum interval between reporting for the carrier address family set to 25 seconds:

```
Router(config)# voice call carrier capacity mir 25
```

Related Commands	Command	Description
	capacity update interval (dial peer)	Changes the capacity update for prefixes associated with a dial peer.
	capacity update interval (trunk group)	Change the capacity update for carriers or trunk groups.
	voice call capacity stw	Set the value for STW.

voice call capacity stw

To set the value for smoothing transition time for weight (STW), use the **voice call capacity stw** command in global configuration mode. To turn off these attributes, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **capacity stw** *seconds*

no voice call { **carrier** | **trunk-group** | **prefix** } **capacity stw**

Syntax Description		
	carrier	Carrier code address family
	trunk-group	Trunk group address family
	prefix	E.164 prefix
	<i>seconds</i>	Transitions time can be from 0 to 60 seconds with a default of 10.

Defaults 10 seconds.

Command Modes Global configuration.

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines Because the available circuit (AC) attribute of a destination is very dynamic, reporting of this attribute should be handled carefully. AC should be reported as frequently as possible so that the location server has better information about the resources. However, the location server should not be overwhelmed with too many updates.

A smoothing algorithm is applied to the quantity of AC being reported. This algorithm eliminates reporting of noise. The degree of smoothing can be configured with the **voice call capacity stw** command. This command sets the smoothing transition time for weight, which is the time it takes for current smoothed value of AC to come half way between the current smoothed value and the current instantaneous value of AC. Lower **stw** values speed the smoothed value of AC as it approaches the instantaneous value of AC. When **stw** is set to 0, the smoothed value is always equal to the instantaneous value of AC.

Examples The following example shows the smoothing time for weight for the carrier address family set to 25 seconds:

```
Router(config)# voice call carrier capacity stw 25
```

Related Commands	Command	Description
	capacity update interval (dial peer)	Changes the capacity update for prefixes associated with a dial peer.
	capacity update interval (trunk group)	Change the capacity update for carriers or trunk groups.
	voice call capacity mir	Set the value for MIR.

voice call capacity reporting

To turn on the reporting of maxima (first derivative) or inflection (second derivative) points in available capacity, use the **voice call capacity reporting** command in global configuration mode. To turn off the reporting, use the **no** form of this command.

```
voice call {carrier | trunk-group | prefix} capacity reporting {maxima | inflection}
```

```
no voice call {carrier | trunk-group | prefix} capacity reporting {maxima | inflection}
```

Syntax Description

carrier	Carrier code address family.
trunk-group	Trunk group address family.
prefix	E.164 prefix.
maxima	Maxima (first derivative) point in available capacity.
inflection	Inflection (second derivative) point in available capacity.

Defaults

The capacity reporting function is turned off.

Command Modes

Global configuration.

Command History

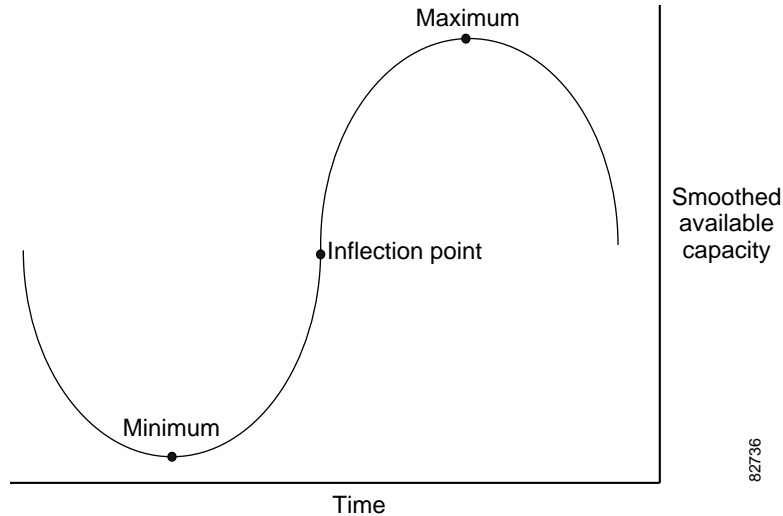
Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

The smoothed curve of the available circuits (AC) has maxima, minima, and inflection points. When the curve has reached these points, this represents a change in the call rate.

