



# Configuring SIP Call-Transfer Features

**Last Updated: September 28, 2012**

This chapter describes how to configure SIP call-transfer features. It describes the following features:

- SIP - Call Transfer Using Refer Method
- SIP - Call Transfer Enhancements Using Refer Method
- SIP Transfer Using the Refer Method and Call Forwarding
- SIP Stack Portability



**Note**

The SIP Stack Portability feature is described in the “Configuring SIP, Timer, and Response Features” chapter.

## Feature History for SIP - Call Transfer Using Refer Method

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB2	The feature was implemented on an additional platform.
12.2(11)T	The feature was integrated into this release and support was added for additional platforms.

## Feature History for SIP - Call Transfer Enhancements Using Refer Method

Release	Modification
12.2(13)T	This feature was introduced.



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

### Feature History for SIP Transfer Using the Refer Method and Call Forwarding

Release	Modification
12.2(13)T	This feature was introduced.

- [Finding Feature Information, page 2](#)
- [Prerequisites for SIP Call Transfer, page 2](#)
- [Restrictions for SIP Call Transfer, page 3](#)
- [Information About SIP Call Transfer, page 4](#)
- [How to Configure SIP Call-Transfer Features, page 20](#)
- [Configuration Examples for SIP Call-Transfer Features, page 41](#)
- [Additional References, page 46](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Prerequisites for SIP Call Transfer

### All SIP Call-Transfer Features

- Establish a working IP network and configure VoIP.



#### Note

For information about configuring VoIP, see "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms".

- Ensure that the gateway has voice functionality configured for SIP.
- Ensure that your Cisco router has minimum memory requirements.
- With all SIP call-transfer methods, configure dial peers for correct functioning of the Refer method.



#### Note

For dial-peer configuration steps, see the "Configure SIP Call Transfer on a POTS Dial Peer".

- As necessary, configure the router to use Greenwich Mean Time (GMT). SIP requires that all times be sent in GMT. The INVITE is sent in GMT. However, the default for routers is to use Coordinated Universal Time (UTC). To configure the router to use GMT, issue the **clock timezone** command in global configuration mode and specify GMT.

### SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Feature

- Load Cisco IOS Release 12.2(15)T or a later release.
- Configure hookflash signaling.
- Write a Tool Command Language (Tcl) Interactive Voice Response (IVR) 2.0 script that implements Cisco IOS call-transfer and call-forward functionality.

## Restrictions for SIP Call Transfer

### All SIP Call-Transfer Features

- The SIP gateway does not support codecs other than those listed in the table titled “SIP Codec Support by Platform and Cisco IOS Release” in the “Enhanced Codec Support for SIP Using Dynamic Payloads” section of the “Configuring SIP QoS Features” document.
- SIP requires that all times be sent in GMT.
- Although SIP Cisco IOS gateways currently support SIP URLs and TEL URLs, the Refer-To header and the Also header must be in SIP URL format to be valid. The TEL URL is only supported in the Refer-To header for blind transfer. The TEL URL format cannot be used because it does not provide a host portion, and without one, the triggered Invite request cannot be routed.
- Only three overloaded headers in the Refer-to header are accepted: Accept-Contact, Proxy-Authorization, and Replaces. All other headers present in the Refer-To are ignored.
- The Refer-To and Contact headers are required in the Refer request. The absence of either header results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4xx class response.
- The Referred-By header is required in a Refer request. The absence of this header results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4xx class response.
- With all SIP call-transfer methods, dial peers must be configured for correct functioning of the Refer method.



#### Note

For dial-peer configuration steps, see "Configure SIP Call Transfer on a POTS Dial Peer".

- With call transfer using the Bye method, the Requested-By header identifies the party initiating the transfer. The Requested-By header is included in the INVITE request that is sent to the transferred-to party only if a Requested-By header was also included in the Bye request.
- With call transfer using the Also method, the Also header identifies the transferred-to party. To invoke a transfer, the user portion of the Also header must be defined explicitly or with wildcards as a destination pattern on a VoIP dial peer. The transferred call is routed using the session target parameter on the dial peer instead of the host portion of the Also header. Therefore, the Also header can contain user@host, but the host portion is ignored for call routing purposes.
- The grammar for the Also and Requested-By headers is not fully supported. Only the name-addr is supported. This implies that the crypto-param, which might be present in the Bye request, is not populated in the ensuing Invite to the transferred-to party.
- Cisco SIP gateways do not support the “user=np-queried” parameter in a Request URI.
- If a Cisco SIP gateway receives an ISDN Progress message, it generates a 183 Session progress message. If the gateway receives an ISDN ALERT, it generates a 180 Ringing message.
- The SIP gateway requires each INVITE to include a Session Description Protocol (SDP) header.

- The contents of the SDP header cannot change between the 180 Ringing message and the 200 OK message.
- VoIP dial peers allow a user to configure the bytes parameter associated with a codec. Cisco SIP gateways present or respond to the a=ptime parameter in the SDP body of a SIP message. However, only one a=ptime attribute is allowed per m-line block.
- If early transfer is attempted, and the call between the originator and final-recipient involves QoS or RSVP, the triggered Invite from the recipient with the Replaces header is not processed and the transfer fails. The session between the originator and the final-recipient remains unchanged.

### **SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Feature**

- SIP call transfer and call forwarding using Tcl IVR 2.0 and VoiceXML applications feature is supported only through Tcl IVR 2.0 and VoiceXML applications; the feature is not supported for Tcl IVR 1.0 applications or the DEFAULT session application.
- Only Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers support the initiating of call transfer and call redirection.
- Cisco SIP customer premise equipment (CPE) such as 79xx and Analog Telephone Adaptors (ATAs) do not currently support TEL URLs.
- RLT on CAS or analog (FXS) ports are necessary to initiate SIP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS, FXO, T1, E1, and CAS phones are not supported.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

## **Information About SIP Call Transfer**

- [SIP Call-Transfer Basics, page 4](#)
- [SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications, page 14](#)

## **SIP Call-Transfer Basics**

- [Basic Terminology of SIP Call Transfer, page 4](#)
- [Types of SIP Call Transfer Using the Refer Method, page 7](#)

### **Basic Terminology of SIP Call Transfer**

The Refer method provides call-transfer capabilities to supplement the Bye and Also methods already implemented on Cisco IOS SIP gateways.

Call transfer allows a wide variety of decentralized multiparty call operations. These decentralized call operations form the basis for third-party call control, and thus are important features for VoIP and SIP. Call transfer is also critical for conference calling, where calls can transition smoothly between multiple point-to-point links and IP level multicasting.

- [Refer Method, page 5](#)

- [Notify Method, page 6](#)

## Refer Method

The SIP Refer method provides call-transfer capabilities to supplement the Bye and Also methods already implemented on Cisco IOS SIP gateways. The Refer method has three main roles:

- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.



