



VoIP Gateway Trunk and Carrier Based Routing Enhancements

Feature History

Release	Modification
12.2(11)T	This feature is introduced for H.323 and SIP interfaces on the Cisco 2600 series, Cisco 3600 series, Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, and Cisco AS5850.
12.3	This feature was integrated into Cisco IOS Release 12.3.

This feature module describes the VoIP Gateway Trunk and Carrier Based Routing Enhancements feature functionality in Cisco IOS Release 12.2(11)T, and includes the following sections:

- [Feature Overview, page 2](#)
- [Supported Platforms, page 25](#)
- [Supported Standards, MIBs, and RFCs, page 25](#)
- [Prerequisites, page 26](#)
- [Configuration Tasks, page 27](#)
- [Configuring Call Capacity Reporting for Unsolicited Calls, page 45](#)
- [Troubleshooting VoIP Gateway Trunk and Carrier Based Routing, page 46](#)
- [Configuration Examples, page 51](#)
- [Command Reference, page 60](#)
- [Glossary, page 177](#)

For information on routing enhancements for gatekeepers, see *VoIP Gatekeeper Trunk and Carrier Based Routing Enhancements*.



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

Copyright © 2004 Cisco Systems, Inc. All rights reserved.

Feature Overview

Voice wholesalers use multiple ingress and egress carriers to route traffic. A call coming in to a gateway on a particular ingress carrier must be routed to an appropriate egress carrier. As networks grow and become more complicated, the dial plans needed to route the carrier traffic efficiently become more complex and the need for carrier sensitive routing (CSR) increases.

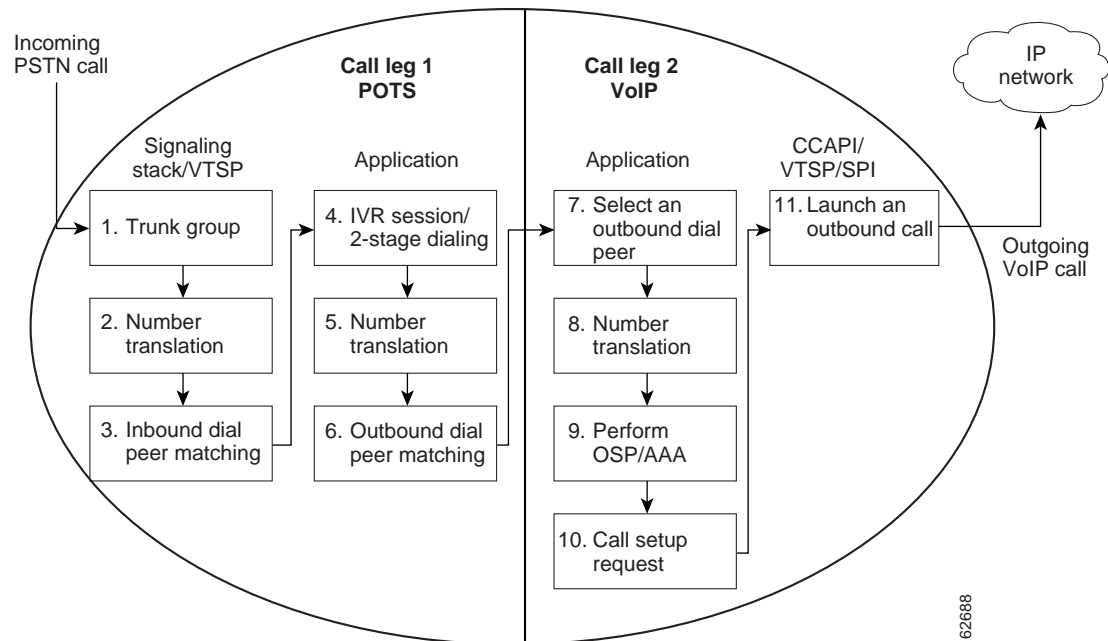
The Gateway Trunk and Carrier Based Routing Enhancements feature implements CSR for Cisco voice gateways. The gateway feature described in this document adds the following routing features:

- Implementation of trunk groups and enhanced key matches on several platforms and interfaces
- Reduction of the number of dial peers in a dial plan by using profile aggregation and multiple trunk group supports
- Enhanced hunting schemes
- Call capacity updates on carriers and trunk groups
- Carrier ID support
- Trunk group label support
- Number translation profiles per trunk group, source IP group, voice port, and dial peer
- Dial peer support of multiple trunk groups with translations per trunk group
- E.164 telephone number mapping (ENUM) support
- Source IP groups
- VoIP access list control
- Enhanced translation rules in SED (stream editor) regular expressions
- Incoming call blocking
- Cisco IVR 2.0 support for carrier ID based dial peer matching, incoming call blocking, and dial peer number translation
- Call detail record (CDR) support
- Virtual private network (VPN) source routing (also referred to as static or basic carrier routing)

The following call handling sequence gives some orientation to these new capabilities.

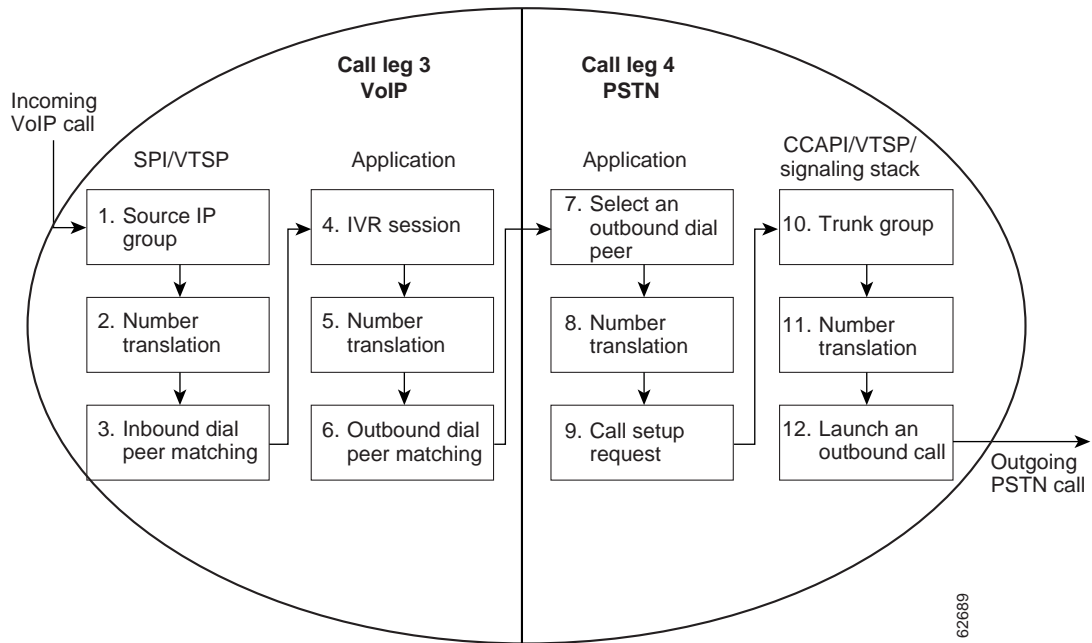
In a typical scenario, a call from the PSTN arrives at a gateway (the ingress gateway), leaves the gateway as a VoIP call, arrives at a destination gateway (the egress gateway), and leaves that gateway as a PSTN call. For this example, trunk groups and translation rules have been defined using commands described later in this document. Figure 1 illustrates what happens to the call in the ingress gateway. Figure 2 illustrates the call handing activities in the egress gateway. Steps affected by the trunk groups and translation rules implemented in this feature are marked as “new”.

Figure 1 Call Processing on the Ingress (Originating) Gateway
Originating Gateway



PSTN Side	VoIP Side
1. A PSTN call arrives from a trunk group that was assigned to physical interfaces during configuration. This is call leg 1.	7. The gateway selects an outbound dial peer from a list of matched dial peers.
2. (new) The gateway associates the incoming call with a carrier ID or trunk group label based on the trunk group configuration. The gateway checks the maximum call limit. The gateway translates the called, calling, and redirect numbers using the incoming translation rule of the trunk group.	8. (new) The gateway translates the numbers using the outgoing translation rule of the outbound dial peer. If the session target is set to ENUM, the gateway performs an ENUM lookup on the translated number.
3. The gateway matches the translated numbers or the source carrier ID or source trunk group label to an inbound dial peer. The gateway checks for incoming call blocking on the dial peer.	9. (new) If Open Settlements Protocol (OSP) is configured on the gateway, the gateway performs OSP on the translated call numbers. The OSP token contains the final translated call numbers.
4. The gateway starts a default or IVR session for the incoming call.	10. The gateway makes a call setup request. If the outgoing call is going to be routed using H.323 or SIP, the gateway sends the call's source carrier ID to the terminating gateway.
5. (new) The gateway translates the numbers again using the incoming translation rule of the inbound dial peer.	11. The outbound VoIP call goes out. This is call leg 2.
6. The gateway uses the newly translated called numbers to match an outbound dial peer. At the same time, the gateway performs AAA operations on the translated call numbers.	

Figure 2 Call Processing in the Egress (Terminating) Gateway
Terminating Gateway



VoIP Side	PSTN Side
1. (new) A VoIP call arrives from a source IP group. This is call leg 3. Depending on how the source IP group is defined, the incoming H.323 or SIP call may be matched to a source carrier ID, source trunk-group label, source IP address, or (for H.323 calls only) source H.323 zone ID. The gateway uses the source IP group to associate the call with a target carrier ID or target trunk-group label for matching the outbound dial peer.	7. The gateway selects an outbound dial peer.
2. (new) The gateway translates the incoming call numbers using the incoming translation rule of the source IP group.	8. (new) The gateway translates the numbers using the outgoing translation rule of the outbound dial peer.
3. The gateway matches the translated numbers to an inbound dial peer and checks for incoming call blocking on the dial peer.	9. The gateway makes a call setup request.
4. The gateway starts a default or IVR session for the incoming call.	10. (new) The gateway selects an interface from a trunk group to make an outbound call. The interface is selected based on the hunt scheme configured for the trunk group.
5. (new) The gateway translates the numbers again using the incoming translation rule of the inbound dial peer.	11. (new) The gateway translates the number using the outgoing translation rule of a trunk group.
6. The gateway uses the newly translated number, the target carrier ID, or the target trunk-group-label to match an outbound dial peer.	12. The outbound PSTN call goes out. This is call leg 4.

The components of the scenarios described in Figures 1 and 2 are detailed in the following sections:

- [Trunk Groups, page 5](#)
- [Carrier IDs, page 12](#)
- [Trunk Group Labels, page 12](#)
- [Number Translation, page 12](#)
- [Multiple Trunk Group Support, page 13](#)
- [ENUM Support, page 14](#)
- [Source IP Groups, page 17](#)
- [Translation Rules and Translation Profiles, page 18](#)
- [Incoming Call Blocking, page 20](#)
- [Call Detail Record Report, page 22](#)

Trunk Groups

This software feature provides these characteristics for a trunk group:

- A trunk group is a logical grouping of interfaces with the same signaling characteristics.
- The trunk group can be configured as the target of an outbound dial peer.
- A dial-peer can have multiple trunk groups configured for it.
- Up to 1000 trunk groups can be configured on the gateway provided the gateway has sufficient memory available.
- The real-time call capacity information for a trunk group is sent to the H.323 gatekeeper using the H.323 Version 4 protocol.
- A trunk group resource manager selects a voice port from the trunk group to make an outgoing call.

In addition to supporting existing trunk group functionality for PRI and BRI interfaces, this trunk and carrier routing enhancements feature adds trunk group support for these interfaces:

- T1/E1 DS-0 group (FXS, FXO, E&M, DID, R2 digital/pulse/analog)
- FXS, FXO, E&M, and DID analog voice ports
- NFAS PRI T1/E1

This feature adds trunk group support for Non-Facility Associated Signaling (NFAS) interfaces. The interfaces should be added as members of a trunk group to control incoming and outgoing call routing. All the interfaces of an NFAS group must belong to the same trunk group. Multiple NFAS groups can belong to the same trunk group with other PRI and BRI interfaces.

[Table 1](#) lists the different voice platforms and their supported trunk group interfaces.

Table 1 *Trunk Group Support on Cisco Platforms*

Voice Platform	Current T1/E1 Trunk Group Support	Expanded T1/E1/R2 Trunk Group Support	Expanded DS-0 T1/E1 Trunk Group Support	Expanded Voice Port Trunk Group Support
Cisco 1750	ISDN (PRI, BRI)	—	—	—
Cisco 26xx	ISDN(PRI, BRI)	ISDN NFAS-group CAS	Yes	FXS, FXO, E&M, DID

Table 1 Trunk Group Support on Cisco Platforms (continued)

Voice Platform	Current T1/E1 Trunk Group Support	Expanded T1/E1/R2 Trunk Group Support	Expanded DS-0 T1/E1 Trunk Group Support	Expanded Voice Port Trunk Group Support
Cisco 36xx	ISDN(PRI, BRI)	ISDN NFAS-group CAS	Yes	FXS, FXO, E&M, DID
Cisco AS5300	—	ISDN (PRI) ¹ ISDN NFAS-group CAS	—	—
Cisco AS5350	—	ISDN (PRI) ¹ ISDN NFAS-group CAS	—	—
Cisco AS5400	—	ISDN (PRI) ¹ ISDN NFAS-group CAS	—	—
Cisco AS5800	ISDN(PRI) ¹	ISDN NFAS-group CAS	—	—
Cisco AS5850	—	ISDN (PRI) ¹ ISDN NFAS-group CAS	—	—
Cisco MC3810	ISDN (PRI, BRI)	—	—	—

1. BRI is not supported on the Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, and Cisco AS5850 platforms.

The trunk group functionality in this feature does not support the existing trunk group functionality on the Cisco 1750 and Cisco MC3810 platforms.

Hunt Schemes

A hunt scheme is a selection procedure for choosing an interface or voice port. A trunk group with several trunk group members uses a hunt scheme to select an idle channel for routing an outgoing call. Several hunt schemes are supported, as illustrated with the following example.

Suppose a trunk group has three trunk group members: A has the smallest preference value, B has the next highest preference value, and C has the highest preference value. Each hunt scheme impacts the trunk group members as described in the following sections.

Least-Used

The software selects the trunk group member that has the highest number of available channels. This high number indicates that the trunk group member is not used as often as other members. If two or more trunk group members have the same number of idle channels, the one with the highest preference (lowest preference value) is searched first. Once the member is selected, the software searches for an idle channel. Optional parameters can modify the search to look in either ascending or descending order for an even-numbered channel, an odd-numbered channel, or either type.