Maximum, minimum and inflection points are illustrated in [Figure 7](#).

Figure 7 Maximum, Minimum, and Inflection Points for Available Capacity



Examples

The following example shows the reporting of the available capacity inflection point on the trunk group is turned on:

```
Router(config)# voice call trunk-group capacity reporting inflection
```

Related Commands

Command	Description
voice call capacity mir	Sets the values for the minimum interval between reporting (MIR) and smoothing transition time for weight (STW).
voice call capacity timer interval	Sets the periodic interval for reporting capacity from carrier, trunk group, or prefix databases
voice call trigger hwm	Sets the value for percentage change, low water mark and high water mark in the available capacity in the trunk group or prefix databases.

voice call capacity timer interval

To set the periodic interval for reporting capacity from carrier, trunk group, or prefix databases, use the **voice call capacity timer interval** command in global configuration mode. To turn off the interval, use the **no** form of this command.

voice call {carrier | trunk-group | prefix} capacity timer interval *seconds*

no voice call {carrier | trunk-group | prefix} capacity timer interval *seconds*

Syntax Description		
carrier	Carrier code address family	
trunk-group	Trunk group address family	
prefix	E.164 prefix	
<i>seconds</i>	Value from 10 to 3600 seconds.	

Defaults 25 seconds

Command Modes Global configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines For the reporting interval, a periodic timer called the capacity update timer handles updates of available circuit (AC) information and can be configured using the **voice call capacity timer interval** command. For example, if AC has changed since the last reporting, the AC is again reported when the capacity update timer expires.

Examples The following example sets the timer interval for the prefixes set at 15 seconds:

```
Router(config)# voice call prefix capacity timer interval 15
```

Related Commands	Command	Description
	voice call capacity mir	Sets the values for the MIR and STW.
	voice call capacity reporting	Turns on the reporting of maxima (first derivative) or inflection (second derivative) points in available capacity.
	voice call trigger hwm	Sets the value for percentage change, low water mark and high water mark in the available capacity in the trunk group or prefix databases.

voice call csr data-points

To set the number of call success rate (CSR) data points, use the **voice call csr data-points** command in global configuration mode. To disable the setting of the CSR data points, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **csr data-points** *value*

no voice call { **carrier** | **trunk-group** | **prefix** } **csr data-points** *value*

Syntax Description

carrier	Carrier code address family
trunk-group	Trunk group address family
prefix	E.164 prefix
<i>value</i>	Value from 10 to 50 data points. Default is 30 data points.

Defaults

30 data points

Command Modes

Global configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Examples

The following example sets the CSR data points for trunk groups at 10:

```
Router(config)# voice call trunk-group csr data-points 10
```

Related Commands

Command	Description
voice call csr recording interval	Sets the recording interval for CSR.
voice call csr reporting interval	Sets the reporting interval for CSR.

voice call csr recording interval

To set the recording interval for call success rates (CSR), use the **voice call csr recording interval** command in global configuration mode. To disable the CSR recording interval, use the **no** form of this command.

voice call {carrier | trunk-group | prefix} csr recording interval *minutes*

no voice call {carrier | trunk-group | prefix} csr recording interval *minutes*

Syntax Description		
carrier	Carrier code address family.	
trunk-group	Trunk group address family.	
prefix	E.164 prefix.	
<i>minutes</i>	Value from 10 to 1000 minutes with a default of 60.	

Defaults 60 minutes

Command Modes Global configuration

Command History	Release	Modification
	12.3(1)	This command was introduced.

Examples The following example sets the CSR recording interval for prefixes at 30 minutes:

```
Router(config)# voice call carrier csr recording interval 30
```

Related Commands	Command	Description
	voice call csr data-points	Sets the number of call success rate (CSR) data points.
	voice call csr reporting interval	Sets the reporting interval for CSR.

voice call csr reporting interval

To set the reporting interval for call success rate (CSR), use the **voice call csr reporting interval** command in global configuration mode. To disable the CSR recording interval, use the **no** form of this command.

voice call {carrier | trunk-group | prefix} csr reporting interval *seconds*

no voice call {carrier | trunk-group | prefix} csr reporting interval *seconds*

Syntax Description

carrier	Carrier code address family.
trunk-group	Trunk group address family.
prefix	E.164 prefix.
<i>seconds</i>	Value from 10 to 10000 seconds with a default of 25.