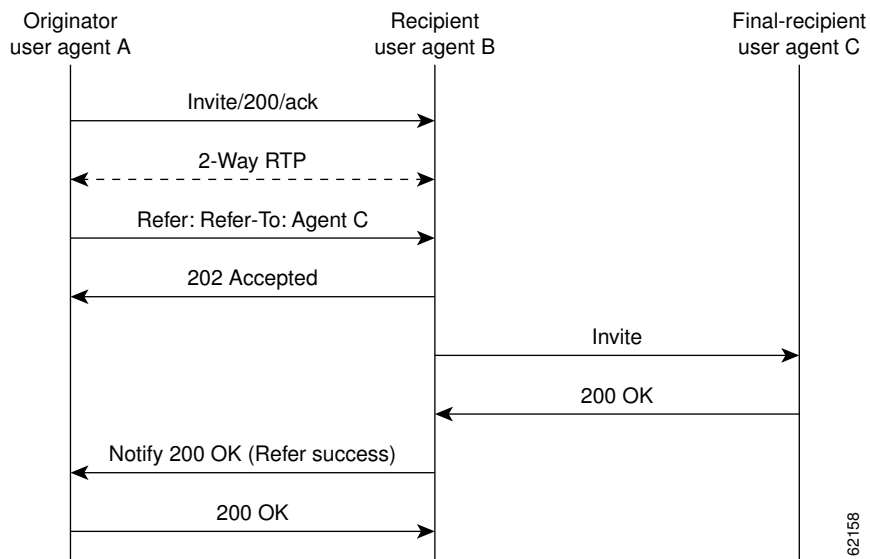
### Note

A gateway can be a recipient or final-recipient; but not an originator.

The Refer method always begins within the context of an existing call and starts with the *originator*. The originator sends a Refer request to the *recipient* (user agent receiving the Refer request) to initiate a triggered Invite request. The triggered Invite request uses the SIP URL contained in the Refer-To header as the destination of the Invite request. The recipient then contacts the resource in the Refer-To header (*final-recipient*), and returns a SIP 202 (Accepted) response to the originator. The recipient also must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify Method, SIP's event notification mechanism. A Notify message with a message body of SIP 200 OK indicates a successful transfer, while a body of SIP 503 Service Unavailable indicates an unsuccessful transfer. If the call was successful, a call between the recipient and the final-recipient results.

The figure below shows the call flow of a successful Refer transaction initiated within the context of an existing call.

**Figure 1 Successful Refer Transaction**



### Refer-To Header

The recipient receives from the originator a Refer request that always contains a single Refer-to header. The Refer-to header includes a SIP URL that indicates the party to invite and must be in SIP URL format.

**Note**

The TEL URL format cannot be used in a Refer-to header, because it does not provide a host portion, and without one, the triggered Invite request cannot be routed.

The Refer-To header may contain three additional overloaded headers to form the triggered Invite request. If any of these three headers are present, they are included in the triggered Invite request. The three headers are:

- **Accept-Contact**--Optional in a Refer request. A SIP IOS gateway that receives an Invite request with an Accept-Contact does not act upon this header. This header is defined in draft-ietf-sip-callerprefs-03.txt and may be used by user agents that support caller preferences.
- **Proxy-Authorization**--A nonstandard header that SIP gateways do not act on. It is echoed in the triggered Invite request because proxies occasionally require it for billing purposes.
- **Replaces**--The Replaces header is used by SIP gateways to indicate whether the originator of the Refer request is requesting a blind or attended transfer. It is required if the originator is performing an attended transfer, and not required for a blind transfer.

All other headers present in the Refer-To are ignored, and are not sent in the triggered invite.

**Note**

The Refer-To and Contact headers are required in the Refer request. The absence of these headers results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4xx class response.

**Referred-By Header**

The Referred-By header is required in a Refer request. It identifies the originator and may also contain a signature (included for security purposes). SIP gateways echo the contents of the Referred-By header in the triggered Invite request, but on receiving an Invite request with this header, gateways do not act on it.

**Note**

The Referred-By header is required in a Refer request. The absence of this header results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4xx class response.

**Notify Method**

Once the outcome of the Refer transaction is known, the recipient of the Refer request must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify method, SIP's event notification mechanism. The notification contains a message body with a SIP response status line and the response class in the status line indicates the success or failure of the Refer transaction.

The Notify message must:

- Reflect the same To, From, and Call-ID headers that were received in the Refer request.
- Contain an Event header refer.
- Contain a message body with a SIP response line. For example: SIP/2.0 200 OK to report a successful Refer transaction, or SIP/2.0 503 Service Unavailable to report a failure. To report that the recipient disconnected before the transfer finished, it must use SIP/2.0 487 Request Canceled.

Two Cisco IOS commands pertain to the Notify method.

- The **timers notify** command sets the amount of time that the recipient should wait before retransmitting a Notify message to the originator.
- The **retry notify** command configures the number of times a Notify message is retransmitted to the originator.

**Note**

For information on these commands, see the *Cisco IOS Voice Command Reference*.

## Types of SIP Call Transfer Using the Refer Method

This section discusses how the Refer method facilitates call transfer.

There are two types of call transfer: blind and attended. The primary difference between the two is that the Replaces header is used in attended call transfers. The Replaces header is interpreted by the final-recipient and contains a Call-ID header, indicating that the initial call leg is to be replaced with the incoming Invite request.

As outlined in the Refer method, there are three main roles:

- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

A gateway can be a recipient or final-recipient but not an originator.

- [Blind Call-Transfer Process, page 7](#)
- [Attended Transfer, page 9](#)
- [Attended Transfer with Early Completion, page 11](#)
- [VSA for Call Transfer, page 13](#)

### Blind Call-Transfer Process

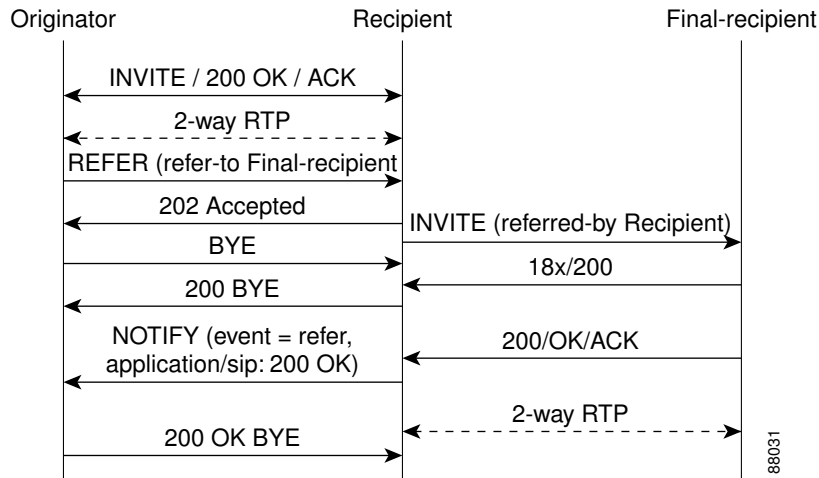
A blind, or unattended, transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. This is different from a consultative, or attended, transfer in which one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller to the third party. Blind transfers are often preferred by automated devices that do not have the capability to make consultation calls.

The basic process of blind transfers works as described in the figure below. In blind transfer, the originator (user agent that initiates the transfer or Refer request) sets up a call with the recipient (user agent that receives the Refer request). After the originator issues a Refer request to the recipient, the recipient, triggered by the Refer request, sends an Invite request to the final-recipient (user agent introduced into a call with the recipient). The recipient returns a SIP 202 (Accepted) response to the originator, and notifies the originator of the outcome of the Refer transaction--if the final-recipient was successfully (SIP 200 OK) or unsuccessfully (SIP 503 Service Unavailable) contacted.