Suppose **hunt-scheme least-used even down** is enabled. The search goes through the trunk group members in descending order (C, B, A) to determine which member has the highest number of even-numbered idle channels. After selecting that trunk group member, the search looks for an even-numbered idle channel. If successful, the search selects an even-numbered idle channel to use for routing the call. If unsuccessful, the search goes through the trunk group members in the same descending order to select an odd-numbered idle channel. If successful, the search selects an odd-numbered idle channel for routing the call.

Least-Idle

The software searches across all the channels in A, B, and C for the channel that has most recently become available. The precedence of A, B, and C is not important because more than two or more channels cannot be least-idle at the same time. Optional parameters can modify the search to look for an even-numbered channel, an odd-numbered channel, or either type.

Suppose **hunt-scheme least-idle even** is enabled. The software searches for an even-numbered channel that has just entered the available queue. An even-numbered channel in C has just become available. That channel is used for the call routing.

If no even-numbered channel is available, the software searches for the odd-numbered channel with the *longest* idle time instead.

Longest-Idle

The software searches across all the channels in A, B, and C for the channel that has been in the available queue the longest. The precedence of A, B, and C is not important because two or more channels cannot be longest-idle at the same time. Optional parameters can modify the search to look for an even-numbered channel, an odd-numbered channel, or either type.

Suppose **hunt-scheme longest-idle odd** is enabled. The software searches for the odd-numbered channel that has been in the idle queue the longest. An odd-numbered channel in B has been available for the longest time. That channel is used for the call routing.

If no odd-numbered channel is available, the software searches for the even-numbered channel with the *least* (shortest) idle time instead.

Random

The trunk group members are searched in random order for an idle channel.

This method does not account for one member being busier than another and does not intentionally balance the call load across all members. (In the long term, load balancing is achieved across all members.) The search can not be modified for even- or odd-numbered channels or for ascending or descending order.

Round-Robin

The trunk group members are searched in turn for an idle channel. The history of the previously selected member is saved to identify the next trunk group member to use for a new idle channel request. This method tries to balance the load of channel use across the trunk group members.

Optional parameters can modify the search to look for an even-numbered channel, an odd-numbered channel, or either type. The trunk group members are searched in order of preference.

Suppose **hunt-scheme round-robin even** is enabled. Trunk group member A is searched first because it has the highest preference. The search looks for an even-numbered idle channel in A. If one is available, that channel is used for the call routing. A new idle channel request would start with member B, which has the next highest precedence value.

If A does not have an available even-numbered channel, the search tries to find an even-numbered channel in the next highest trunk group member, which is B. If successful, that channel is used for the call routing. A new idle channel request would start with C.

If B does not have any available even-numbered channels, the search tries to find an even-numbered channel in the next highest trunk group member, which is C. If successful, that channel is used for the call routing. A new idle channel request would start with A.

If C has no available even-numbered channels, the search repeats the process to find an odd-numbered channel. In this instance, the search would start again with A.

Sequential

This hunt scheme is similar to round-robin, except that a new idle channel request always starts its search with the highest precedence trunk group member, regardless of the member used for the previous request. In our example, A would always be the first trunk group member searched, even if B was successful the last time.

Optional parameters can modify the search to look in either ascending or descending order for an even-numbered channel, an odd-numbered channel, or either type.

Suppose **hunt-scheme sequential odd down** is enabled. Trunk group member A is searched first because it has the highest precedence value. The search checks A's channels in descending order for an odd-numbered available channel. If successful, the channel is used for the call routing. The next idle channel request starts its search with A.

If A does not have an available even-numbered channel, the search tries to find an even-numbered channel in the next highest trunk group member, which is B. If successful, that channel is used for the call routing. A new idle channel request would start with A.

If B does not have any available even-numbered channels, the search tries to find an even-numbered channel in the next highest trunk group member, which is C. If successful, that channel is used for the call routing. A new idle channel request would start with A.

If C has no available even-numbered channels, the search repeats the process to find an odd-numbered channel. In this instance, the search would start again with A.

Call Capacity Reporting Updates

The gateway automatically sends call capacity information for H.323 calls to the gatekeeper using the H.323 Version 4 protocol. The gateway receiving the H.323 call sends the report as soon as the call arrives.

SIP and TDM calls (non-H.323 calls) arriving at a gateway do not generate an automatic update to the gatekeeper's capacity database. The gateway must be configured to send call capacity updates for these calls at user-defined regular intervals to the gatekeeper.

When periodic call capacity reporting is enabled, the carrier resource manager (CRM) subsystem on the gateway scans the list of carrier IDs or trunk group labels at the user-specified time interval. If the capacity information has changed but has not been reported, the CRM sends a message to the H.323 subsystem on the gatekeeper with the updated information.

If periodic call capacity reporting is enabled, the gateway and gatekeeper call capacity information eventually converges (if no calls connect or disconnect). The following examples illustrate this process.

Sample Call Capacity Updates

Suppose a gatekeeper GK-A has two registered gateways, GW1 and GW2. The two gateways use VoIP to communicate with each other. GW1 uses carrier 1 with a capacity of 23 channels. GW2 uses carrier2 with a capacity of 24 channels. Both gateways have periodic call capacity reporting enabled with a timer interval of 30 seconds.

If no calls are active, the gatekeeper and gateways have the following information:

GK-A# **show gatekeeper circuits**

Circuit	Endpoint	Max Calls	Avail Calls	Resources	Zone
carrier2	Total Endpoints: 1				
	GW2	24	24	Available	
carrier1	Total Endpoints: 1				
	GW1	23	23	Available	

GW1# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier1
Max calls:24
Max Voice (in) :      23 Cur Voice (in) :      0
Max Voice (out):      23 Cur Voice (out):      0
Max Data (in)  :      23 Cur Data (in)  :      0
Max Data (out) :      23 Cur Data (out) :      0
```

GW2# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier2
Max calls:23
Max Voice (in) :      24 Cur Voice (in) :      0
Max Voice (out):      24 Cur Voice (out):      0
Max Data (in)  :      24 Cur Data (in)  :      0
Max Data (out) :      24 Cur Data (out) :      0
```

Now suppose that six calls are made from GW2 to GW1. Each call starts 30 seconds after the previous call was initiated and each call lasts five minutes. The call capacity reporting interval is 30 seconds.

The following output illustrates the gatekeeper and gateway call capacity information 90 seconds after the first call started:

GK-A# **show gatekeeper circuits**

Circuit	Endpoint	Max Calls	Avail Calls	Resources	Zone
carrier2	Total Endpoints: 1				
	GW2	24	21	Available	
carrier1	Total Endpoints: 1				
	GW1	23	19	Available	

GW1# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier1
Max calls:23
Max Voice (in) :      23 Cur Voice (in) :      4
Max Voice (out):      23 Cur Voice (out):      0
Max Data (in)  :      23 Cur Data (in)  :      0
```

```
Max Data (out) :      23 Cur Data (out) :      0
```

```
GW2# show crm
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier2
Max calls:24
Max Voice (in) :      24 Cur Voice (in) :      0
Max Voice (out):      24 Cur Voice (out):      4
Max Data (in)  :      24 Cur Data (in)  :      0
Max Data (out) :      24 Cur Data (out) :      0
```

At 90 seconds, GW2 has sent out 4 calls, GW1 has received 4 calls, but the fourth call is not yet connected on the PSTN side of GW2. When the scan sends the call capacity update to the gatekeeper, GK-A shows only 3 active calls (21 available channels) for GW2. At the next reporting interval, the scan will show 4 active calls (20 available channels) for GW2.

The last call is connected (after 150 seconds) and all the calls are still active. A snapshot of the call capacity information taken at 180 seconds shows all the statistics match up:

```
GK-A# show gatekeeper circuits
```

Circuit	Endpoint	Max Calls	Avail Calls	Resources	Zone
carrier2	Total Endpoints: 1				
	GW2	24	18	Available	
carrier1	Total Endpoints: 1				
	GW1	23	17	Available	

```
GW1# show crm
```

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier1
Max calls:23
Max Voice (in) :      23 Cur Voice (in) :      6
Max Voice (out):      23 Cur Voice (out):      0
Max Data (in)  :      23 Cur Data (in)  :      0
Max Data (out) :      23 Cur Data (out) :      0
```

```
GW2# show crm
```

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier2
Max calls:24
Max Voice (in) :      24 Cur Voice (in) :      0
Max Voice (out):      24 Cur Voice (out):      6
Max Data (in)  :      24 Cur Data (in)  :      0
Max Data (out) :      24 Cur Data (out) :      0
```

At 330 seconds (5.5 minutes), the first and second call are disconnected. The call capacity information at this point shows:

```
GK-A# show gatekeeper circuits
```

Circuit	Endpoint	Max Calls	Avail Calls	Resources	Zone
carrier2	Total Endpoints: 1				
	GW2	24	19	Available	
carrier1	Total Endpoints: 1				
	GW1	23	18	Available	

GW1# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier1
  Max calls:23
  Max Voice (in) :      23 Cur Voice (in) :      4
  Max Voice (out):      23 Cur Voice (out):      0
  Max Data (in)  :      23 Cur Data (in)  :      0
  Max Data (out) :      23 Cur Data (out) :      0
```

GW2# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier2
  Max calls:24
  Max Voice (in) :      24 Cur Voice (in) :      0
  Max Voice (out):      24 Cur Voice (out):      4
  Max Data (in)  :      24 Cur Data (in)  :      0
  Max Data (out) :      24 Cur Data (out) :      0
```

In this case, the gatekeeper has not received an update about the second call being disconnected, which is why the gatekeeper still shows 5 active calls (24-19 and 23-18) while the two gateways show 4 active calls. Once the PSTN on the terminating gateway (GW1) releases the channel, the next call capacity scan will show the release of the second channel.

At 450 seconds (7.5 minutes), all the calls are disconnected. A scan taken at this time shows the following output:

GK-A# **show gatekeeper circuits**

Circuit	Endpoint	Max Calls	Avail Calls	Resources	Zone
carrier2	Total Endpoints: 1				
	GW2	24	23	Available	
carrier1	Total Endpoints: 1				
	GW1	23	23	Available	

GW1# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier1
  Max calls:23
  Max Voice (in) :      23 Cur Voice (in) :      0
  Max Voice (out):      23 Cur Voice (out):      0
  Max Data (in)  :      23 Cur Data (in)  :      0
  Max Data (out) :      23 Cur Data (out) :      0
```

GW2# **show crm**

```
CRM periodic update enabled
CRM periodic updates at interval: 30 seconds
Carrier:carrier2
  Max calls:24
  Max Voice (in) :      24 Cur Voice (in) :      0
  Max Voice (out):      24 Cur Voice (out):      0
  Max Data (in)  :      24 Cur Data (in)  :      0
  Max Data (out) :      24 Cur Data (out) :      0
```

The gatekeeper shows GW2 with one active channel. GW2 did not receive the channel release indication from CRM in time for the call capacity scan to update the gatekeeper.

The next scan will show that all the call capacity statistics on the gatekeeper and gateways match.

Carrier IDs

A *carrier ID* is a new attribute consisting of up to 127 alphanumeric characters that identifies the carrier handling an H.323 or SIP call. One or more trunk groups can refer to the same carrier ID.

To support carrier ID routing, the egress (also called terminating) gateway uses the carrier ID routing tag as a matching key to select an outbound dial peer. In addition, the gateway forwards the call capacity information for the carrier IDs to the gatekeeper connected to a gatekeeper transaction management protocol (GKTMP) server application.



Note

Configure the gateway and the network for either carrier ID or trunk group routing. Using them together is not supported and will have unpredictable behavior.

Trunk Group Labels

The gateway trunk and carrier routing enhancements support trunk group routing by enabling a gateway routing tag called *trunk group label*. This new routing tag co-exists with the *carrier ID* gateway routing tag.

Carriers have the option of routing using either the trunk group label or carrier ID.



Note

Configure the gateway and the network for either carrier ID or trunk group routing. Using them together is not supported and will have unpredictable behavior.

Additional considerations:

- A trunk group label is a trunk group identifier of up to 127 alphanumeric characters.
- All trunk group labels and carrier IDs must be unique on the gateway. A trunk group label assigned to any trunk group must not match any carrier ID assigned to any trunk group on the gateway.
- Similarly, the source names in a source IP group and in dial-peers must be unique on the gateway. A trunk group label source name must not be the same as any carrier ID source name.

All other aspects of implementing carrier ID routing and trunk group label routing remain the same.

If a carrier ID is not configured for a trunk group, the call capacity reporting uses the trunk group label capacity. The capacity message includes an indication that the capacity's source is a trunk group label.

Number Translation

The gateway trunk and carrier routing enhancements implement translation profiles with translation rules in SED expression format. Trunk groups refer to these profiles for incoming and outgoing call number translation.

The ingress (or originating) gateway uses the translation profile to translate POTS call numbers coming in on a voice port or a trunk group. A voice port matches a single interface, DS-0 group, or analog port; a trunk group matches multiple interfaces, DS-0 groups, or analog ports. If the voice port does not have a translation profile but is a member of a trunk group that does have one, the gateway uses the trunk group's translation profile to translation the call. But if a voice port has a translation profile and is a member of a trunk group, the voice port translation profile overrides the trunk group profile.

A translation profile can be defined for one or more individual controllers, also called *Non-Facility Associated Signaling (NFAS) interfaces*, under voice services. In an NFAS group, all the interfaces share the same voice port, which forces all the interfaces to share the same translation profile. This translation profile translates incoming and outgoing calls on a per-controller basis. When an incoming or outgoing call seizes an NFAS B-channel, the translation profile for the controller overrides the translation profile of the voice port.

Previous number translation schemes performed the number translation *after* matching the incoming call with an inbound dial peer. The new number translation occurs *before* matching the inbound dial peer. Translating the call numbers before matching may affect the selection of an inbound dial peer.

After the incoming call is matched with an inbound dial peer, the translation of the dial peer occurs.

In the egress (or terminating) gateway, new number translation is performed on an incoming H.323 or SIP call using the incoming translation profile of a source IP group or a global VoIP incoming translation profile. If an incoming VoIP call is matched with a source IP group that has an incoming translation profile, the source IP group's profile overrides the global VoIP translation profile.