Defaults

25 seconds

Command Modes

Global configuration

Command History

Release	Modification
12.3(1)	This command was introduced.

Examples

The following example sets the CSR reporting interval for trunk groups at 40 seconds:

```
Router(config)# voice call carrier csr reporting interval 40
```

Related Commands

Command	Description
voice call csr data-points	Sets the number of CSR data points.
voice call csr recording interval	Sets the recording interval for CSR.

voice call trigger hwm

To set the value for high water mark in the available capacity in the trunk group or prefix databases, use the **voice call trigger hwm** command in global configuration mode. To disable the trigger point, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **trigger hwm** *percent*

no voice call { **carrier** | **trunk-group** | **prefix** } **trigger hwm** *percent*

Syntax Description

carrier	Carrier code address family
trunk-group	Trunk group address family
prefix	E.164 prefix
<i>percent</i>	Value can be 50 to 100 percent with a default of 80. If set to 100, this trigger will be turned off.

Defaults

80 percent

Command Modes

Global configuration.

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

Available circuits are reported when the value of AC goes above a threshold, called the *high water mark*. This can be configured with the **voice call trigger hwm** command. When the **hwm** option is selected and the value is set to 100, no update is sent due to high water mark.

Examples

The following example sets the trigger for available capacity on trunk groups to send at a high water mark of 75%:

```
Router(config)# voice call trunk-group trigger hwm 75
```

Related Commands

Command	Description
voice call capacity mir	Sets the values for the minimum interval between reporting (MIR) and smoothing transition time for weight (STW).
voice call capacity reporting	Turns on the reporting of maxima (first derivative) or inflection (second derivative) points in available capacity.
voice call capacity timer interval	Sets the periodic interval for reporting capacity from carrier, trunk group, or prefix databases

■ voice call trigger hwm

Command	Description
voice call trigger lwm	Sets the value for low water mark in the available capacity for carrier, trunk group, or prefix databases
voice call trigger percent-change	Sets the value for percentage change in the available capacity for carrier, trunk group, or prefix databases

voice call trigger lwm

To set the value for low water mark in the available capacity in the trunk group or prefix databases, use the **voice call trigger lwm** command in global configuration mode. To disable the trigger point, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **trigger lwm** *percent*

no voice call { **carrier** | **trunk-group** | **prefix** } **trigger lwm** *percent*

Syntax Description	Parameter	Description
	carrier	Carrier code address family
	trunk-group	Trunk group address family
	prefix	E.164 prefix
	<i>percent</i>	Value can be 0 to 30 percent with a default of 10. If set to 0, this trigger will be turned off.

Defaults 10 percent

Command Modes Global configuration.

Command History	Release	Modification
	12.3(1)	This command was introduced.

Usage Guidelines Available circuits are reported when the value of AC falls below a threshold, called the *low water mark*. When the **lwm** option is selected and the value is set to 0, no update is sent due to low water mark.

Examples The following example sets the trigger for available capacity for E.164 prefixes to send at a low water mark of 25%:

```
Router(config)# voice call prefix trigger lwm 25
```

Related Commands	Command	Description
	voice call capacity mir	Sets the values for the minimum interval between reporting (MIR) and smoothing transition time for weight (STW).
	voice call capacity reporting	Turns on the reporting of maxima (first derivative) or inflection (second derivative) points in available capacity.
	voice call capacity timer interval	Sets the periodic interval for reporting capacity from carrier, trunk group, or prefix databases

Command	Description
voice call trigger hwm	Sets the value for high water mark in the available capacity for carrier, trunk group, or prefix databases
voice call trigger percent-change	Sets the value for percentage change in the available capacity for carrier, trunk group, or prefix databases

voice call trigger percent-change

To set the value for percentage change, low water mark and high water mark in the available capacity in the trunk group or prefix databases, use the **voice call trigger** command in global configuration mode. To disable the trigger point, use the **no** form of this command.

voice call { **carrier** | **trunk-group** | **prefix** } **trigger percent-change** *percent*

no voice call { **carrier** | **trunk-group** | **prefix** } **trigger percent-change** *percent*

Syntax Description

carrier	Carrier code address family
trunk-group	Trunk group address family
prefix	E.164 prefix
<i>percent</i>	<p>If percent-change is selected, value can be 0 to 100 percent with a default of 30. If set to 0, this trigger will be turned off.</p> <p>If lwm is selected, value can be 0 to 30 percent with a default of 10. If set to 0, this trigger will be turned off.</p> <p>If hwm is select, value can be 50 to 100 percent with a default of 80. If set to 100, this trigger will be turned off.</p>

Defaults

30 percent

Command Modes

Global configuration.