If successful, a call is established between the recipient and the final-recipient. The original signaling relationship between the originator and recipient is terminated when a Bye request is sent by one of the parties. On a successful transfer, if the originator does not send a Bye request after receiving an acknowledgement for the Notify message, the recipient initiates a Bye request. The figure below shows a

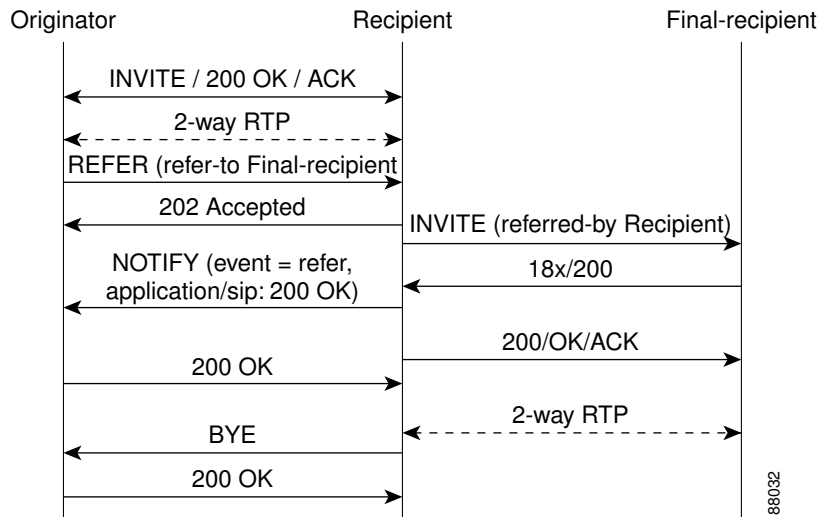
successful blind or unattended call transfer in which the originator initiates a Bye request to terminate signaling with the recipient.

**Figure 2 Successful Blind or Unattended Transfer--Originator Initiating a Bye Request**



The figure below shows a successful blind or unattended call transfer in which the recipient initiates a Bye request to terminate signaling with the originator. A Notify message is always sent by the recipient to the originator after the final outcome of the call is known.

**Figure 3 Successful Blind or Unattended Transfer --Recipient Initiating a Bye Request**



If a failure occurs with the triggered Invite to the final-recipient, the call between the originator and the recipient is not disconnected. The originator sends a re-Invite which takes the call off hold and returns to the original call with the recipient. With prior blind transfer functionality, if the recipient receives an 18x informational response from the final-recipient and then the call fails, the originator can not recover the call with the recipient.

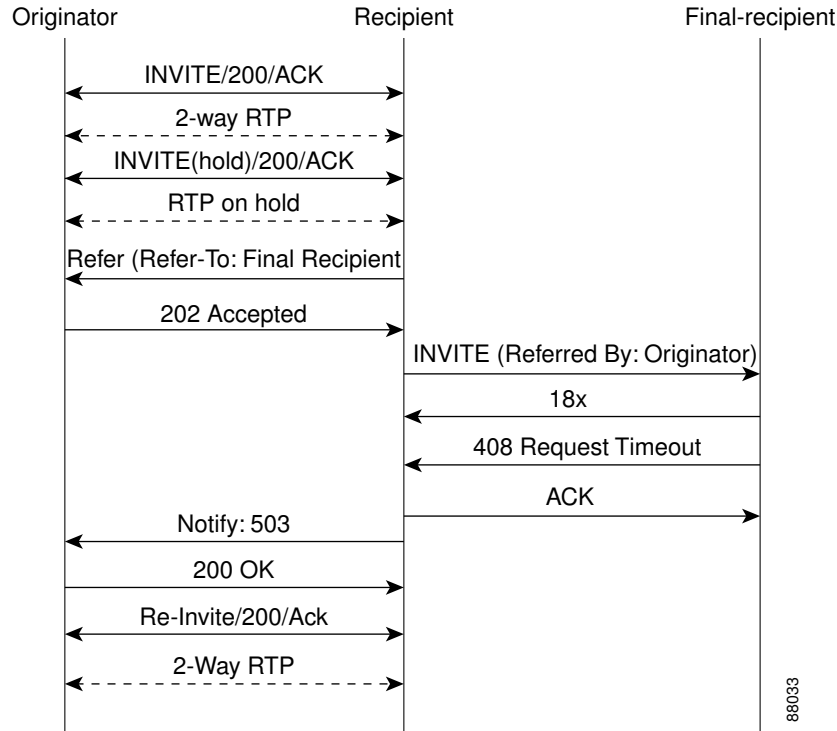
A failure can be caused by an error condition or timeout.

The figure below shows that the call leg between the originator and the recipient remains active. Thus, if the Invite to the final-recipient fails (408 Request Timeout), the recipient notifies the originator of the



failure with a Notify message. The originator sends a re-Invite and returns to the original call with the recipient.

**Figure 4** Failed Blind Transfer--Originator Returns to Original Call with Recipient



**Attended Transfer**

In attended transfers, the Replaces header is inserted by the initiator of the Refer request as an overloaded header in the Refer-To and is copied into the triggered Invite request sent to the final-recipient. The header has no affect on the recipient, but is interpreted by the final-recipient as a way to distinguish between blind transfer and attended transfer. The attended transfer process is described in the table below.

**Table 1** Attended Transfer Process

Process	Description or Detail
1 Originator sets up a call with the recipient.	After the call is set up, originator places recipient on hold.
1 Originator establishes a call to the final-recipient.	--
1 Originator sends recipient a Refer request with an overloaded Replaces header in the Refer-To header.	--

Process	Description or Detail
1 Upon receipt of the Refer request, recipient sends a triggered Invite request to the final-recipient.	The Invite request received by final-recipient includes the Replaces header, identifying the call leg between the originator and final-recipient.
1 Recipient returns a SIP 202 (Accepted) response to the originator.	The SIP 202 (Accepted) acknowledges that the Invite has been sent.
1 Final-recipient establishes a direct signaling relationship with recipient.	Receipt of the Replaces header is what indicates that the initial call leg is to be shut down and replaced by the incoming Invite request.
1 Recipient notifies originator of the outcome of the Refer transaction.	Recipient notifies the originator if the final-recipient was successfully or unsuccessfully contacted.
1 Recipient terminates the session with originator by sending a Bye request.	--

### Replaces Header

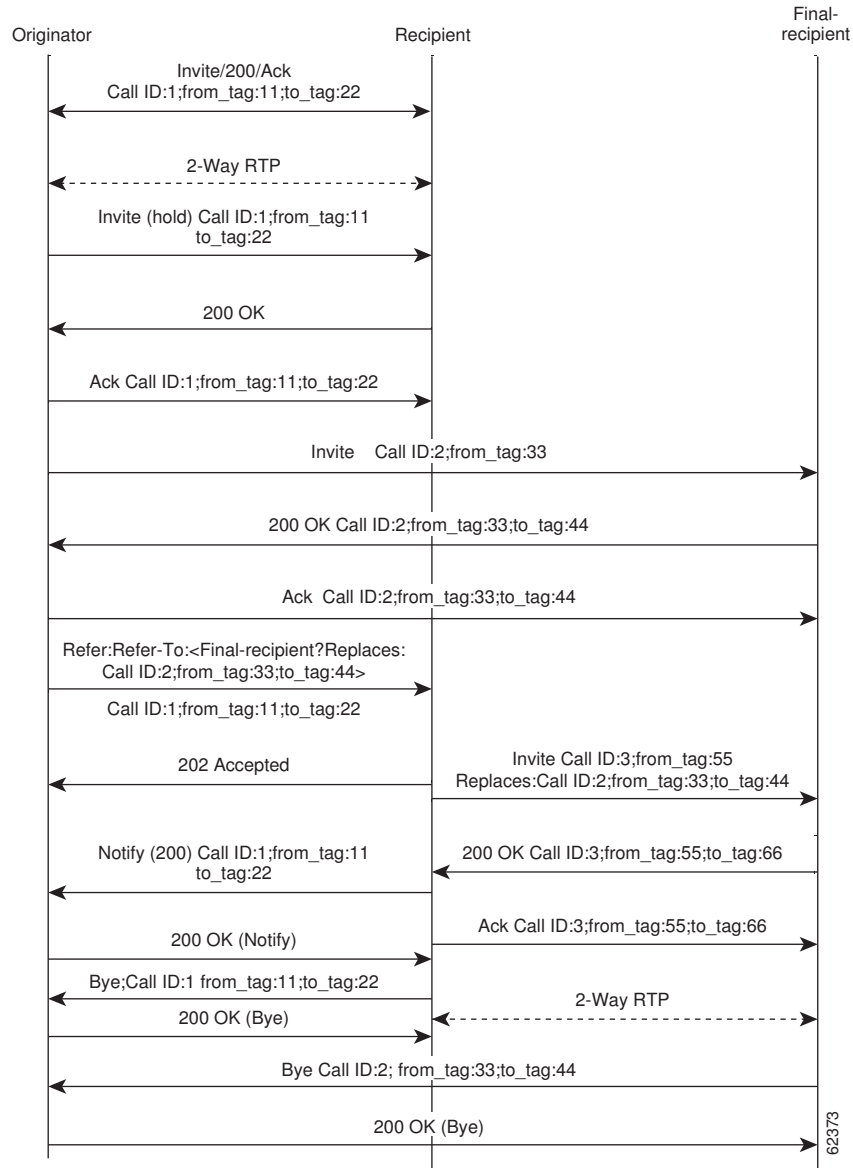
The Replaces header is required in attended transfers. It indicates to the final-recipient that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming Invite request. The final-recipient sends a Bye request to the originator to terminate its session.