At the terminating end, the gateway matches the outbound call with an outbound dial peer, using one or more trunk groups as a target. The gateway translates the number using the outgoing translation profile of the outbound dial peer. If the outbound dial peer definition contains a voice port or trunk group, the gateway translates the number again based on the outgoing translation profile of the voice port, NFAS interface, or trunk group.

If the call is not routed successfully and the gateway needs to search for another outbound dial peer, the gateway uses the call number available before any previous outbound dial-peer translations. The call number is translated again based on the subsequent outbound dial-peer.

Multiple Trunk Group Support

Multiple trunk group support permits up to 64 trunk groups to be provisioned as a target in a POTS dial-peer. Because the dial peer can have more than one target destination, this capability reduces the number of dial peers that need to be configured.

During a call setup request for an outbound call, the software searches for an idle channel in an outbound dial peer with a list of trunk groups. The trunk groups are searched sequentially by priority. If an idle channel is not available from the highest priority trunk group, the next priority trunk group is searched. When a channel is found, the trunk group member containing the channel is used for the outbound call.

If the call setup returns a glare condition and the software attempts a retry for the call, the search for another idle channel starts from the beginning of the list of multiple trunk groups.

The following examples illustrate the method for selecting a trunk group.

Example 1

In Example 1, trunk group 11 is searched first because it has the highest priority.

```
dial peer voice 102 POTS
  carrier-id target xyz
  trunkgroup 11 1
  trunkgroup 12 2
  trunkgroup 13 3
  translation-profile outgoing 1
```

Example 2

In Example 2, trunk group 12 has no specified preference, so its priority is assumed to be 0, which is the highest preference. Therefore, trunk group 12 is searched first because it has the highest priority. If an idle channel is not available in trunk group 12, trunk group 11 is searched next.

```
dial peer voice 103 POTS
  carrier-id target xyz
  trunkgroup 12
  trunkgroup 11 1
  trunkgroup 13 2
  translation-profile outgoing 1
```

Example 3

In Example 3, trunk group 12 and trunk group 11 have no specified preference, so their priority is assumed to be 0. Trunk groups with equal priority are selected in the order as they appear listed in the dial peer. Therefore, trunk group 12 is searched first because it was configured before trunk group 11. If an idle channel is not available in trunk group 12, trunk group 11 is searched next.

```
dial peer voice 104 POTS
  carrier-id target xyz
  trunkgroup 12
  trunkgroup 11
  trunkgroup 13 1
  translation-profile outgoing 1
```

ENUM Support

The supported implementation of ENUM (E.164 telephone number mapping) for gateway trunk and carrier routing uses a set of user-defined rules to translate the called number into a DNS name for finding an outgoing VoIP dial-peer. Each rule requires a matching pattern, replacement rule, E.164 domain name, and preference number. The rules are collected into a table in order of their preference numbers (the lowest integer is the highest preference).

ENUM Call Flow

The process for handling an ENUM call is described below.

1. Users who want ENUM service must register their URLs into DNS service. These URLs identify each user as a destination for a call and may represent various access services, such as SIP, H.323, telephone, fax, email, instant messaging, and personal web pages.
2. The person initiating a call dials a telephone number, such as 555-1212. This is the called, or dialed number identification service (DNIS), number.
3. The gateway uses the DNIS number to match an outgoing VoIP dial peer. This dial peer must be configured with an ENUM match table containing rules that the software uses to convert the DNIS number into an E.164 number.
4. The gateway converts this called number into an E.164 number, such as +1-408-555-1212. See the next section, [“ENUM Number Conversion”](#), for an explanation of the conversion process.
5. The gateway converts the E.164 number into a fully qualified domain name (FQDN) according to the naming authority pointer (NAPTR) algorithm. For example, the number might look like 2.1.2.1.5.5.5.8.0.4.1.e164.arpa.
6. The gateway sends the FQDN to the DNS server.
7. If the DNS server finds a match for the FQDN, the server returns a NAPTR resource record (RR). The response includes one or more RRs needed to reach the called party. See the [“NAPTR Resource Records”](#) section on page 16 for an explanation of NAPTR RRs.

If the DNS server cannot find a match for the FQDN, the translation process repeats in the ENUM match table using the matching pattern with the next highest preference (step 3 above).

8. Upon receiving the response from the DNS server, the initiating party can make the call directly using the URLs in sequential order. The call attempts use only those URLs appropriate for the session protocol.

ENUM Number Conversion

The gateway uses the ENUM match table configured for the outgoing VoIP dial peer to select a rule for converting the DNIS, or called, number into an E.164 number. The ENUM match table has one or more translation rules containing matching and replacement patterns.

The DNIS number is matched against the matching pattern of the rule with the highest preference. If the called number does not match the first rule, the software looks at the rule with the next highest preference. When a match is found, the called number is translated according to the replacement rule. The result is an E.164 number.

The NAPTR algorithm removes all characters except digits from the E.164 number, reverses the digits, and inserts dots between each digit. The domain name, specified in the ENUM rule, is concatenated to the translated number. This concatenated number forms the DNS name, or FQDN, which the DNS server uses to find the URL of the call's destination. (Refer to RFC 2916 for more information on DNS names and searches.) See the [“NAPTR Resource Records” section on page 16](#) for an explanation of the NAPTR RRs returned by the DNS server.

ENUM Match Table Rules

An ENUM match table contains one or more rules for converting called (DNIS) numbers into E.164 numbers, which are used by the DNS server for locating a destination target. ENUM supports a maximum of 15 search tables, each with its own set of rules.



Note Remember to configure an outgoing VoIP dial peer with a session target set to `enum: enum-match-table-number`.

Rules can be entered into a table in any order. Each rule contains an identification number, a preference number, a matching pattern, a replacement pattern, and a domain name. The matching and replacement patterns are SED-like expressions. The preference numbers determines how the software searches through the table of rules to find a match. The search sequence can be changed easily by changing the preference numbers of the rules.

To illustrate how this works, suppose the set of rules in [Table 2](#) defines a match table:

Table 2 *ENUM Match Table Entries*

Matching Pattern	Replacement Pattern	Domain Name	Preference
<code>^(.*)/</code>	<code>^1/</code>	E164.arpa	6
<code>/^9(1.*)/</code>	<code>/+1/</code>	E164.cisco.com	1
<code>/^9011(.*)/</code>	<code>/+14081/</code>	E164.arpa	3
<code>/^201(.*)/</code>	<code>/+12011/</code>	E164.cisco.com	7

The software searches through them in order of preference:

Matching Pattern	Replacement Pattern	Domain Name	Preference
/^9(1.*\)/	/+1/	E164.cisco.com	1
/^9011(.*\)/	/+14081/	E164.arpa	3
^(.*\)/	/1/	E164.arpa	6
/^201(.*\)/	/+12011/	E164.cisco.com	7

Suppose a call comes in with a called number of “90115325755”. The processing sequence is:

1. Compare the number to the matching patterns in the table. It matches the second rule (the one with preference number “3”).
2. The replacement pattern translates the number into “+14085325755”.
3. The digits are reversed and the domain name “e164.arpa” is attached to the number to form the FQDN, or DNS name. Using the guidelines of RFC 2916, the DNS name becomes “5.5.7.5.2.3.5.8.0.4.1.e164.arpa”.
4. The expanded number is sent to the DNS server handling destination targets.

If this DNS name does not result in a destination URL for the call, the translation process begins again, starting with the third rule (the one with preference number “6”).

NAPTR Resource Records

A NAPTR RR is the response from the DNS server after it has located a destination target for the called number. Following are sample RRs.

```

                                ord pr fl  service  regexp replacement
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 10 10 "U"  "h323+E2U"  "/^.*$/h323:5551212@10.1.1.1
/" .
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 30 10 "U"  "sip+E2U"   "/^.*$/sip:5551212@reno.vegas
.com/" .
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 40 10 "S"  "h323+E2U"  ". _potscall._udp.cisco.com.
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 50 10 ""  "sip+E2U"   "/^(.*)$/6661313/" .
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 60 10 "U"  "h323+E2U"  "/^.*$/tel:5551212/" .
2.1.2.1.5.5.5.8.0.4.1.e164.arpa.IN NAPTR 60 10 "U"  "sip+E2U"   "/^.*$/tel:5551212/" .

```

Table 3 explains the fields in the NAPTR RR.

Table 3 NAPTR RR Field Descriptions

Field	Description
ord	Order of the NAPTR record. Specifies the order in which the records should be processed.
pr	Preference of the NAPTR record. Specifies the order in which the records should be processed when records have the same order value.
fl	Flag of the NAPTR record. This modifier affects what happens in the next DNS lookup.
service	Specifies the resolution protocol and resolution service.
regexp	Regular expression used for the rewrite rule.
replacement	Replacement expression used for the rewrite rule.

The gateway sorts the records by order and preference. Then it validates the service, regexp, and replacement fields and applies the regular or replacement expression. The resulting string is the URL for the destination if the lookup is terminal; otherwise, the lookup continues with the resulting string.

Source IP Groups

A source IP group is associated with an incoming VoIP call so that the terminating gateway can initiate appropriate services, such as number translation and incoming call control. The source IP group can be identified using one of several possibilities, in the following order of precedence:

- Source carrier ID or Source trunk group label (these two identifiers have the same preference)
- Source zone ID (only for incoming H.323 calls)
- Source IP address—Several source IP addresses can be grouped together in an access list, which is associated with a source IP group.

A maximum of 1000 source IP groups can be configured, if the source IP groups do not include an access list. A maximum of 99 source IP groups can be configured using access lists, because each source IP group must refer to a unique access list numbered between 1 and 99.

Blocking Incoming VoIP Calls using Access Lists

Identifying the source IP group for an incoming VoIP call is done before selecting an inbound dial peer. If the source for the call is found using an access list, the rules that define the access list determine whether the call is accepted or rejected.

If the call is rejected, the gateway returns a default “NO SERVICE” message or a user-specified disconnect cause to the source. The following disconnect causes are supported:

- Invalid-number
- Unassigned-number
- User-busy
- Call-rejected

Number Translation

A source IP group may refer to a translation profile for incoming VoIP call number translation. Even if a global VoIP translation rule is defined, the source IP group’s translation profile is used instead.

The terminating gateway performs the number translation *before* matching the call to an inbound dial peer. This is a change from earlier Cisco IOS software releases that performed the number translation after matching the call to an inbound dial peer. Performing the number translation first may affect the results of the inbound dial peer matching.

Calls Originating from Other IP Sources

Source IP groups that refer to carrier IDs have special procedures to handle incoming H.323 calls that do not contain source carrier IDs or target carrier IDs. This situation occurs when a call comes in from a non-Cisco gateway or an IP client.

The terminating gateway tries to identify the source IP group using an IP address or a zone ID. The gateway saves the source and target carrier IDs of the source IP group and sends that information to its gatekeeper.

If the target carrier ID is “null,” the gatekeeper and its GKTMP server application try to find the target carrier ID using the source carrier ID. If the gatekeeper finds the target carrier ID, it sends the target carrier ID back to the gateway. If the gatekeeper cannot find the target carrier ID, it sends a “null” message back to the gateway.

If the terminating gateway receives a “null” message for the target carrier ID, the gateway uses only the destination pattern as the matching key to select an outbound dial peer.

Translation Rules and Translation Profiles

One of the new routing capabilities is the implementation of translation profiles to capture translation characteristics for a set of call numbers. These profiles work with translation rules to provide flexibility in routing calls.



Note

Cisco supports old translation rules configured with the **translation-rule** command and new translation rules configured with the **voice translation-rule** and **voice translation-profile** commands. Do not configure a gateway with both old and new translation rules, as this may cause unpredictable behavior.

The configuration sequence for defining translation rules and profiles is:

1. Define one or more translation rules.
2. Define a translation profile that refers to one or more of the translation rules.
3. Associate the translation profile with a group, voice port, interface, or dial peer.

The following sections explain these steps.

Translation Rules

The new set of translation rules has the following attributes:

- Follows SED-like regular expression matching. This feature converts number translation rules and operations in the older format into SED regular expressions.
- Supports escape sequences using backslashes.
- Supports SED-like regular expressions for the keywords “NULL” and “ANY.”
- Supports up to 15 rules per translation rule table.
- Supports up to 128 translation rules.
- Does not support the “|” regular expression character.
- Requires a CTRL-v before entering a “?” in order to save the “?” regular expression symbol as a match pattern.
- Supports both “&” and “\0” for copying the substring matched by the match pattern.



Note

The **rule (voice translation-rule)** command introduced in this feature is a subcommand of the **voice translation-rule** command. An earlier version of this command uses the same name but is a subcommand of the **translation-rule** command and has a slightly different command syntax. Going forward, we recommend that you use this newer version to define rules for call matching. Eventually, the **translation-rule** command will not be supported.

Table 4 illustrates some of the translation rules:

Table 4 Translation Rule Examples

Match Pattern	Replacement Pattern	Input String	Result String	Description
/^\$/	//			Null string to null string.
/^.*\$/	//	4084552711		Any string to null string.
//	//	4085551234	4085551234	Match any number but no replacement.
/^456\(.*/	/853\1/	4567123	8537123	Match from the beginning of the input string.
/^(^...)\456\(...)/	/\1853\2/	4084567777	4088537777	Match from the middle of the input string.
/^(.*)8920/	/\15555/	4088538920	4088535555	Match from the end of the input string.
/^1#\(.*/	/\1/	1#2345	2345	Replace match string with null string.
/^408...\(8333)/	/853\1/	4087778333	8538333	Match multiple patterns.
/1234/	/00&00/	5551234	55500123400	Matches the substring.
/1234/	/00\000/	5551234	55500123400	Matches the substring (same as &).

Each rule includes a precedence value, a match pattern, a replacement pattern, an optional number type, and an optional number plan. When a rule is invoked, the processing sequence is as follows:

1. The software looks for the rule with the highest precedence value. If the call number matches the match pattern, this rule's replacement pattern is used for the translation. If the call number does not match, the rule with the next highest precedence value is used. This checking continues until a rule is found that matches the call number. A translation rule table can contain a maximum of 15 rules.
2. The call is translated according to the replacement pattern.
3. If the optional number type (or plan) is included in the rule, the call's number and type (or plan) are checked against the match pattern and type (or plan) of the rule with the highest precedence. If both match, this rule is used for the translation. If one of the parameters does not match, the rule with the next highest precedence is checked. After a matching rule is found, the call's number is translated according to the replacement pattern and the type (or plan) is changed according to the rule's replacement type (or plan).