Command History

Release	Modification
12.3(1)	This command was introduced.

Usage Guidelines

Available circuits are reported when the absolute percent change is above a threshold. When the **percent-change** option is selected and the value is set to 0, no update for percent change is sent

Examples

The following example sets the trigger for available capacity on the carrier codes to send at a percentage change of 15%:

```
Router(config)# voice call carrier trigger percent-change 15
```

Related Commands

Command	Description
voice call capacity mir	Sets the values for the minimum interval between reporting (MIR) and smoothing transition time for weight (STW).
voice call capacity reporting	Turns on the reporting of maxima (first derivative) or inflection (second derivative) points in available capacity.

Command	Description
voice call capacity timer interval	Sets the periodic interval for reporting capacity from carrier, trunk group, or prefix databases
voice call trigger hwm	Sets the value for high water mark in the available capacity for carrier, trunk group, or prefix databases
voice call trigger lwm	Sets the value for low water mark in the available capacity for carrier, trunk group, or prefix databases

Glossary

ASP—Application Service Provider. A service provider that offers a unique application or service, typically hosted on another provider's core network

CLEC—Competitive Local Exchange Carrier

CSPS—Cisco SIP Proxy Server

Domain—Within this feature, domain is used interchangeably with ITAD (see below). It is assumed that in most cases the service provider will configure the entire network to act as a single domain so that routes and routing policies are consistent across his network. However, there may be some cases where service providers wish to configure multiple domains within their network representing major geographical regions.

E-TRIP—TRIP communication between location servers within different ITADs.

ILEC—Incumbent Local Exchange Carrier

Inter-domain—Refers to routes exchanged between location servers in different ITAD's.

Intra-domain—Refers to routes exchanged among the locations servers within a single ITAD.

ITAD—IP Telephony Administrative Domain. The set of resources (gateways, location servers, etc.) under control of a single administrative authority.

ITSP—Internet Telephony Service Provider. VoIP based network providers offer telephony service or voice transport.

I-TRIP—TRIP communication between location servers within an ITAD

IXC—Inter-Exchange Carrier. Service providers traditionally focused on long distance voice transport.

Local Route—Routes injected into TRIP at a location server are considered a "local route" by that location server. These routes may be injected by TGREP or by manual means. Routes received from another TRIP peer through I-TRIP or E-TRIP are not considered local routes.

LS—Location server. A centralized resource (e.g. SIP Proxy, SIP Redirect, H.323 Gatekeeper) that identifies the next hop for the voip connection and communicates that to the originating gateway or terminal.

Proxy Server—A SIP entity used for signaling and routing calls within a SIP network

PTT—Post, Telephone and Telegraph. A European national version of an ILEC.

RIB—Routing Information Base. Local routing table, maintained by the location server to reflect reachability and route attributes learned via routing protocol updates.

Route Aggregation—Route aggregation is the summarization of routes which fall within a class-based network boundary or within a class-less (CIDR) prefix length into a single announcement from the TRIP speaker.

Route Server—Generic term for server based applications designed to provided advanced routing logic within VoIP networks. Cisco route servers include CSR, ARS and NAM.

Static Route—Static routes are a concept specific to the proxy server. These are routes that have been manually provisioned into the proxy server. Generally, there is no dynamic information about the existence, availability, or capacity of this route or it would have been discovered using TGREP. Based on policy within the location server, these static routes may be injected into TRIP, in which case they are also considered local routes to that location server.

TRIP—Telephony Routing over IP. IETF draft defining how reachable telephony routes can be communicated across a VoIP network

TGREP—Telephony gateway registration protocol that allows the gateway to send updates to the TRIP system as local PSTN routes and resources disappear and reappear. TRIP updates may be preferable to H.323 or SIP registration for adding and withdrawing individual routes.

**Note**

Refer to the [Internetworking Terms and Acronyms](#) for terms not included in this glossary.