If the information provided by the Replaces header does not match an existing call leg, or if the information provided by the Replaces header matches a call leg but the call leg is not active (a Connect, 200 OK to the Invite request has not been sent by the final-recipient), the triggered Invite does not replace the initial call leg and the triggered Invite request is processed normally.

Any failure resulting from the triggered Invite request from the recipient to final-recipient does not destroy the call between the originator and the final-recipient. In these scenarios, all calls that are active (originator

to recipient and originator to final-recipient) remain active after the failed attended transfer attempt. The figure below shows a call flow for a successful attended transfer.

**Figure 5 Successful Attended Transfer**



**Attended Transfer with Early Completion**

Attended transfers allow the originator to have a call established between both the recipient and the final-recipient. With attended transfer with early completion, the call between the originator and the final-recipient does not have to be active, or in the talking state, before the originator can transfer it to the recipient. The originator establishes a call with the recipient and only needs to be in the process of setting up a call with the final-recipient. The final-recipient may be ringing, but has not answered the call from the originator when it receives a re-Invite to replace the call with the originator and the recipient. The figure

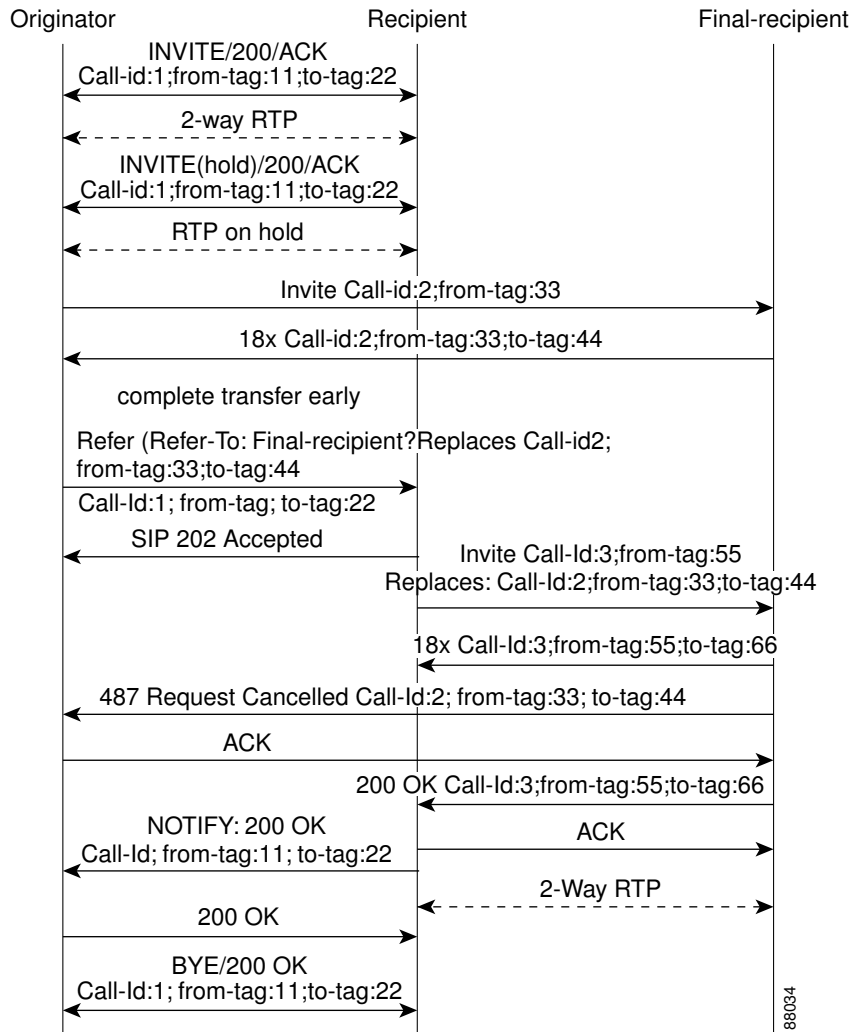
below shows the process of attended transfer with early completion, and the detailed actions involved are described in the table below.

**Table 2** *Attended Transfer with Early Completion Process*

Process	Description or Detail
1 Originator sets up a call with recipient.	After the call is set up, originator places recipient on hold.
1 Originator contacts final-recipient.	--
1 When originator gets an indication that final-recipient is ringing, it sends recipient a Refer request with an overloaded Replaces header in the Refer-to header.	The Replaces header is required in attended transfers and distinguishes between blind transfer and attended transfers.
1 Recipient returns a SIP 202 (Accepted) response to originator.	The SIP 202 (Accepted) acknowledges that the Invite has been sent.
1 Upon receipt of the Refer request, recipient sends a triggered Invite request to final-recipient.	The Invite request received by final-recipient includes the Replaces header, which indicates that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming Invite request.
1 Final-recipient establishes a direct signaling relationship with recipient.	Final-recipient tries to match the Call-ID header and the To or From tags in the Replaces header of the incoming Invite with an active call leg in its call control block. If a matching active call leg is found, final-recipient replies with exactly the same status as the found call leg. However it then terminates the found call leg with a 487 Request Cancelled response.
<b>Note</b> If early transfer is attempted and the call involves quality of service (QoS) or Resource Reservation Protocol (RSVP), the triggered Invite from the recipient with the Replaces header is not processed and the transfer fails. The session between originator and final-recipient remains unchanged.	
1 Recipient notifies originator of the outcome of the Refer transaction--that is, whether final-recipient was successfully or unsuccessfully contacted.	--

Process	Description or Detail
1	Recipient or originator terminates the session by sending a Bye request.

**Figure 6 Attended Transfer with Early Completion**



**VSA for Call Transfer**

You can use a vendor-specific attribute (VSA) for SIP call transfer.

**Referred-By Header**

For consistency with existing billing models, the Referred-By and Requested-By headers are populated in call history tables as a VSA. Cisco VSAs are used for VoIP call authorization. The new VSA tag **supp-svc-xfer-by** helps to associate the call-legs for Call Detail Records (CDR) generation. The call-legs could be originator to recipient or recipient to final-recipient.

The new VSA tag **supp-svc-xfer-by** contains the user@host portion of the SIP URL of the Referred-By header for transfers performed with the Refer method. For transfers performed with the Bye/Also method, the tag contains the user@host portion of the SIP URL of the Requested-By header. For each call on the gateway, there are two RADIUS records that are generated: start and stop. The **supp-svc-xfer-by** VSA is only generated for stop records and is only generated on the recipient gateway--the gateway receiving the Refer or Bye/Also message.