Translation Profiles

A translation profile defines the translation characteristics for a set of calls. Translation profiles associate translation rules for one or more of the following types of call numbers:

- Called
- Calling
- Redirect-called

Each type of call number in the translation profile can reference a different translation rule. Trunk and carrier based routing supports up to 1000 translation profiles.

Associating a Translation Profile with a Group or Port

After a translation profile is defined, it can be referenced by any of the following:

- Trunk group
 - The trunk group can have two different translation profiles for incoming and outgoing POTS calls.
 - If the trunk group has an outgoing translation profile, the number translation is performed during call setup.
- Source IP group
 - The source IP group can have a translation profile for incoming VoIP calls.
- Dial peer
 - The dial peer can have two different translation profiles for incoming and outgoing calls.
 - If the dial peer has an incoming translation profile, an IVR or session application invokes the number translation during the 2-stage or overlap dialing.
 - If the dial peer has an outgoing translation profile, an IVR or session application invokes the number translation before performing OSP/AAA or call setup.
- Voice port
 - A voice port can have a translation profile for incoming POTS calls. If the voice port is a member of a trunk group, the incoming translation profile of the voice port overrides the translation profile of the trunk group.
 - A voice port can have a translation profile for outgoing POTS calls. If the voice port is a member of a trunk group, the outgoing translation profile of the voice port overrides the translation profile of the trunk group.
- VoIP incoming
 - A global translation profile can be defined to translate all incoming VoIP calls. If an incoming H.323 call is associated with a source IP group that has a defined translation profile, the source IP group's translation profile overrides the global VoIP translation profile.

Incoming Call Blocking

Incoming call blocking is available to POTS, VoIP, VoATM, and VoFR dial peers. The gateway blocks the call during the session and IVR applications after the 2-stage dialing or 1-stage dialing is completed.



Note

The only option for call blocking is in the incoming direction. From the perspective of the voice gateway, the incoming direction can be either of the following:

- Incoming from a telephony device directly attached to a voice port on the gateway toward the gateway itself
- Incoming by way of an inbound VoX call from a peer gateway

To configure incoming call blocking, define a translation rule with a **reject** keyword. For example:

```
voice translation-rule 1
  rule 1 reject /408252*/
```

Apply the rule to a translation profile for called, calling, or redirect-called numbers, such as:

```
voice translation profile call_block_profile
  translate calling 1
```

Include the translation profile within a dial peer definition. For example:

```
dial-peer voice 111 pots
  call-block translation-profile incoming call_block_profile
  call-block disconnect-cause incoming invalid_number
```

In this example, the gateway blocks any incoming TDM (time-division multiplexing) call that successfully matches inbound dial-peer 111 and has a calling number that starts with 408252. The gateway also returns the disconnect cause “invalid number” to the source of the call. (Other disconnect causes can be assigned: unassigned-number, user-busy, or call-rejected.)

Following are two possible call-blocking scenarios:

Scenario 1: Block Inbound Calls from the PSTN/PBX/CO

We place the rejection profile right on the POTS dial peer that is associated with the voice port on which we expect the inbound call. When the inbound call attempt is made, we see in the CCAPI debugs that POTS dial-peer 9 is matched for the telephony call leg. The call-block rule is checked and we send back user-busy to the switch.

```
voice translation-rule 1
  rule 1 reject /9193927582/ <<<<----- filter out calls from this CallerID

voice translation-profile reject_ANI
  translate calling 1

dial-peer voice 9 pots
  destination-pattern 9T
  direct-inward-dial
  port 1/0:23
  call-block translation-profile incoming reject_ANI
  call-block disconnect-cause incoming user-busy
```

Scenario 2: Block Inbound VoX Calls from Using Local POTS Resources

We place the rejection profile right on a VoIP/VoATM/VoFR dial peer that matches an inbound VoX call attempt. When the inbound call attempt is made, we see in the CCAPI debugs that VoIP dial-peer 7 is matched for the IP call leg. The call-block rule is checked and we send back user-busy to the switch.

```
voice translation-rule 1
  rule 1 reject /9193927582/ <<<<----- filter out calls from this CallerID

voice translation-profile reject_ANI
  translate calling 1

dial-peer voice 7 voip
  destination-pattern 7T
  session target ipv4:A.B.C.D
  incoming called-number . <<<<----- force inbound IP call-leg match
  call-block translation-profile incoming reject_ANI
  call-block disconnect-cause incoming user-busy
```

Call Detail Record Report

The Call Detail Record (CDR) report has several new attributes.

Number Translations

The Gateway Trunk and Carrier Based Routing feature adds the following attributes to the call active and call history records to resolve billing issues caused by number translations:

- Original and translated called number
- Original and translated calling number
- Original and translated redirect number

GKTMP Server Application Routing

The Gatekeeper Trunk and Carrier Based Routing feature implements carrier sensitive routing functionality through the external GKTMP server application and is closely linked to the Gateway Trunk and Carrier Based Routing feature.

The gatekeeper feature adds the following attributes to the call active and call history records for POTS and VoIP calls:

- Source carrier ID
- Target carrier ID
- Source trunk-group-label
- Target trunk-group-label

SIP Gateway CSR

SIP CSR for a Single IP Carrier Domain

SIP CSR for a single IP Carrier Domain is useful to you if you do not have a CSR-enabled server or a CSR application in their SIP network. SIP CSR for a single IP Carrier Domain requires the same source IP group and dial-peer configuration enhancements introduced by the Gateway Trunk and Carrier Based Routing Enhancements feature. Also, a SIP VoIP dial peer requires no special configuration to use configured CSR trunk groups, other than specifying SIP as the session protocol on the outbound dial peer. See the [“Configuring an Outbound VoIP Dial Peer”](#) section on page 40 for the complete outbound dial peer configuration.

With SIP CSR, trunk groups are assigned a carrier ID. When a call arrives at an originating gateway, the gateway determines if there is a carrier ID associated with the particular interface group (DS0/trunk group/analog port) that the call came in on. If there is, then the SIP gateway matches the inbound dial-peer based on the previous carrier ID. The SIP gateway uses the outbound dial-peer matching rules and constructs an INVITE message containing the previous carrier ID and sends it to the terminating gateway.

The previous carrier ID is sent in the form of a via-extension parameter in the Via header of the INVITE message. The INVITE message may go through a proxy server before reaching the terminating gateway. The terminating gateway receives the INVITE message containing the previous carrier ID and extracts the previous carrier ID. The terminating gateway also uses it for source IP group matching and determines the target carrier ID. The gateway stores the previous and target carrier ID for later use in the outbound call leg.

During a call setup request for an outbound call, the outbound call leg selects an outbound POTS dial-peer based on the destination carrier ID. Selecting a dial peer allows the call to be routed out the appropriate port and trunk group.

All other aspects of a typical SIP call remain the same.

For backward compatibility purposes, the terminating gateway can receive an INVITE message from an originating gateway that is not CSR-aware. The previous carrier ID is not included in the INVITE message, but the terminating gateway tries to determine a target carrier to use for outbound dial-peer matching. If no target carrier can be determined, normal outbound dial-peer selection is performed.

x-route-tag Parameter

The carrier ID is sent to the terminating gateway in the form of a Via-extension parameter in the Via header of the SIP INVITE message. The Via-extension parameter can be the **x-route-tag** parameter. It is added if a source carrier ID is configured for the trunk group that the call came in on. The new parameter is ignored by proxy servers and gateways that are not CSR-aware. The terminating gateway parses the Via header and extracts the carrier ID. If it does not find the new parameter in the most recent Via header, it parses previous Via header until either the parameter is found, or all Via headers have been parsed. The parsing allows for SIP CSR support in SIP deployments that use a proxy server.

```
x-route-tag = "cid:carrierid@host",
where carrierid = carrierid (text) specified in trunk group configuration at the
originating gateway (OGW), and
host = ip address or fully qualified domain name (FQDN) of OGW
```

Example Via Header:

```
Via: SIP/2.0/UDP gw1.foo.com:5060;x-route-tag="cid:TrunkGroupName@gw1.foo.com"
```

tsp Parameter

The carrier ID can also be sent to the terminating gateway in the form of the **tsp** parameter. The **tsp** parameter is defined in RFC 2806 as a URL parameter.

```
tsp = "carrierid.fqdn",
where carrierid = carrierid (text) specified in trunk group configuration at the OGW,
and fqdn = fully qualified domain name of OGW
```

Example Via Header:

```
Via: SIP/2.0/UDP gw1.foo.com:5060;tsp="TrunkGroupName.gw1.foo.com"
```

The terminating gateway parses the Via header for the **x-route-tag** parameter. If the **x-route-tag** does not find one, it then parses for the **tsp** parameter.

Use of Escape Characters

If the source and target carrier IDs contain either the "@" or "." character, SIP gateways *escape* those characters when adding the **x-route-tag** parameter to an INVITE message. This is done so the gateway can recognize that those characters are part of the carrier ID and not a separator in the **x-route-tag** or **tsp** parameter. The gateway converts the escaped characters back into "@" or "." when extracting the carrier ID from the **x-route-tag** or **tsp** parameter of the INVITE message.

Benefits

Streamlined Configuration Process

Customer-defined translation profiles permit dial peers and interfaces to be configured with similar characteristics but without repetitive configuration steps.

Flexible Routing

The gateway trunk and carrier based routing enhancements feature focuses on flexible routing by implementing carrier sensitive routing using incoming and outgoing translation rules.

VoIP routes and POTS routes are treated in a more uniform manner using VoIP source IP groups and POTS trunk groups.

Dial Peer Enhancements for Routing

Trunk groups allow trunks to be grouped under dial peers for routing purposes, permitting more flexibility in doing routing.

Restrictions

Gateway trunk and carrier based routing does not support the following features:

- Statistical routing
- Summary range of numbers in SED regular expressions (this is a SED restriction)

Related Features and Technologies

- Voice over IP (VoIP)
- Gatekeeper Trunk and Carrier Based Routing Enhancements

Related Documents

General reference documents:

- [Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.2](#)
- [Cisco IOS Voice, Video, and Fax Command Reference, Release 12.2T](#)

Feature documents:

- [VoIP Gatekeeper Trunk and Carrier Based Routing Enhancements](#)

Platform documents:

- [Cisco 2600 Series product documentation](#)
- [Cisco 3600 Series product documentation](#)
- [Cisco AS5300 product documentation](#)
- [Cisco AS5350 product documentation](#)
- [Cisco AS5400 product documentation](#)
- [Cisco AS5850 product documentation](#)

Supported Platforms

- Cisco 2600 access router
- Cisco 3600 series multiservice platforms
- Cisco AS5300 universal access server
- Cisco AS5350 universal gateway
- Cisco AS5400 universal gateway
- Cisco AS5850 universal gateway

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, 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>

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

MIBs

- CISCO-VOICE-DIAL-CONTROL-MIB, with modifications to support the following new attributes in the dial peer:
 - Source Trunk Group Label
 - Target Trunk Group Label

- Source Carrier ID
- Target Carrier ID

The Session Target attribute is modified to support ENUM.

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

- RFC 2806, *URLs for Telephone Calls*
- RFC 2915, *The Naming Authority Pointer (NAPTR) DNS Resource Record*
- RFC 2916, *E.164 number and DNS*

Prerequisites

- Set up the gateways.
Refer refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2, to configure H.323 and SIP gateways.
- Prepare a dial plan for the trunk groups and dial peers.
Refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2, for information on dial plans.
- Prepare translation rules and translation profiles.
The *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2, has information on translation rules, but that information has been modified by this VoIP Gateway Trunk and Carrier Based Routing Enhancements feature. See the earlier section [Translation Rules and Translation Profiles](#), page 18, in this document for guidance on translation rules and translation profiles.
- Prepare source IP groups at the terminating gateways.
The *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2, has information on source IP groups, but that information has been modified by this VoIP Gateway Trunk and Carrier Based Routing Enhancements feature. See the earlier section [Source IP Groups](#), page 17, in this document for guidance on source IP groups.

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- [Configuring a Translation Rule](#) (required)
- [Verifying the Translation Rule Configuration](#) (optional)
- [Troubleshooting Tips for a Translation Rule Configuration](#) (optional)
- [Configuring a Translation Profile](#) (optional)
- [Configuring an ENUM Match Table](#) (optional)
- [Configuring the ENUM Server](#) (required if ENUM Match Table is configured)
- [Verifying the ENUM Match Table](#) (optional)
- [Configuring a Trunk Group](#) (required)
- [Verifying the Trunk Group Configuration](#) (optional)
- [Troubleshooting Tips for the Trunk Group Configuration](#) (optional)
- [Configuring a Source VoIP Group](#) (optional)
- [Verifying the Source VoIP Group Configuration](#) (optional)
- [Troubleshooting Tips for the Source IP Group Configuration](#) (optional)
- [Configuring a Voice Port](#) (optional)
- [Verifying the Voice Port Configuration](#) (optional)
- [Assigning Translation Profiles to Inbound Dial Peers](#) (optional)
- [Configuring Call Blocking on Inbound Dial Peers](#) (optional)
- [Configuring an Outbound POTS Dial Peer](#) (optional)
- [Verifying the Dial Peer Configuration](#)
- [Configuring an Outbound VoIP Dial Peer](#) (optional)
- [Assigning a Translation Profile to an NFAS Interface](#) (optional)
- [Configuring an NFAS Interface](#) (optional)
- [Verifying the NFAS Interface Configuration](#) (optional)
- [Troubleshooting Tips for the NFAS Interface Configuration](#) (optional)
- [Configuring a DS-0 Group](#) (optional)
- [Verifying the DS-0 Group Configuration](#) (optional)
- [Troubleshooting Tips for the DS-0 Group Configuration](#) (optional)
- [Configuring a Global Translation Profile for Incoming VoIP Calls](#) (optional)
- [Configuring Call Capacity Reporting for Unsolicited Calls](#) (optional)

Configuring a Translation Rule



Note

The **rule (voice translation-rule)** command introduced in this feature is a subcommand of the **voice translation-rule** command. An earlier version of this command uses the same name but is a subcommand of the **translation-rule** command and has a slightly different command syntax. Going forward, we recommend that you use this newer version to define rules for call matching. Eventually, the **translation-rule** command will not be supported.

Gateway trunk and carrier routing enhancements support a maximum of 128 translation rules. Each translation rule supports a maximum of 15 rule definitions. Follow these steps to configure a translation rule, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# voice translation-rule <i>number</i>	Enters voice-translation-rule configuration mode and initiates a translation rule definition.
Step 2	<p>Match and Replace Rule</p> <pre>Router(cfg-translation-rule)# rule <i>precedence</i> /<i>match-pattern</i>/ /<i>replace-pattern</i>/ [type {<i>match-type</i> <i>replace-type</i>} plan {<i>match-type</i> <i>replace-type</i>}] [plan {<i>match-type</i> <i>replace-type</i>}]</pre> <p>Reject Rule</p> <pre>Router(cfg-translation-rule)# rule <i>precedence</i> reject /<i>match-pattern</i>/ [type <i>match-type</i> plan <i>match-type</i>] [plan <i>match-type</i>]</pre>	<p>Defines the rule parameters for replacing specific call number patterns. The match pattern and replace pattern are SED expressions.</p> <p>Note The <i>replace-pattern</i> must be in a valid E.164 format that can include the acceptable special characters; otherwise, the expression is treated as an unrecognized command.</p> <p>Defines the rule parameters for rejecting a call number pattern.</p>
Step 3	Router(cfg-translation-rule)# exit	Ends the translation rule definition.

Verifying the Translation Rule Configuration

See the command descriptions later in this document for sample outputs and their explanation.

- Enter **show voice translation-rule** to display the characteristics of all translation rules defined on the gateway or the **show voice translation-rule** *number* to display a specific translation-rule. For example, suppose a rule was defined as

```
Router(config)# voice translation-rule 5
Router(cfg-translation-rule)# rule 1 /201/ /102/
Router(cfg-translation-rule)# exit
```

The output for the show voice-translation rule would be:

```
Router# show voice translation-rule 5
Translation-rule tag:5
  Rule 5:
  Match pattern: 201
  Replace pattern: 102
  Match type: none           Replace type: none
  Match plan: none          Replace plan: none
```

- Enter **test voice translation-rule** *number input-test-string* with a rule number and an input string to test the functionality of a translation-rule. For example:

```
Router# test voice translation-rule 5 2015551212

Matched with rule 5
Original number:2015551212   Translated number:1025551212
Original number type: none   Translated number type: none
Original number plan: none   Translated number plan: none

Router# test voice translation-rule 5 301
301Didn't match with anyof rules
```

- Enter **test voice translation-rule** *number input-test-string* to verify the format of the replacement pattern. For example:

```
Router(config)# voice translation-rule 10
Router(cfg-translation-rule)# rule 1 /^7/ /7/ type any national
Router(cfg-translation-rule)# end

Router# test voice translation-rule 1 700%7
Error: Couldn't translate number 700%7 using rule 1

Router# test voice translation-rule 100 700
Error:Ruleset 100 not found
```

Troubleshooting Tips for a Translation Rule Configuration

Enter **debug voice translation** to display the trace of the translation rule performance. For example, for a translation rule 5 with rule 1 /201/ /102/ and a test string of 2015551212, the debug output would look like the following.

```
Router# debug voice translation

VoIP Translation Rule debugging is enabled
00:51:56:regxrule_stack_pop_RegXruleNumInfo:stack=0x63DECAF4; count=3
00:51:56:regxrule_stack_push_RegXruleNumInfo:stack=0x63DECAF4; count=0
00:51:56:regxrule_profile_translate:number=2015551212 type=unknown plan=unknown
numbertype=calling
00:51:56:regxrule_profile_match:Matched with rule 1 in ruleset 5
00:51:56:sed_subst:Successful substitution; pattern=2015551212 matchPattern=201
replacePattern=102 replaced pattern=1025551212
00:51:56:regxrule_subst_num_type:Match Type = none, Replace Type = none Input Type =
unknown
00:51:56:regxrule_subst_num_plan:Match Plan = none, Replace Plan = none Input Plan =
unknown
```

Configuring a Translation Profile

Gateway trunk and carrier routing enhancements support a maximum of 1000 translation profiles. Follow these steps to define a translation profile. Use at least one of the calling types (called, calling, or redirect-called) in the translation profile, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# voice translation-profile <i>name</i>	Enters voice-translation-profile configuration mode and initiates the voice translation-profile definition identified by <i>name</i> .
Step 2	Router(cfg-translation-profile)# translate called <i>translation-rule-number</i>	(Optional) Associates a translation rule with called numbers.
Step 3	Router(cfg-translation-profile)# translate calling <i>translation-rule-number</i>	(Optional) Associates a translation rule with calling numbers.
Step 4	Router(cfg-translation-profile)# translate redirect-called <i>translation-rule-number</i>	(Optional) Associates a translation rule with redirected numbers.
Step 5	Router(cfg-translation-profile)# exit	Ends the voice translation-profile definition.

Configuring an ENUM Match Table

Follow these steps to configure an ENUM match table for outbound dial peers, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# voice enum-match-table <i>number</i>	Enters voice-enum-match-table configuration mode and initiates the ENUM match table definition.
Step 2	Router(config-enum)# rule <i>rule-number</i> <i>preference</i> [/match-pattern/ /replacement-pattern/ domain-name]	Defines a rule in SED expression format.
Step 3	Router(config-enum)# exit	Ends the voice enum-match-table configuration mode.

Configuring the ENUM Server

Follow these steps to configure the ENUM server, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip name-server <i>ip-address</i>	Designates the server with the specified IP address as the ENUM server.

Verifying the ENUM Match Table

See the command descriptions later in this document for sample outputs and their explanation.

- Enter **show voice enum-match-table** to display the characteristics of all ENUM match tables defined on the gateway or the **show voice enum-match-table** *number* to display a specific ENUM match table.

```
Router# show voice enum-match-table

voice enum-match-table 3
rule 1 5 /^9(1.*)/ /+\1/ cisco
```

```
rule 2 4 /^9011\(.*\)/ /+1408\1/ arpa
rule 10 1 /^(.*)/ /\1/ e164.cisco.com
```

```
voice enum-match-table 5
rule 2 4 /^9011\(.*\)/ /+1408\1/ arpa
rule 10 1 /^(.*)/ /\1/ e164.cisco.com
```

```
Router# show voice enum-match-table 5
```

```
voice enum-match-table 5
rule 2 4 /^9011\(.*\)/ /+1408\1/ arpa
rule 10 1 /^(.*)/ /\1/ e164.cisco.com
```

- Enter **test enum table-number input-pattern** with a table number and an input pattern to test the functionality of an ENUM match table. The command displays the output pattern for the given input pattern. For example:

```
Router# test enum 2 4085551212
```

Configuring a Trunk Group

Follow these steps to configure a trunk group, beginning in global configuration mode:



Note Up to 1000 trunk groups can be configured on a gateway provided the gateway has sufficient memory. Be careful of scripts that automate trunk group configuration. If you see the message “Trunk group *name* could not be added, as the threshold is reached”, run **debug tgrm** and check the number of trunk groups and the memory usage.

	Command	Purpose
Step 1	Router(config)# trunk group <i>name</i>	Enters trunk-group configuration mode and initiates the definition of a trunk group identified by <i>name</i> .
Step 2	Router(config-trunk-group)# carrier-id <i>name</i>	Specifies the carrier that owns the trunk group.
Step 3	Router(config-trunk-group)# max-calls { any data voice } <i>number</i> [direction in out]	Indicates the maximum number of calls that the trunk group can handle.

	Command	Purpose
Step 4	<p>Least -idle Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme least-idle [even odd both]</pre> <p>Least-used Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme least-used [even odd both [up down]]</pre> <p>Longest-idle Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme longest-idle [even odd both]</pre> <p>Random Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme random</pre> <p>Round-robin Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme round-robin [even odd both [up down]]</pre> <p>Sequential Hunt Scheme</p> <pre>Router(config-trunk-group)# hunt-scheme sequential [even odd both [up down]]</pre>	Specifies the search method for finding an available voice channel in the trunk group.
Step 5	<pre>Router(cfg-source-grp)# description text</pre>	(Optional) Describes the trunk group.
Step 6	<pre>Router(config-trunk-group)# translation-profile incoming name</pre>	Associates a translation profile with incoming calls.
Step 7	<pre>Router(config-trunk-group)# translation-profile outgoing name</pre>	Associates a translation profile with outgoing calls.
Step 8	<pre>Router(config-trunk-group)# exit</pre>	Ends the trunk group profile definition.

The following example illustrates a trunk group configuration.

```
Router(config)# trunk group 100
Router(config-trunk-group)# description This trunk is for the MAIN carrier
Router(config-trunk-group)# carrier-id main
Router(config-trunk-group)# max-calls any 100 direction in
Router(config-trunk-group)# max-retry 3
Router(config-trunk-group)# hunt-scheme sequential
Router(config-trunk-group)# translation-profile incoming sanjose
Router(config-trunk-group)# translation-profile outgoing newyork
Router(config-trunk-group)# exit
```

Verifying the Trunk Group Configuration

See the command descriptions later in this document for sample outputs and their explanation.

- Enter **show run** to display the gateway configuration. Look for these lines in the output display:

```
Router# show run
!
!
trunk group 100
description This trunk is for the MAIN carrier
carrier-id main
```

```

max-calls any 100 direction in
max-retry 3
hunt-scheme sequential
translation-profile incoming sanjose
translation-profile outgoing newyork
!

```

- Enter **show trunk group name** to display the configuration for a specific trunk group.



Note Do not use **show trunk-group**.

Using trunk group 100 configured above, this display appears:

```

Router# show trunk group 100

Trunk group: 100
  Description: This trunk is for the MAIN carrier
  Carrier ID:main

  Translation profile (Incoming):sanjose
  Translation profile (Outgoing):newyork

  Hunt Scheme is sequential
  Max Calls (Incoming): 100 (Any) NOT-SET (Voice) NOT-SET (Data)
  Max Calls (Outgoing): NOT-SET (Any) NOT-SET (Voice) NOT-SET (Data)
  Retries:3
  Trunk          Preference DEFAULT
    Total channels available : 23
    Data = 0, Voice = 0, Pending = 0, Free = 23

  Total calls for trunk group: Data = 0, Voice = 0, Pend = 0, Free = 23

```

- Both **show trunk group** and **show crm** display values for the maximum number of calls. These values originate from different configuration procedures.
 - In **show trunk group**, the Max Calls value originates from the **max-calls** command in the trunk group configuration. For example:

```

Max Calls (Incoming): 100 (Any) NOT-SET (Voice) NOT-SET (Data)
Max Calls (Outgoing): NOT-SET (Any) NOT-SET (Voice) NOT-SET (Data)
Retries:3

```

- In **show crm**, Max calls indicates the maximum number of available channels after the carrier ID or trunk group label is assigned to an interface using the **trunk-group (interface)** command. For example:

```

Router# show crm
!
!
Trunk Group Label: 100
  Max calls:400
  Max Voice (in) : 400   Cur Voice (in) :      0
  Max Voice (out): 400   Cur Voice (out):    0
  Max Data (in)  : 400   Cur Data (in)  :    0
  Max Data (out) : 400   Cur Data (out) :    0
!
!

```

See the “[Configuring an NFAS Interface](#)” section on page 41 for the procedure to assign an interface to a trunk group.

Troubleshooting Tips for the Trunk Group Configuration

Do not assign a trunk group label with the same name as an existing carrier ID. You will see the following message while configuring the trunk group.

```
Router(config)# trunk group main

trunk group label conflicts with carrier ID
```

To correct this problem, use a different name for the trunk group label.

Configuring a Source VoIP Group

Follow these steps to configure a source VoIP group, beginning in global configuration mode:



Note

Configure the gateway and the network for either carrier ID or trunk group routing. Using them together is not supported and will have unpredictable behavior.

	Command	Purpose
Step 1	Router(config)# voice source-group <i>name</i>	Enters voice-source-group configuration mode and begins the definition of the VoIP source group.
Step 2	Router(cfg-source-grp)# access-list <i>number</i>	Identifies the access list to use to find the source of an incoming VoIP call.
Step 3	Carrier ID Routing Router(cfg-source-grp)# carrier-id <i>source name</i>	Specifies the carrier as the source of an incoming VoIP call.
	Trunk Group Label Routing Router(cfg-source-grp)# trunk-group-label <i>source name</i>	Specifies the trunk group as the source of an incoming VoIP call.
Step 4	Carrier ID Routing Router(cfg-source-grp)# carrier-id <i>target name</i>	Specifies the carrier as the target of the outgoing POTS call.
	Trunk Group Label Routing Router(cfg-source-grp)# trunk-group-label <i>target name</i>	Specifies the trunk group as the target of the outgoing POTS call.
Step 5	Router(cfg-source-grp)# description <i>text</i>	(Optional) Describes the VoIP source group.
Step 6	Router(cfg-source-grp)# disconnect-cause <i>reason</i>	Specifies a reason for blocking calls.
Step 7	Router(cfg-source-grp)# translation-profile <i>incoming name</i>	Associates a translation profile with incoming calls.
Step 8	Router(cfg-source-grp)# h323zone-id <i>name</i>	(For H.323 calls only) Specifies the zone for an incoming H.323 call.
Step 9	Router(cfg-source-grp)# exit	Ends the voice source group profile definition.

Verifying the Source VoIP Group Configuration

See the command descriptions later in this document for sample outputs and their explanation.

- Enter **show voice source-group** to display the details of all source IP groups or **show voice source-group name** to display the details of a specific source IP group. For example:

```
Router# show voice source-group
```

```
Router# show voice source-group abc
```

- To test the functionality of a source IP group, enter one of the following commands:
 - **test source-group carrier-id name**
 - **test source-group h323zone-id name**
 - **test source-group ip-address ip-address**
 - **test source-group trunk-group-label source name**

Troubleshooting Tips for the Source IP Group Configuration

- Do not use the same name for more than one source IP group. If this happens, you will see a message such as:

```
Router(config)# voice source-group 130
Router(cfg-source-grp)# carrier-id source MAIN
```

```
Error: Input source carrier id is defined as source carrier id in another source
group.
```

- Do not mix carrier ID routing with trunk group label routing. The following example illustrates an invalid configuration.

```
Router# show voice source-group abc
```

```
Source Group: abc
  description="",
  carrier-id source="cisco",
  carrier-id target="",
  trunk-group-label source="",
  trunk-group-label target="first",
  h323zone-id="",
  access-list=,
  disconnect-cause="no-service",
  translation-profile="",
Router#
```

The following example illustrates a valid configuration.

```
Router# show voice source-group abc
```

```
Source Group: abc
  description="",
  carrier-id source="cisco",
  carrier-id target="mainoffice",
  trunk-group-label source="",
  trunk-group-label target="",
  h323zone-id="",
  access-list=,
  disconnect-cause="no-service",
```

```
translation-profile="",
Router#
```

If you are using trunk-group-label routing, both trunk-group-label fields would be filled in and the carrier-id fields would be blank.