The VSA is generated when a gateway that acts as a recipient receives a Refer or Bye/Also message with the Referred-By or Requested-By headers. There are usually two pairs of start and stop records. There is a start and stop record between the recipient and the originator and also between the recipient to final-recipient. In the latter case, the VSA is generated between the recipient to final-recipient only.

### Business Group Field

A new business group VSA field has also been added that assists service providers with billing. The field allows service providers to add a proprietary header to call records. The VSA tag for business group ID is **cust-biz-grp-id** and is only generated for stop records. It is generated when the gateway receives an initial Invite with a vendor dial-plan header to be used in call records. In cases when the gateway acts as a recipient, the VSA is populated in the stop records between the recipient and originator and the recipient final-recipient.



#### Note

---

For more information about VSAs and CDRs, see the *CDR Accounting for Cisco IOS Voice Gateways* guide.

---

## SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications

- [SIP Call Transfer and Call Forwarding with a Tcl IVR Script, page 14](#)
- [Release Link Trunking on SIP Gateways, page 15](#)
- [SIP Gateway Initiation of Call Transfers, page 17](#)
- [SIP Call Forwarding, page 19](#)

### SIP Call Transfer and Call Forwarding with a Tcl IVR Script

When using a Tcl IVR 2.0 application, you can implement SIP support of blind or attended call-transfer and call-forwarding requests from a Cisco IOS gateway. A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. An attended transfer is one that is consultative--one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller. Blind transfers are often preferred by automated devices that do not have the capability to make consultative calls.

Before implementing blind transfer and call forwarding, you must write a custom Tcl IVR 2.0 script that implements call transfer and call forwarding. The script is responsible for receiving the hookflash event, providing dial tone, matching against the dial plan, initiating call transfer, and reestablishing the original call if the transfer attempt fails.

**Note**

---

For information on writing a Tcl IVR script, see the *Tcl IVR API Version 2.0 Programmer's Guide*.

---

When the Tcl IVR script runs on the Cisco gateway, it can respond to requests to initiate blind call transfer (transfer without consultation) on a SIP call leg. SIP call forwarding on ephones (IP phones that are not configured on the gateway) is also supported.

**Note**

---

SIP Call Transfer and Call Forwarding is compliant with Voice Extensible Markup Language (VXML). VXML scripts can also be used to implement call transfer and call forwarding.

---

## Release Link Trunking on SIP Gateways

RLT functionality has been added to Cisco IOS SIP gateways. With RLT functionality, SIP call transfer can now be triggered by CAS trunk signaling, which the custom Tcl IVR application can monitor. After a SIP call transfer has transpired and the CAS interface is no longer required, the CAS interface can be released.

The RLT functionality can be used to initiate blind transfers on SIP gateways. Blind call transfer uses the Refer method. A full description of blind transfer and the refer Method can be found in "Call Transfer Capabilities Using the Refer Method" documentation.

### RLT and SIP Call Transfers

Call transfer can be triggered by CAS trunk signaling and then captured by the custom Tcl IVR script on a gateway. The process begins with the originator (the SIP user agent that initiates the transfer or Refer request) responding with a dial tone once the originator receives the signal or hookflash from the PSTN call leg. The originator then prepares to receive dual-tone multifrequency (DTMF) digits that identify the final-recipient (the user agent introduced into a call with the recipient).

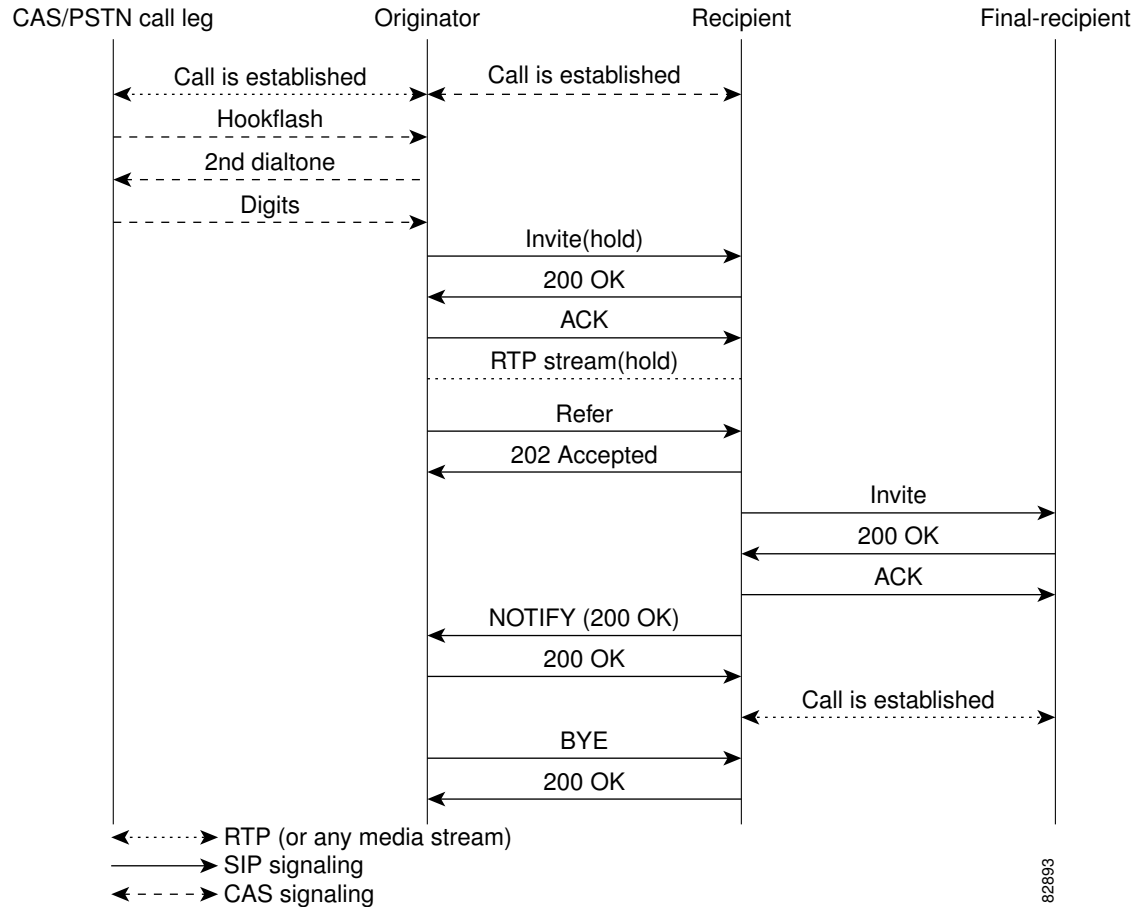
Once the first DTMF digit is received, the dial tone is discontinued. DTMF-digit collection is not completed until a 4-second interdigit timeout occurs, or an on-hook is received on that specific CAS time slot. Call transfer starts when DTMF-digit collection is successful. If digit collection fails, for example if not enough DTMF digits or invalid digits are collected, the initial call is reestablished.

Once the DTMF digits are successfully collected, the custom Tcl IVR script can initiate call transfer. SIP messaging begins when the transfer is initiated with the Refer method. The originator sends an Invite to the recipient (the user agent that receives the Refer request and is transferred to the final-recipient) to hold the call and request that the recipient not return Real-Time Transport Protocol (RTP) packets to the originator. The originator then sends a SIP Refer request to the recipient to start the transfer process. When the recipient receives the request, the recipient returns a 202 *Accepted* acknowledgement to the originator. The Tcl IVR script run by the originator can then release the CAS trunk and close the primary call (see the figure below).

If the recipient does not support the Refer method, a 501 *Not implemented* message is returned. However, for backward compatibility purposes, the call transfer is automatically continued with the Bye/Also method. The originator sends a Bye/Also request to the recipient and releases the CAS trunk with the PSTN call leg. The primary call between the originator and the recipient is closed when a 200 OK response is received.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down.

**Figure 7 Call Transfer Using the Refer Method**



### SIP and TEL URLs in Call Transfers

When the SIP call-transfer originator collects DTMF digits from the CAS trunk, it attempts to find a dial peer. If a dial peer is found, the session target in the dial peer is used to formulate a Session Initiation Protocol Uniform Resource Locator (SIP URL). This URL can be used with both the Refer method and the Bye/Also method. A SIP URL is in the following form:

```
sip:JohnSmith@example.com
```

If a valid dial peer is not found, a Telephone Uniform Resource Locator (TEL URL) is formulated in the Refer-To header. A TEL URL is in the following form:

```
tel:+11231234567
```

The choice of which URL to use is critical when correctly routing SIP calls. For example, the originating gateway can send out a Bye with an Also header, but the Also header can carry only a SIP URL. The Also header cannot carry a TEL URL. That is, if the gateway decides to send a Bye/Also but cannot find a matched dial peer, the gateway reports an error on the transfer gateway and sends a Bye without the Also header.



If the recipient of a SIP call transfer is a SIP phone, the phone must have the capability to interpret either the Refer method or the Bye/Also method for the call transfer to work. If the recipient is a Cisco IOS gateway, there needs to be a matching dial peer for the Refer-To *user* . *User*, looking at the previous example, can be either *JohnSmith* or *11231234567* . The dial peer also needs to have an application session defined, where session can be the name of a Tcl IVR application. If there's no match, a 4xx error is sent back and no transfer occurs. If there's a POTS dial peer match, a call is made to that POTS phone. Before the 12.2(15)T release, if there's a VoIP match, the Refer-To URL is used to initiate a SIP call. In release 12.2(15)T and later releases, the application session target in the dial peer is used for the SIP call.

**Note**

For information on the application session target, see the "Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer".

## SIP Gateway Initiation of Call Transfers

SIP gateways can also initiate, or originate, attended call transfers. The process begins when the originator establishes a call with the recipient. When the user on the PSTN call leg wants to transfer the call, the user uses hookflash to get a second dial tone and then enters the final-recipients number. The Tcl IVR script can then put the original call on hold and set up the call to the final-recipient, making the originator active with the final-recipient. The Refer request is sent out when the user hangs up to transfer the call. The Refer request contains a Replaces header that contains three tags: *SIP CallID* , *from* , and *to* . The tags are passed along in the Invite from the recipient to the final-recipient, giving the final-recipient adequate information to replace the call leg. The host portion of the Refer request is built from the established initial call. The following is an example of a Refer request that contains a Replaces header:

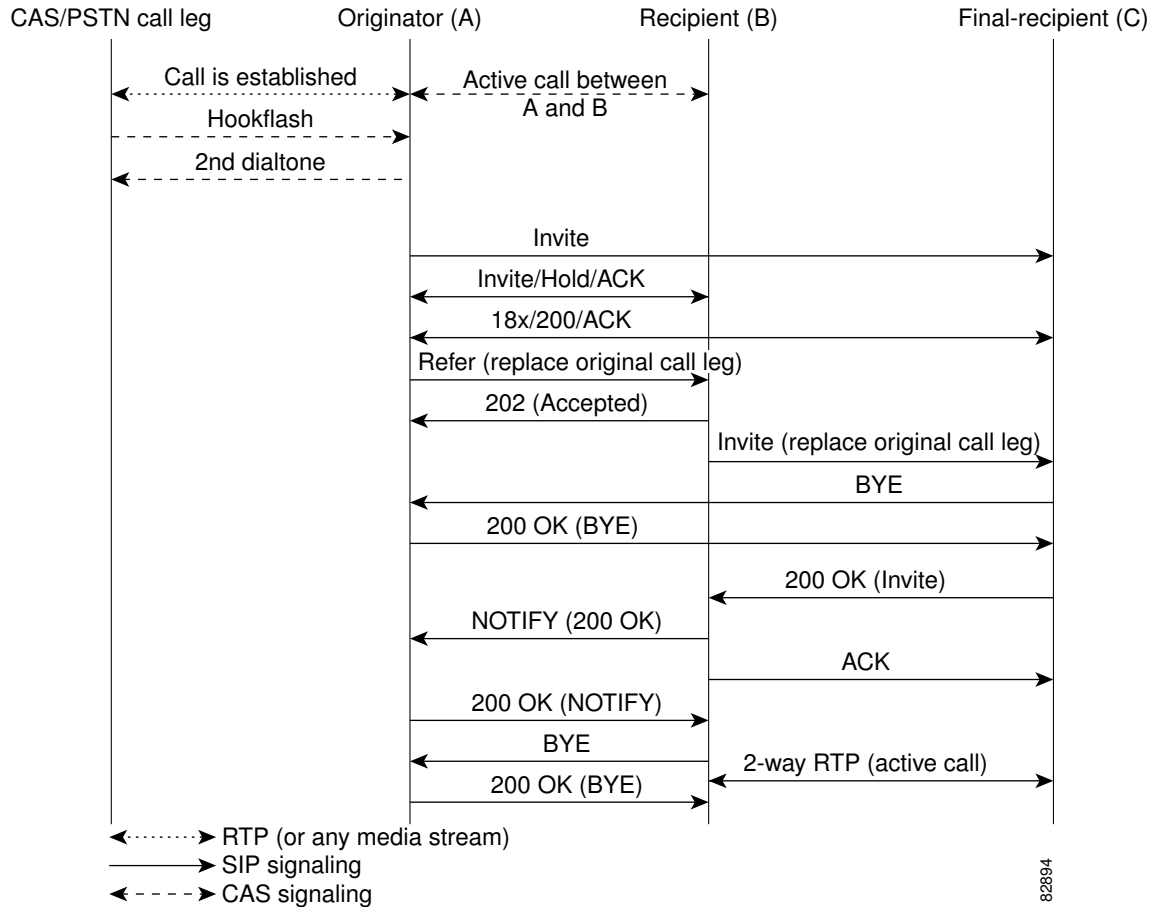
**Note**

IP addresses and hostnames in examples are fictitious.

```
Refer sip:3100801@172.16.190.100:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.16.190.99:5060
From: "5550100" <sip:5550100@172.16.190.187>
To: <sip:3100801@172.16.190.187>;tag=A7C2C-1E8C
Date: Sat, 01 Jan 2000 05:15:06 GMT
Call-ID: c2943000-106ae5-1c5f-3428@172.16.197.182
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 946685709
CSeq: 103 Refer
Refer-To: sip:3100802@10.102.17.217?
Replaces=DD713380-339C11CC-80BCF308-92BA812C@172.16.195.77;to-tag=A5438-23E4;from-
tag=C9122EDB-2408
Referred-By: <sip:3100802@172.16.190.99>
Content-Length: 0
```

After the NOTIFY is received by the originator, the Tcl IVR script can disconnect the call between the originator and the recipient. The call between the originator and the final-recipient is disconnected by the recipient sending a BYE to the originator. The figure below shows a call flow of a successful call transfer.