Configuring a Voice Port

Use a voice port to configure an analog interface to a trunk group. Follow these steps to configure a translation profile for a voice port, beginning in global configuration mode. Include a profile for incoming, outgoing, or both:

	Command	Purpose
Step 1	Router(config)# voice-port <i>number</i>	Enters voice-port configuration mode and specifies the voice port to be defined.
Step 2	Router(config-voiceport)# translation-profile incoming <i>name</i>	Associates a number translation profile with incoming calls.
Step 3	Router(config-voiceport)# translation-profile outgoing <i>name</i>	Associates a translation profile with outgoing calls.
Step 4	Router(config-voiceport)# trunk-group <i>name</i> [<i>preference</i>]	Assigns the voice port to a trunk group.
Step 5	Router(config-)# exit	Ends the voice port definition.

The following example illustrates a trunk group configuration for an analog interface.

```
Router(config)# voice-port 3/0/0
Router(config-voiceport)# trunk-group 100
Router(config-voiceport)# translation-profile incoming marketing
Router(config-voiceport)# translation-profile outgoing sales
Router(config-voiceport)# exit
```

Verifying the Voice Port Configuration

Enter **show voice port** *number* to display the configuration for a specific voice port or **show voice port summary** to display the configurations for all voice ports on the gateway. For example:

```
Router# show voice port
```

```
Router# show voice port summary
```

Assigning Translation Profiles to Inbound Dial Peers

Follow these steps to assign a translation profile to incoming VoIP, VoATM, VoFR, and POTS dial peers, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# dial-peer voice num [pots voip vofr voatm]	Enters dial-peer configuration mode that creates or modifies a dial peer.
Step 2	Router(config-dial-peer)# translation-profile incoming name	Associates a number translation profile with incoming calls.
Step 3	Router(config-dial-peer)# exit	Exits the dial peer configuration mode.

Configuring Call Blocking on Inbound Dial Peers

Follow these steps to configure call blocking for inbound POTS and VoIP dial peers, beginning in global configuration mode.



Note

Configure the gateway and the network for either carrier ID or trunk group routing. Using them together is not supported and will have unpredictable behavior.

	Command	Purpose
Step 1	Router(config)# dial-peer voice num [pots voip voatm vofr]	Enters dial-peer configuration mode that creates or modifies a dial peer.
Step 2	Router(config-dial-peer)# call-block translation-profile incoming name	Associates a call blocking translation profile for incoming calls. This command is available only for VoIP and POTS dial peers.
Step 3	Router(config-dial-peer)# call-block disconnect-cause incoming cause	Specifies a disconnect cause for blocking incoming calls.
Step 4	Carrier ID Routing Router(config-dial-peer)# carrier-id source name	Specifies the carrier as the source of an incoming call.
	Trunk Group Label Routing Router(config-dial-peer)# trunk-group-label source name	Specifies the trunk group as the source of an incoming call.
Step 5	Router(config-dial-peer)# exit	Exits dial peer configuration mode.

Configuring an Outbound POTS Dial Peer

Follow these steps to assign a translation profile to outbound POTS dial peers, beginning in global configuration mode:



Note

Configure the gateway and the network for either carrier ID or trunk group routing. Using them together is not supported and will have unpredictable behavior.

	Command	Purpose
Step 1	<code>Router(config)# dial-peer voice <i>number</i> pots</code>	Enters dial-peer configuration mode that creates or modifies a dial peer.
Step 2	<p>Carrier ID Routing</p> <code>Router(config-dial-peer)# carrier-id target <i>name</i></code>	Assigns the carrier as the target of the outgoing call.
	<p>Trunk Group Label Routing</p> <code>Router(config-dial-peer)# trunk-group-label target <i>name</i></code>	Assigns the trunk group as the target of the outgoing call.
Step 3	<code>Router(config-dial-peer)# trunkgroup <i>name</i> [<i>preference</i>]</code>	Assigns the trunk group to a dial peer. Repeat this command as needed to assign the dial peer to multiple trunk groups.
Step 4	<code>Router(config-dial-peer)# translation-profile outgoing <i>name</i></code>	Assigns a translation profile for outgoing calls.
Step 5	<code>Router(config-dial-peer)# exit</code>	Exits the dial peer configuration mode.

The following example illustrates a trunk group configuration for a POTS dial peer.

```
Router(config)# dial-peer voice 201 pots
Router(config-dial-peer)# trunkgroup TEXAS 1
Router(config-dial-peer)# trunkgroup NEWYORK 2
Router(config-dial-peer)# carrier-id source WEST
Router(config-dial-peer)# carrier-id target EAST
Router(config-dial-peer)# translation-profile incoming ca101
Router(config-dial-peer)# translation-profile outgoing ca105
Router(config-dial-peer)# exit
```

Following are a few considerations about this example.

- This POTS dial peer is assigned to two trunk groups. TEXAS has a higher preference than NEWYORK, so the software will attempt to route calls coming to this dial peer using the TEXAS trunk group. If the TEXAS trunk group is not able to handle the call, the software will use the NEWYORK trunk group.
- The carrier ID value is case-sensitive. “WEST” and “west” are different carrier IDs.
- Include **carrier-id source** if calls originate using this POTS dial peer.
- Include **carrier-id target** if calls terminate using this POTS dial peer.
- The carrier ID is not validated during the configuration.
- Do not use both **carrier-id** and **trunk-group-label** in the same dial peer configuration.

Verifying the Dial Peer Configuration

- Enter **show dial-peer voice *number*** to display the configuration for a specific dial peer or **show dial-peer voice summary** to display the configurations for all dial peers on the gateway. For example:

```
Router# show dial-peer voice 110
```

See the command descriptions later in this document for sample outputs and their explanation.

- Enter **show dialplan in-carrier *carriername* pots** to display the dial peer configuration for a specific carrier ID or trunk-group-label. For example:

```

Router# show dialplan in-carrier WEST pots

Inbound pots dialpeer Matching based on source carrier-id
VoiceEncapPeer201
  information type=voice,
  description="",
  tag = 201, destination-pattern="",
  answer-address="", preference=0,
  source carrier-id="WEST", target carrier-id="EAST",
  source trunk-group-label="", target trunk-group-label="",
  numbering type="unknown",
  group=201, Admin state is up, Operation state is up,
  incoming called-number="", connections/maximum=0/unlimited,
  DTMF Relay=disabled,
  Translation profile (incoming):ca101
  Translation profile (outgoing):ca105
  incoming call blocking:
  translation-profile="",
  disconnect-cause="invalid-number"

```

Resource Monitor and Dial Peers

If you use the resource monitor to display digital signal processor (DSP) and DS-0 resources, the trunk group resources are included in the display only if the dial peer is operationally up. Individual voice ports assigned to the dial peer are displayed regardless of the status of the dial peer. Although a trunk group is a set of voice ports, those resources assigned to the trunk group are displayed only when the dial peer is in an up state.

For example, suppose a gateway has this configuration:

```

controller T1 7/3
  framing esf
  linecode ami
  ds0-group 1 timeslots 1-4 type e&m-fgb dtmf dnis
  cas-custom 1
  trunk-group gw14-tg1

dial-peer voice 1 pots
  trunkgroup gw14-tg1
  carrier-id target abcd

```

The trunk group is associated with four timeslots in the DS-0 group and routing is done through a POTS dial peer. If you enter **show call resource voice stats**, the output when the dial peer is up looks like:

```

Resource Monitor - Dial-up Resource Statistics Information

Utilization: 0 percent
total channels: 12
Inuse channels: 0
Disabled channels: 0
Pending channels: 0
Free channels: 12

DS0 Statistics:
Utilization: 0 percent
Total channels: 96
Addressable channels: 4
Inuse channels: 0
Disabled channels: 0
Free channels: 4

```

If the dial peer is down, the Addressable channels and Free channels fields would be 0.

To check the status of the dial peer, enter **show dial-peer voice *number* | inc Operation state**.

Configuring an Outbound VoIP Dial Peer

Follow these steps to configure an outbound VoIP dial peer, beginning in global configuration mode:

	Command	Purpose
Step 1	<code>Router(config)# dial-peer voice number voip</code>	Enters dial peer configuration mode and initiates the dial peer definition.
Step 2	<code>Router(config-dial-peer)# session protocol sipv2</code>	Configures the dial peer to use IETF SIP. SIP users should use this option.
Step 3	<code>Router(config-dial-peer)# destination-pattern [+]<i>string</i> [T]</code>	Defines the telephone number associated with this VoIP dial peer. <ul style="list-style-type: none"> +—(Optional) Character indicating an E.164 standard number. <i>string</i>—Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 to 9, the letters A to D, and special character
Step 4	<code>Router(config-dial-peer)# session target enum: <i>table-num</i></code>	Specifies an ENUM search table for the target session.
Step 5	<code>Router(config-dial-peer)# translation-profile outgoing <i>name</i></code>	Specifies a translation profile for outgoing calls.
Step 6	<code>Router(config-dial-peer)# exit</code>	Ends the dial peer definition.

The following example illustrates an outbound VoIP dial peer.

```
Router(config)# dial-peer voice 2201 voip
Router(config-dial-peer)# destination-pattern 2..
Router(config-dial-peer)# session target enum:2
Router(config-dial-peer)# translation-profile outgoing ca103
Router(config-dial-peer)# exit
```

Assigning a Translation Profile to an NFAS Interface

Follow these steps to assign a translation profile to an NFAS interface, beginning in global configuration mode:

	Command	Purpose
Step 1	<code>Router(config)# voice service pots</code>	Enters voice-service configuration mode and initiates the voice service POTS definition.
Step 2	<code>Router(conf-voi-serv)# translation-profile [<i>incoming</i> <i>outgoing</i>] controller [T1 E1] <i>unit-number name</i></code>	Assigns the translation profile <i>name</i> to the NFAS interface.
Step 3	<code>Router(conf-voi-serv)# exit</code>	Ends the voice service POTS definition.

Configuring an NFAS Interface

Follow these steps to configure an NFAS interface as a trunk group member, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface serial slot_number	Enters interface configuration mode.
Step 2	Router(config-if)# trunk-group name [<i>preference_num</i>]	Assigns the trunk group to the NFAS interface.
Step 1	Router(config-if)# exit	Ends interface configuration mode.

The following example illustrates assigning an ISDN interface to a trunk group.

```
Router(config)# interface Serial7/1:23
Router(config-if)# trunk-group 100 1
Router(config-if)# exit
```

More than one ISDN interface can be assigned to the same trunk group. For example:

```
Router(config)# interface Serial7/1:23
Router(config-if)# trunk-group 100 1
Router(config-if)# exit
```

```
Router(config)# interface Serial7/0:23
Router(config-if)# trunk-group 100 1
Router(config-if)# exit
```

Verifying the NFAS Interface Configuration

- Enter **show run** to display the gateway configuration. Look for these lines in the output display.

```
Router# show run

!
!
interface Serial7/1:23
  no ip address
  trunk-group 100 1
  isdn switchtype primary-4ess
  isdn incoming-voice modem
!
```

- Enter **show trunk group name** to display the configuration for a specific trunk group. For example:

```
Router# show trunk group 100

Trunk group: 100
  Description: This trunk is for the MAIN carrier
  Carrier ID: main
  Translation profile (Incoming):
  Translation profile (Outgoing):
  Hunt Scheme is sequential
  Max Calls (Incoming): 100 (Any) NOT-SET (Voice) NOT-SET (Data)
  Max Calls (Outgoing): NOT-SET (Any) NOT-SET (Voice) NOT-SET (Data)
  Retries: 3

Trunk 7/1:D Preference 1
  Total channels available: 23
```

```

    Data = 0, Voice = 0, Pending = 0, Free = 23
Trunk 7/0:D Preference 1
    Total channels available: 23
    Data = 0, Voice = 0, Pending = 0, Free = 23

Total calls for trunk group: Data = 0, Voice = 0, Pend = 0, Free = 46

```

Notice that the trunk group has two ISDN interfaces assigned to it and both interfaces have the same preference number.

- If **show trunk group name** displays 0 for Free calls, then the serial interface is not up yet. For example:

```

Router# show trunk group 100

!
!
!
Trunk 7/1:D Preference 1
    Total channels available: 23
    Data = 0, Voice = 0, Pending = 0, Free = 23
Trunk 7/0:D Preference 1
    Total channels available: 23
    Data = 0, Voice = 0, Pending = 0, Free = 0
!

```

Verify the status of the serial interface by entering **show ISDN stat**. If the controller is down, enter **shut** and **no shut** on the interface to try to start the interface.

Troubleshooting Tips for the NFAS Interface Configuration

- Do not assign an ISDN interface to a trunk group that contains a CAS trunk. You will see the following message.

```

Router(config)# interface Serial7/1:23
Router(config-if)# trunk-group 200

```

Attempt to add ISDN interface into a TG that contains CAS trunk

Assign a different trunk group to this interface.

- Use the correct command when assigning the DS-0 group to the trunk group. The following example will generate an error message.

```

Router(config)# interface Serial7/1:23
Router(config-if)# trunk group NEWYORK
Router(config-controller)# exit

```

This will not add trunk to ISDN interface

The **trunk group** command creates a new trunk group named NEWYORK, but it does not assign NEWYORK to the ISDN interface. Use **trunk-group NEWYORK** instead.

Configuring a DS-0 Group

Follow these steps to configure DS-0 group as a trunk group member, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# controller {t1 e1} slot/port	Enters controller configuration mode.
Step 2	Router(config-if)# trunk-group name [preference_num]	Assigns the trunk group to the DS-0 group.
Step 3	Router(config-if)# exit	Ends controller configuration mode.