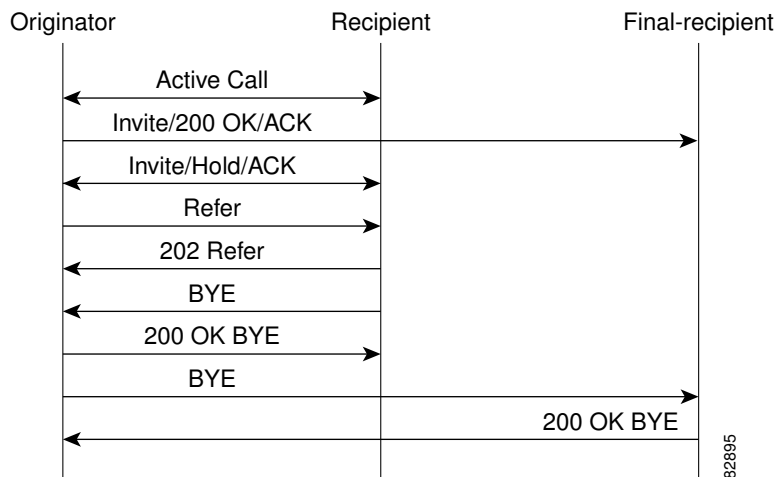
**Figure 8 Successful Attended Call Transfer Initiated by the Originator**



If the recipient does not support the Refer method, a 501 *Not implemented* message is returned.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down. The figure below shows the recipient hanging up the call before the transfer completes. The item to notice is that the NOTIFY message is never sent.

**Figure 9** *Unsuccessful Call Transfer--Recipient Hangs Up Before Transfer Completes*



## SIP Call Forwarding

SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS, FXO, T1, E1, and CAS phones are not supported.

With ephones, there are four different types of SIP call forwarding supported:

- Call Forward Unavailable
- Call Forward No Answer
- Call Forward Busy
- Call Forward Unconditional

In all four of these call forwarding types, a 302 *Moved Temporarily* response is sent to the user agent client. A Diversion header included in the 302 response indicates the type of forward.

The 302 response also includes a Contact header, which is generated by the calling number that is provided by the custom Tcl IVR script. The 302 response also includes the host portion found in the dial peer for that calling number. If the calling number cannot match a VoIP dial-peer or POTS dial-peer number, a 503 *Service Unavailable* message is sent, except in the case of the Call Forward No Answer. With Call Forward No Answer, call forwarding is ignored, the phone rings, and the expires timer clears the call if there is no answer.



### Note

By default, SIP credentials for forwarded calls on Cisco IOS voice gateways are based on the calling number. To globally enable a gateway to use the redirecting number, instead, use the **authenticate redirecting-number** command. To configure this behavior for a specific dial peer on a gateway, use the **voice-class sip authenticate redirecting-number** command. For detailed information, see these commands in the *Cisco IOS Voice Command Reference*.

**Note**

In Cisco IOS Release 12.2(15)T and later releases, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

## How to Configure SIP Call-Transfer Features

**Note**

For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

- [Configuring SIP Call Transfer Using the Refer Method](#), page 20
- [Configuring SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications](#), page 27
- [Verifying SIP Call Transfer](#), page 38
- [Troubleshooting Tips](#), page 41

## Configuring SIP Call Transfer Using the Refer Method

- [Configure SIP Call Transfer on a POTS Dial Peer](#), page 20
- [Configure SIP Call Transfer on a VoIP Dial Peer](#), page 22
- [Configure the SIP Call-Transfer Session Target](#), page 23
- [Configure SIP Refer and Notify Message Settings](#), page 25

### Configure SIP Call Transfer on a POTS Dial Peer

**Note**

To handle all call-transfer situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer for a POTS dial peer.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag pots*
4. **application** *application-name*
5. **destination-pattern** *[+]string[T]*
6. **port** *slot / port*
7. **exit**

## DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <code>enable</code></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p><b>Step 3</b> <code>dial-peer voice tag pots</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 25 pots</pre>	Enters dial-peer configuration mode for the specified POTS dial peer.
<p><b>Step 4</b> <code>application application-name</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# application session</pre>	<p>Enables a specific application on a dial peer. The argument is as follows:</p> <ul style="list-style-type: none"> <li><i>application-name</i> --Name of the predefined application that you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers.</li> </ul>
<p><b>Step 5</b> <code>destination-pattern [+]<i>string</i>[T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination-pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer: Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li><code>+</code> --(Optional) Character indicating an E.164 standard number.</li> <li><i>string</i> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li><b>T</b> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable length dial string.</li> </ul>
<p><b>Step 6</b> <code>port slot / port</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# port 1/1</pre>	<p>Associates a dial peer with a voice slot number and a specific local voice port through which incoming VoIP calls are received.</p> <p><b>Note</b> To find the correct port argument for your router, see the <i>Cisco IOS Voice Command Reference</i>.</p>

Command or Action	Purpose
<b>Step 7</b> <code>exit</code>  <b>Example:</b>  <code>Router(config-dial-peer)# exit</code>	Exits the current mode.

## Configure SIP Call Transfer on a VoIP Dial Peer



**Note** To handle all call-transfer situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer for a VoIP dial peer.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `application application-name`
5. `destination-pattern [+]string [T]`
6. `session target ipv4 :destination-address`
7. `session protocol sipv2`
8. `exit`

### DETAILED STEPS

Command or Action	Purpose
<b>Step 1</b> <code>enable</code>  <b>Example:</b>  <code>Router&gt; enable</code>	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
<b>Step 2</b> <code>configure terminal</code>  <b>Example:</b>  <code>Router# configure terminal</code>	Enters global configuration mode.

Command or Action	Purpose
<p><b>Step 3</b> <code>dial-peer voice tag voip</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 29 voip</pre>	Enters dial-peer configuration mode for the specified dial peer.
<p><b>Step 4</b> <code>application application-name</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# application session</pre>	<p>Enables a specific application on a dial peer. The argument is as follows:</p> <ul style="list-style-type: none"> <li><code>application-name</code> --Name of the predefined application that you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers.</li> </ul>
<p><b>Step 5</b> <code>destination-pattern [+]<i>string</i> [T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination-pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li><code>+</code> --(Optional) Character that indicates an E.164 standard number.</li> <li><code>string</code> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li><code>T</code> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable-length dial string.</li> </ul>
<p><b>Step 6</b> <code>session target ipv4 :<i>destination-address</i></code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# session target ipv4:10.10.1.3</pre>	<p>Specifies a network-specific address for a dial peer. Keyword and argument are as follows:</p> <ul style="list-style-type: none"> <li><code>ipv4 :<i>destination address</i></code> --IP address of the dial peer, in this format: <code>xxx.xxx.xxx.xxx</code></li> </ul>
<p><b>Step 7</b> <code>session protocol sipv2</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# session protocol sipv2</pre>	Configures the VoIP dial peer to use IETF SIP.
<p><b>Step 8</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# exit</pre>	Exits the current mode.

## Configure the SIP Call-Transfer Session Target

To configure the SIP call-transfer session target, perform the following steps.

This task configures a SIP server as a session target. Although it is not required, configuring a SIP server as a session target is useful if there is a Cisco SIP proxy server (CSPS) present in the network. With a CSPS, you can configure the SIP server option and have the interested dial peers use the CSPS by default.

To determine the call-transfer destination on the originator, check if there is a matching dial peer:

- If yes, check the session target for the dial peer. If the session target is a SIP server, configure the SIP server as described in the task below. If the session target is not a SIP server, the session target configured in the VoIP dial peer is used.
- If no, a TEL URL is sent.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **sip-server dns : *host-name***
5. **exit**
6. **dial-peer voice *tag* voip**
7. **destination-pattern [+]*string* [T]**
8. **session target sip-server**
9. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  Example: <pre>Router&gt; enable</pre>	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	<b>configure terminal</b>  Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<b>sip-ua</b>  Example: <pre>Router(config)# sip-ua</pre>	Enters SIP user-agent configuration mode.