The following example illustrates assigning a DS-0 group to a trunk group.

```
Router(config)# controller T1 7/7
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# ds0-group 1 timeslots 1-23 type e&m-fgb dtmf dnis
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk-group ma200
Router(config-controller)# exit
```

Multiple DS-0 groups can be assigned to trunk groups. The following example assigns two different DS-0 groups to different trunk groups.

```
Router(config)# controller T1 7/7
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# ds0-group 1 timeslots 1-10 type e&m-fgb dtmf dnis
Router(config-controller)# ds0-group 2 timeslots 11-23 type e&m-fgd dtmf dnis
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk-group ma200
Router(config-controller)# cas-custom 2
Router(config-controller)# trunk-group ny150
Router(config-controller)# exit
```

The following example illustrates two different DS-0 groups assigned to the same trunk group.

```
Router(config)# controller T1 7/7
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# ds0-group 1 timeslots 1-10 type e&m-fgb dtmf dnis
Router(config-controller)# ds0-group 2 timeslots 11-23 type e&m-fgd dtmf dnis
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk-group ny150
Router(config-controller)# cas-custom 2
Router(config-controller)# trunk-group ny150
Router(config-controller)# exit
```

The following example illustrates two different controllers assigned to the same trunk group.

```
Router(config)# controller T1 0/1
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# ds0-group 1 timeslots 1-23 type e&m-fgb dtmf dnis
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk-group ny150
Router(config-controller)# exit

Router(config)# controller T1 7/7
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# ds0-group 1 timeslots 1-23 type e&m-fgd dtmf dnis
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk-group ny150
Router(config-controller)# exit
```



Note Do not assign a DS-0 group to two different trunk groups.

Verifying the DS-0 Group Configuration

Enter `show run` to display the gateway configuration. Look for these lines in the output display.

```
Router# show run

!
!
controller T1 7/7
  framing esf
  linecode b8zs
  ds0-group 1 timeslots 1-23 type e&m-fgb dtmf dnis
  cas-custom 1
  trunk-group ma200
!
```

Troubleshooting Tips for the DS-0 Group Configuration

Use the correct command when assigning the DS-0 group to the trunk group. The following example will generate an error message.

```
Router(config)# controller T1 7/6
Router(config-controller)# cas-custom 1
Router(config-controller)# trunk group TEXAS
Router(config-controller)# exit
```

This will not add trunk to CAS custom 1

The **trunk group** command creates a new trunk group named TEXAS, but it does not assign TEXAS to the CAS interface. Use **trunk-group TEXAS** instead.

Configuring a Global Translation Profile for Incoming VoIP Calls

Use this command to assign a translation profile to all incoming VoIP calls, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# voip-incoming translation-profile <i>name</i>	Assigns the translation profile to all incoming VoIP calls.

Configuring Call Capacity Reporting for Unsolicited Calls

Follow these steps to configure call capacity reporting of unsolicited calls, beginning in global configuration mode:

	Command	Purpose
Step 1	Carrier ID Routing Router(config)# <code>voice call carrier capacity active</code>	Enables the periodic call capacity reporting of H.323, SIP, and TDM calls for carrier ID routing.
	Trunk Group Label Routing Router(config)# <code>voice call trunkgroup capacity active</code>	Enables periodic call capacity reporting for H.323, SIP, and TDM calls for trunk group label routing.
Step 2	Carrier ID Routing Router(config)# <code>voice call carrier capacity timer interval [timer-interval]</code>	Sets the timing interval for periodic reporting of H.323, SIP, and TDM calls for carrier ID routing.
	Trunk Group Label Routing Router(config)# <code>voice call trunkgroup capacity timer interval [timer-interval]</code>	Sets the timing interval for periodic reporting of H.323, SIP, and TDM calls for trunk group label routing.

Monitoring and Maintaining Gateway Trunk and Carrier Based Routing

To monitor and maintain the gateway trunk and carrier based routing enhancements, use the following commands in privileged EXEC mode:

Command	Purpose
Router# <code>debug crm</code>	Displays the Carrier Resource Manager information.
Router# <code>debug csm tgrm</code>	Displays the Call Switching Module trunk group resource manager information.
Router# <code>debug dialpeer</code>	Displays the dial peer information.
Router# <code>debug isdn tgrm</code>	Displays the ISDN trunk group resource manager information.
Router# <code>debug tgrm</code>	Displays trunk group resource manager information.
Router# <code>debug voice enum [detail summary]</code>	Displays the ENUM match table information.
Router# <code>debug voice source-group</code>	Displays the source IP group information.
Router# <code>debug voice translation</code>	Displays the translation-rule information.
Router# <code>debug voip enum [detail summary]</code>	Displays the ENUM match table information.
Router# <code>show crm</code>	Displays carrier's call capacities information.
Router# <code>show dialplan number dialstring</code>	Displays the dial peer that is reached when a specific telephone number is dialed.
Router# <code>show trunk group</code>	Displays the channel status of individual trunk group members.

Troubleshooting VoIP Gateway Trunk and Carrier Based Routing

This section provides troubleshooting guidelines for the VoIP Gateway Trunk and Carrier Based Routing feature.

Frequently asked questions regarding trunk and carrier configurations include:

- Why are my trunk or carrier based calls not going through?
- Why are incoming calls not blocked as expected?
- Why are the number translations not working properly?
- Why is ENUM not working properly?

Why are my trunk or carrier based calls not going through?

Several causes may contribute to calls not going through. Check the following configuration components:

- [Troubleshooting Trunk Group Configurations](#)
- [Troubleshooting Dial Peer Configurations](#)
- [Troubleshooting Source IP Group Configurations](#)
- The problem may be with incoming call blocking. See the [“Why are incoming calls not blocked as expected?” section on page 49](#) for troubleshooting guidelines.
- The problem may be with the translation rules. See the [“Why are the number translations not working properly?” section on page 50](#) for troubleshooting guidelines.

Troubleshooting Trunk Group Configurations

Verify the following configuration parameters:

- Ensure that the maximum number of calls is not set to zero (0).
Use **show trunk group** to display the value for Max Calls. If the value is 0, use the **max-calls** command in the trunk group configuration mode to reset the value. See the [“Configuring a Trunk Group” section on page 31](#) for the trunk group configuration procedure.
- Ensure that the trunk group is assigned to the correct interface.
Use **show trunk group** to display the interface associated with the trunk group. If the interface is incorrect, use one of the configuration procedures described earlier in this section to associate the trunk group to the correct interface.
- Verify that a channel is available for the call.
Use **show trunk group** to display the number of available channels. If no channels are available, then do one of the following:
 - Wait for one of the channels to free up.
 - If this is a constant problem,
- Verify that the trunk group is configured under an outbound POTS dial peer.
- Verify that the carrier ID is configured under an outbound POTS dial peer.

- Use **debug tgrm** to trace the output messages from the trunk group at the terminating gateway. Look for these lines in the output to verify the carrier ID and trunk group used by the calls.

```
TGRM: tgrm_tg_get_carrier_id: carrier_id=711
! Successfully reserves the intended trunk group and timeslot.
TGRM:tgrm_select_by_state() - Success! Reserved tgTag=711,tgmPref#=65,Ts=1
```

Troubleshooting Dial Peer Configurations

Verify the following dial peer parameters:

- Verify that the source carrier ID is configured under an inbound dial peer.

Use **show run** to display the gateway configuration. Look for source carrier ID lines in the dial peer configurations. Look for lines that include carrier-id source, such as

```
dial-peer voice 1311 pots
  carrier-id source s-alpha
  call-block translation-profile incoming cb_xrule
  translation-profile incoming xrule
```

If you do not see entries such as these, refer to [Configuring Call Blocking on Inbound Dial Peers](#) earlier in this section for the procedure to assign a source carrier ID to an inbound dial peer.

- Verify that the target carrier ID is configured under an outbound POTS dial peer.

Use **show run** to display the gateway configuration. Look for target carrier ID lines in the dial peer configurations. Look for an entry that includes carrier-id target, such as

```
dial-peer voice 205 pots
  trunkgroup BOSTON
  carrier-id target main
  translation-profile outgoing tx17
  forward digits all
```

If you do not see entries such as these, refer to [Configuring an Outbound POTS Dial Peer](#) earlier in this section for the procedure to assign a target carrier ID to an outbound dial peer.

- Use **debug dialpeer** to find the matching dial peer.

On the originating gateway, look for indications that the dial peer has been matched. For example,

```
Router# debug dialpeer
!
!
00:22:28: dpAssociateIncomingPeer_T:Matching route label abc1
00:22:28: Inside dpMatchCore:
!
!
00:22:28: destination pattn:5108880101 expanded string:5108880101
00:22:28:MatchNextPeer:Peer 1311 matched
```

On the terminating gateway, look for indications that the target carrier id (CID) has been matched. For example,

```
Router# debug dialpeer
!
!
00:22:28: dpMatchPeersMoreArg: Match Dest. pattern + Target CID; called (1025168853)
target CID (abc1)
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:1025168853 expanded string:1025168853
!
!
```

```
00:22:28:MatchNextPeer:Peer 201 matched
```

- Use **show dial-peer voice summary** to display the status of the dial peer. Verify that the status of the trunk groups is up.

Troubleshooting Source IP Group Configurations

Verify the following source IP group parameters:

- Verify that the target carrier ID or target trunk-group-label is configured. Use **show voice source-group**. If a target is not configured for the source IP group, see [Configuring a Source VoIP Group](#) earlier in this section for the procedure to configure a target carrier ID or target trunk-group-label.

- Verify that the gateway can find the carrier ID or trunk-group-label configured for the source IP group. Use **test source-group**. For example:

```
Router# test source-group carrier-id source abc

A source-group is found with source carrier-id abc
```

```
Source Group: 130
  description="",
  carrier-id source="abc"
  carrier-id target="xyz"
  trunk-group-label source="",
  trunk-group-label target="",
  h323zone-is="",
  access-list=,
  disconnect-cause="no-service"
  translation-profile="abc-profile-sip",
Router#
```

If the carrier ID or trunk-group-label is not found, see [Configuring a Source VoIP Group](#) for the procedure to configure the carrier ID or trunk-group-label.

- If the source IP group is an entry in an access list, verify that the access list is not blocking the call. For example, suppose a router has the following configuration:

```
!
voice source-group 101
  access-list 1
!
!
access-list 1 permit 10.8.50.13
access-list 1 deny 10.8.56.99
!
```

Test the permit and deny IP addresses:

```
Router# test source-group ip-address 10.8.50.13

A source group is found with ip address=10.8.50.13
```

```
Source Group:101
  description="",
  carrier-id source="",
  carrier-id target="",
  trunk-group-label source="",
  trunk-group-label target="",
  h323zone-id="",
  access-list=1,
```

```

disconnect-cause="no-service",
destination-pattern="",
incoming called-number="",
translation-profile="",

Router# test source-group ip-address 10.8.56.99

A source group is found with ip address=10.8.56.99
An ip address 10.8.56.99 is rejected with disc-cause="no-service"

Source Group:101
description="",
carrier-id source="",
carrier-id target="",
trunk-group-label source="",
trunk-group-label target="",
h323zone-id="",
access-list=1,
disconnect-cause="no-service",
destination-pattern="",
incoming called-number="",
translation-profile=""

```

- Verify that the source IP group is not configured with both a carrier ID and a trunk-group-label. Use [show voice source-group](#) and verify that the source and target entries are both carrier IDs or both trunk-group-labels. If the display shows one carrier ID and one trunk-group-label, see [Configuring a Source VoIP Group](#) earlier in this section for the procedure to configure a source IP group.

Why are incoming calls not blocked as expected?

Call blocking requires specific configuration procedures.

- Verify that call blocking is configured for the called number, calling number, or redirect number. Use **show run** to check if call blocking is configured on an incoming dial peer or source IP group.

If call blocking is not configured, use translation rules for an incoming dial peer or use an access list for a source IP group.

- To use translation rules, configure a translation rule that rejects the incoming number. For example, to block a call with calling number 912230XXXX, create a translation rule such as

```

Router(config)# voice translation-rule 1
Router(cfg-translation-rule)# rule 1 reject /912230*/

```

Assign the translation-rule to a profile, such as

```

Router(config)# voice translation profile blocking
Router(cfg-translation-profile)# translate calling 1

```

Assign the call blocking profile to an inbound dial peer, such as

```

Router(config)# dial-peer voice 201 pots
Router(config-dial-peer)# call-block translation-profile incoming blocking
Router(config-dial-peer)# call-block disconnect-cause incoming unassigned-number

```

Use [debug voice translation](#) to verify that the call blocking is working. Look in the output display for a call block rule line, such as

```

Router# debug voice translation
!
00:51:56:regxrule_stack_pop_RegXruleNumInfo:stack=0x63DECAF4; count=1

```

```
00:51:56:regxrule_profile_match:Matched a call block rule; number 912230456 rule
precedence=1
00:51:56:regxrule_profile_block:Matched with rule 1 in ruleset 1
00:51:56:regxrule_stack_pop_RegXruleNumInfo:stack=0x63DECAF4; count=0
```

- To use an access list, assign a standard access list to a source IP group that has no source carrier ID configured. For example, to block calls coming from IP address 11.0.0.5, the access list would look like

```
Router(config)# access-list 15 deny host 11.0.0.5
```

To add the access list to the source IP group, the commands would be

```
Router(config)# voice source-group 115
Router(cfg-source-grp)# access-list 15
```

Use **show voice source-group** to ensure that the access list is configured and that the source IP group does not have a source carrier ID or source trunk-group-label.

- Verify that the translation rule is configured. Use **show voice translation-rule** to verify the translation rule parameters. If the translation rule does not exist, see [Configuring a Translation Rule](#) earlier in this section for the procedure to configure a translation rule.
- Verify that call blocking is configured under an incoming dial peer. Use **show run** to see if an incoming dial peer has call blocking configured. If call blocking is not configured, see [Configuring Call Blocking on Inbound Dial Peers](#) earlier in this section for the procedure.

Why are the number translations not working properly?

The following factors can account for number translations not working properly.

- Translations can be defined for called, calling, or redirect numbers. Verify that a translation rule exists for the numbers you want translated. Use **show voice translation-rule** to display the patterns defined for the translation rules.
- Verify that the translation rules are configured correctly. Check the syntax of the rule. See the [“Number Translation”](#) section on page 12 for guidelines on translation rule syntax. See the [“Configuring a Translation Rule”](#) section on page 28 for the procedure to define and verify a translation rule.
- Check the translation rules in a given translation profile to ensure that none of the rules interfere with each other. A new translation rule added to a profile may affect the functioning of an earlier translation rule. Use **show voice translation-profile** to display the rules of one or more translation profiles.
- Verify that the translation rule is assigned to the translation profile being used for the interface. Use **show voice translation-profile** to display the rules of a translation profile.
- Verify that the interface is configured with the correct translation profile. Use a **show** command for the interface to display the translation profiles assigned to it. Use **show voice translation-profile** to display the rules in the translation profiles.
- For incoming H.323 calls, the translation profile of a source IP group overrides the global translation profile for incoming VoIP calls. A call you expect to be translated by the global VoIP translation profile may be affected by a source IP group translation profile. Verify that the source IP group profiles do not conflict with the global VoIP profile.
- For outgoing DS-0 calls, verify that the translation profile assigned to the “trunk-group” configured for the DS-0 interface is also assigned to the “trunkgroup” configured for an outbound POTS dial peer.

Why is ENUM not working?

Check the following ENUM configuration components.

- Verify that the session target is configured in an outbound VoIP dial peer. Use **show run** to display the configurations of the dial peers. If an outbound dial peer does not have session target configured, see [Configuring an Outbound VoIP Dial Peer](#) earlier in this section for the procedure to configure the session target.
- Verify that the default matching pattern exists in the ENUM table. Use **show voice enum-match-table** to display the configuration of the ENUM table. Use **test enum** and **debug voice enum** to verify that the matching pattern works properly.
- Verify that DNS is reachable.
- Verify that DNS is configured for ENUM.
- Verify that the destination URL is reachable.

Configuration Examples

This section provides the following configuration examples:

- [Configuring Trunk and Carrier Based Routing on an Originating Gateway: Example](#)
- [Configuring Trunk and Carrier Based Routing on a Terminating Gateway: Example](#)

Configuring Trunk and Carrier Based Routing on an Originating Gateway: Example

The following output shows how to configure trunk and carrier based routing on the originating gateway.

```
!  
version 12.2  
no service single-slot-reload-enable  
no service pad  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
service internal  
!  
hostname first  
!  
!  
resource-pool disable  
!  
!  
voice-fastpath enable  
ip subnet-zero  
no ip domain-lookup  
ip host dirt 192.168.254.254  
ip name-server 172.20.140.108  
!  
no ip dhcp-client network-discovery  
lcp max-session-starts 0  
isdn switch-type primary-5ess  
!  
voice enum-match-table 1
```

```

rule 1 5 /^9(1.*\)/ /+\1/ e164.cisco.com
rule 10 1 /^(.*)/ /\1/ e164.cisco.com
!
voice translation-rule 1
rule 1 /408/ /804/
rule 2 /804/ /732/
rule 3 /875/ /785/
!
voice translation-rule 2
rule 1 /785/ /786/
rule 2 /786/ /875/
rule 3 /732/ /408/
!
voice translation-rule 3
rule 1 /303/ /330/
rule 2 /330/ /331/
!
voice translation-rule 4
rule 1 reject /4087775555/
rule 2 reject /6503232222/
!
voice translation-profile cb_xrule
translate called 4
translate calling 4
translate redirect-called 4
!
voice translation-profile xrule
translate calling 1
translate called 2
translate redirect-called 3
!
fax interface-type modem
mta receive maximum-recipients 0
!
controller T1 7/0
framing sf
linecode ami
!
controller T1 7/1
framing esf
linecode ami
ds0-group 1 timeslots 1-4 type e&m-fgb dtmf dnis
ds0-group 2 timeslots 5-8 type e&m-fgb dtmf dnis
ds0-group 3 timeslots 9-12 type e&m-fgb dtmf dnis
ds0-group 4 timeslots 13-16 type e&m-fgb dtmf dnis
ds0-group 5 timeslots 17-20 type e&m-fgb dtmf dnis
ds0-group 6 timeslots 21-24 type e&m-fgb dtmf dnis
cas-custom 1
trunk-group 11
cas-custom 2
trunk-group 11
cas-custom 3
trunk-group 11
cas-custom 4
trunk-group 11
cas-custom 5
trunk-group 11
cas-custom 6
trunk-group 11
!
controller T1 7/2
framing esf
linecode b8zs
pri-group timeslots 1-24

```

```
!  
controller T1 7/3  
  framing esf  
  linecode b8zs  
  pri-group timeslots 1-24  
!  
controller T1 7/4  
  framing esf  
  linecode b8zs  
  pri-group timeslots 1-24 nfas_d primary nfas_int 1 nfas_group 1  
!  
controller T1 7/5  
  framing esf  
  linecode b8zs  
  pri-group timeslots 1-24 nfas_d none nfas_int 2 nfas_group 1  
!  
controller T1 7/6  
  framing sf  
  linecode ami  
!  
controller T1 7/7  
  framing sf  
  linecode ami  
!  
!  
interface FastEthernet0/0  
  ip address 172.16.50.13 255.255.0.0  
  no ip route-cache  
  no ip mroute-cache  
  duplex auto  
  speed auto  
!  
interface FastEthernet0/1  
  ip address 172.20.140.96 255.255.0.0  
  no ip route-cache  
  no ip mroute-cache  
  duplex auto  
  speed auto  
!  
interface Serial7/2:23  
  no ip address  
  trunk-group 12  
  isdn switch-type primary-5ess  
  isdn incoming-voice modem  
!  
interface Serial7/3:23  
  no ip address  
  trunk-group 13  
  isdn switch-type primary-5ess  
  isdn incoming-voice modem  
  isdn T310 60000  
  no cdp enable  
!  
interface Serial7/4:23  
  no ip address  
  trunk-group 13  
  isdn switch-type primary-5ess  
  isdn incoming-voice modem  
  isdn guard-timer 3000  
  isdn T306 30000  
  isdn T310 10000  
  no cdp enable  
!  
ip classless
```

```

ip route 0.0.0.0 0.0.0.0 172.16.0.1
ip route 192.168.254.0 255.255.255.0 FastEthernet0/0
no ip http server
!
!
trunk group 11
  carrier-id s-mci
!
trunk group 12
  carrier-id s-att
!
trunk group 13
  carrier-id s-sprint
!
voice-port 7/1:1
!
voice-port 7/1:2
!
voice-port 7/1:3
!
voice-port 7/1:4
!
voice-port 7/1:5
!
voice-port 7/1:6
!
voice-port 7/2:D
!
voice-port 7/3:D
!
voice-port 7/4:D
!
dial-peer voice 5108888 voip
  destination-pattern 510888....
  session protocol sipv2
  session target ipv4:172.16.50.14
!
dial-peer voice 1311 pots
  carrier-id source s-alpha
  call-block disconnect-cause incoming user-busy
  call-block translation-profile incoming cb_xrule
  translation-profile incoming xrule
  direct-inward-dial
!
dial-peer voice 1312 pots
  carrier-id source s-beta
  call-block translation-profile incoming cb_xrule
  translation-profile incoming xrule
!
dial-peer voice 1313 pots
  carrier-id source s-gamma
  call-block translation-profile incoming cb_xrule
  translation-profile incoming xrule
  direct-inward-dial
!
dial-peer voice 510889 voip
  destination-pattern 510889....
  session protocol sipv2
  session target enum:1
!
dial-peer voice 510890 voip
  destination-pattern 510890....
  session target enum:1
!

```

```

dial-peer voice 999 voip
 destination-pattern 510.....
 session target ipv4:172.16.50.14
!
exec-timeout 0 0
 logging synchronous
line aux 0
 logging synchronous
line vty 0 4
 login
line 1/00 1/107
 no flush-at-activation
 modem InOut
line 3/00 3/107
 no flush-at-activation
 modem InOut
!
scheduler allocate 10000 400
end

```

Configuring Trunk and Carrier Based Routing on a Terminating Gateway: Example

The following output shows how to configure trunk and carrier based routing on the terminating gateway.

```

!
version 12.2
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname second
!
!
resource-pool disable
dial-tdm-clock priority 1 freerun
!
voice-fastpath enable
no ip subnet-zero
no ip domain-lookup
ip host second 192.168.254.254 255.255.255.0
ip name-server 172.21.188.171
!
no ip dhcp-client network-discovery
lcp max-session-starts 0
isdn switch-type primary-dms100
voice call carrier capacity active
voice call carrier capacity timer interval 45
!
voice source-group sg-alpha
 carrier-id source s-alpha
 carrier-id target t-alpha
 description route calls from source alpha to target alpha
 translation-profile incoming xrule
!
voice source-group sg-beta
 carrier-id source s-beta
 carrier-id target t-beta

```

```

description route calls from source beta to target beta
translation-profile incoming xrule
!
voice source-group sg-gamma
carrier-id source s-gamma
carrier-id target t-gamma
description route calls from source gamma to target gamma
translation-profile incoming xrule
!
voice source-group sg-ip
access-list 1
translation-profile incoming xrule
!
voice translation-rule 1
rule 1 reject /5108880101/
rule 2 /^510888(01..\)/ /\10101/
rule 3 /5108880.../ /5108880101/
!
voice translation-rule 2
rule 1 /51088802../ /5108880101/
rule 2 /51088803../ /5108880101/
rule 3 /510889.../ /5108880101/
rule 4 /510890.../ /5108880101/
!
voice translation-rule 3
rule 1 /65088801../ /6508880101/
!
voice translation-profile xrule
translate called 1
translate calling 2
translate redirect-called 3
!
fax interface-type modem
mta receive maximum-recipients 0
!
trunk activate port-threshold 33
!
controller T1 7/0
framing sf
linecode ami
!
controller T1 7/1
framing esf
linecode ami
ds0-group 1 timeslots 1-4 type e&m-fgb dtmf dnis
ds0-group 2 timeslots 5-8 type e&m-fgb dtmf dnis
ds0-group 3 timeslots 9-12 type e&m-fgb dtmf dnis
ds0-group 4 timeslots 13-16 type e&m-fgb dtmf dnis
ds0-group 5 timeslots 17-20 type e&m-fgb dtmf dnis
ds0-group 6 timeslots 21 type e&m-fgb dtmf dnis
ds0-group 7 timeslots 22 type e&m-fgb dtmf dnis
ds0-group 8 timeslots 23 type e&m-fgb dtmf dnis
ds0-group 9 timeslots 24 type e&m-fgb dtmf dnis
ds0 busyout 22-23 hard
cas-custom 1
trunk-group 11 1
cas-custom 2
trunk-group 11 2
cas-custom 3
trunk-group 11 3
cas-custom 4
trunk-group 11 4
cas-custom 5
trunk-group 11 5

```

```
cas-custom 6
trunk-group 12 1
cas-custom 7
trunk-group 12 2
cas-custom 8
trunk-group 12 3
cas-custom 9
trunk-group 12 4
!
controller T1 7/2
framing esf
linecode b8zs
pri-group timeslots 1-24 nfas_d primary nfas_int 1 nfas_group 1
!
controller T1 7/3
framing esf
linecode b8zs
pri-group timeslots 1-24 nfas_d none nfas_int 2 nfas_group 1
!
controller T1 7/4
framing esf
linecode ami
ds0-group 1 timeslots 1-4 type e&m-fgb dtmf dnis
ds0-group 2 timeslots 5-8 type e&m-fgb dtmf dnis
ds0-group 3 timeslots 9-12 type e&m-fgb dtmf dnis
ds0-group 4 timeslots 13-16 type e&m-fgb dtmf dnis
ds0-group 5 timeslots 17-20 type e&m-fgb dtmf dnis
ds0-group 6 timeslots 21 type e&m-fgb dtmf dnis
ds0-group 7 timeslots 22 type e&m-fgb dtmf dnis
ds0-group 8 timeslots 23 type e&m-fgb dtmf dnis
ds0-group 9 timeslots 24 type e&m-fgb dtmf dnis
ds0 busyout 21-22,24 hard
cas-custom 1
trunk-group 11 5
cas-custom 2
trunk-group 11 6
cas-custom 3
trunk-group 11 7
cas-custom 4
trunk-group 11 8
cas-custom 5
trunk-group 12 5
cas-custom 6
trunk-group 12 5
cas-custom 7
trunk-group 12 5
cas-custom 8
trunk-group 12 5
cas-custom 9
trunk-group 12 5
!
controller T1 7/5
framing esf
linecode b8zs
pri-group timeslots 1-24
!
controller T1 7/6
framing sf
linecode ami
!
controller T1 7/7
framing sf
linecode ami
!
```

```

interface FastEthernet0/0
 ip address 172.16.50.14 255.255.0.0
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.0.0.211 255.0.0.0
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
!
interface Serial7/2:23
 no ip address
 isdn switch-type primary-dms100
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.0.1
ip route 192.168.254.0 255.255.255.0 FastEthernet0/0
ip route 192.168.254.0 255.255.255.0 172.16.0.1
no ip http server
!
access-list 1 permit 172.16.50.12
!
!
trunk group 11
 carrier-id t-alpha
 hunt-scheme longest-idle
!
trunk group 12
 carrier-id t-alpha
 hunt-scheme least-idle
!
trunk group 13
 carrier-id t-beta
 hunt-scheme random
!
trunk group 14
 carrier-id t-gamma
 hunt-scheme random
!
call rsvp-sync
!
voice-port 7/1:1
!
voice-port 7/1:2
!
voice-port 7/1:3
!
voice-port 7/1:4
!
voice-port 7/1:5
!
voice-port 7/1:6
!
voice-port 7/1:7
!
voice-port 7/1:8
!
voice-port 7/1:9
!
voice-port 7/2:D

```

```
!
voice-port 7/4:1
!
voice-port 7/4:2
!
voice-port 7/4:3
!
voice-port 7/4:4
!
voice-port 7/4:5
!
voice-port 7/4:6
!
voice-port 7/4:7
!
voice-port 7/4:8
!
voice-port 7/4:9
!
voice-port 7/5:D
!
mgcp profile default
!
dial-peer voice 20001 voip
  carrier-id source alpha
!
dial-peer voice 20002 voip
  carrier-id source beta
!
dial-peer voice 20003 voip
  carrier-id source gamma
!
dial-peer voice 10001 pots
  trunkgroup 11 1
  trunkgroup 12 2
  translation-profile outgoing 1
  carrier-id target t-alpha
  forward-digits all
!
dial-peer voice 10002 pots
  trunkgroup 13
  translation-profile outgoing 2
  carrier-id target t-beta
!
dial-peer voice 10003 pots
  trunkgroup 14
  translation-profile outgoing 1
  carrier-id target t-gamma
  forward-digits all
!
line con 0
  exec-timeout 0 0
  logging synchronous
line aux 0
  logging synchronous
line vty 0 4
  login
line 1/00 1/107
  no flush-at-activation
  modem InOut
!
scheduler allocate 10000 400
end
```

Command Reference

All commands used with this feature are documented in the [Cisco IOS Voice Command Reference, Release 12.3](#).