Command or Action	Purpose
<p><b>Step 4</b> <code>sip-server dns : host-name</code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# sip-server dns:example.sip.com</pre>	<p>Sets the global SIP server interface to a Domain Name System (DNS) hostname. If you do not specify a hostname, the default DNS defined by the <b>ip name-server</b> command is used.</p>
<p><b>Step 5</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# exit</pre>	<p>Exits the current mode.</p>
<p><b>Step 6</b> <code>dial-peer voice tag voip</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 29 voip</pre>	<p>Enters dial-peer configuration mode for the specified dial peer.</p>
<p><b>Step 7</b> <code>destination-pattern [+]<i>string</i> [T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination- pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <b>+</b> --(Optional) Character that indicates an E.164 standard number.</li> <li>• <i>string</i> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li>• <b>T</b> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable-length dial string.</li> </ul>
<p><b>Step 8</b> <code>session target sip-server</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# session target sip-server</pre>	<p><b>Instructs the dial-peer session target to use the global SIP server .</b> Doing so saves you from having to repeatedly enter the SIP server interface address for each dial peer.</p>
<p><b>Step 9</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# exit</pre>	<p>Exits the current mode.</p>

## Configure SIP Refer and Notify Message Settings

**Note**

The Refer request is initiated by the originating gateway and signals the start of call transfer. Once the outcome of the SIP Refer transaction is known, the recipient of the Refer request notifies the originating gateway of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. Notification is accomplished using the Notify method.

- Configure dial peers for correct functioning of the Refer method.

**Note**

For dial-peer configuration steps, see "Configure SIP Call Transfer on a POTS Dial Peer".

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers notify** *milliseconds*
5. **retry notify** *number*
6. **exit**

**DETAILED STEPS**

Command or Action	Purpose
<b>Step 1</b> <b>enable</b>  <b>Example:</b> Router> enable	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
<b>Step 2</b> <b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b> <b>sip-ua</b>  <b>Example:</b> Router(config)# sip-ua	Enters SIP user-agent configuration mode.

Command or Action	Purpose
<p><b>Step 4</b> <code>timers notify <i>milliseconds</i></code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# timers notify 500</pre>	<p>Sets the amount time that the user agent waits before retransmitting the Notify message. The argument is as follows:</p> <ul style="list-style-type: none"> <li><code>milliseconds</code> --Time, in ms. Range: 100 to 1000. Default: 500.</li> </ul>
<p><b>Step 5</b> <code>retry notify <i>number</i></code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# retry notify 10</pre>	<p>Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows:</p> <ul style="list-style-type: none"> <li><code>number</code> --Number of notify message retries. Range: 1 to 10.</li> </ul>
<p><b>Step 6</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# exit</pre>	<p>Exits the current mode.</p>

## Configuring SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications

- [Load the Tcl IVR Application on the Gateway, page 27](#)
- [Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer, page 29](#)
- [Configure SIP Call Transfer and Call Forwarding on a VoIP Dial Peer, page 31](#)
- [Configure the SIP Call-Transfer and Call-Forwarding Session Target, page 33](#)
- [Configure SIP Refer and Notify Message Settings, page 36](#)

### Load the Tcl IVR Application on the Gateway

- Before you implement SIP support of blind or attended call-transfer and call-forwarding requests from a Cisco IOS gateway, you must load a custom Tcl IVR 2.0 or VXML script on the gateway. Write a Tcl IVR 2.0 script that implements Cisco IOS call-transfer and call-forwarding services. The Tcl IVR script is responsible for receiving the hookflash event, providing dial tone, matching against the dial plan, initiating the call transfer, and reestablishing the original call if the transfer attempt fails.



#### Note

For information on writing a Tcl IVR script, see "Tcl IVR API Version 2.0 Programmer's Guide".

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **call application voice** *application-name* *location*
4. **call application voice** *application-name* **language** *number* *language*
5. **call application voice** *application-name* **set-location** *language* *category* *location*
6. **exit**
7. **all application voice load** *application-name*

**DETAILED STEPS**

Command or Action	Purpose
<p><b>Step 1</b> <b>enable</b></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.</p>
<p><b>Step 2</b> <b>configure terminal</b></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <b>call application voice</b> <i>application-name</i> <i>location</i></p> <p><b>Example:</b></p> <pre>Router(config)# call application voice transfer_app flash:app_h450_transfer.tcl</pre>	<p>Loads the Tcl IVR script and specifies its application name. Arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <i>application-name</i> --Name used to reference the call application. This is a user-defined name and does not have to match the document name.</li> <li>• <i>location</i> --Location of the Tcl IVR file in URL format. For example, flash memory (flash:filename), TFTP (tftp://../filename) or HTTP server paths (http://../filename) are valid locations.</li> </ul>

Command or Action	Purpose
<p><b>Step 4</b> <code>call application voice</code> <i>application-name</i> <code>language</code> <i>number language</i></p> <p><b>Example:</b></p> <pre>Router(config)# call application voice transfer_app language 1 en</pre>	<p>(Optional) Sets the language for dynamic prompts used by the application. Arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <i>application-name</i> --Name of the Tcl IVR application to which the language parameters pass.</li> <li>• <i>number</i> --Number that identifies the language used by the audio files for the IVR application.</li> <li>• <i>language</i> --Language of the associated audio file. Valid values are as follows: <ul style="list-style-type: none"> <li>◦ <b>en</b>--English</li> <li>◦ <b>sp</b>--Spanish</li> <li>◦ <b>ch</b>--Mandarin</li> <li>◦ <b>aa</b>--All</li> </ul> </li> </ul>
<p><b>Step 5</b> <code>call application voice</code> <i>application-name set-location</i> <i>language category location</i></p> <p><b>Example:</b></p> <pre>Router(config)# call application voice transfer_app set-location en 0 flash:/ prompts</pre>	<p>(Optional) Defines the location and category of the audio files that are used by the application for dynamic prompts. Arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <i>application-name</i> --Name of the Tcl IVR application.</li> <li>• <i>language</i> --Language of the associated audio file. Valid values are as above.</li> <li>• <i>category</i> --Category group (0 to 4) for the audio files from this location. For example, audio files for the days and months could be category 1, audio files for units of currency could be category 2, and audio files for units of time (seconds, minutes, and hours) could be category 3. Range is from 0 to 4. The value 0 means all categories.</li> <li>• <i>location</i> --URL of the directory that contains the language audio files used by the application, without filenames. For example, flash memory (flash) or a directory on a server (TFTP, HTTP, or RTSP) are valid locations.</li> </ul>
<p><b>Step 6</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config)# exit</pre>	<p>Exits the current mode.</p>
<p><b>Step 7</b> <code>all application voice load</code> <i>application-name</i></p> <p><b>Example:</b></p> <pre>Router# call application voice load transfer.app</pre>	<p>(Optional) Reloads the Tcl script after it has been modified. The argument is as follows:</p> <ul style="list-style-type: none"> <li>• <i>application-name</i> --Name of the Tcl IVR application to reload.</li> </ul>

## Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer

**Note**

To handle all call-transfer and call-forwarding situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer and call forwarding for a POTS dial peer.

- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide*.

**Note**

To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:<http://www.cisco.com/web/psa/products/index.html>.

**Note**

In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **dial-peer voice tag pots**
4. **application application-name**
5. **destination-pattern** [+]*string* [T]
6. **port slot / port**
7. **exit**

**DETAILED STEPS**

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b>  Router> enable	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	<b>configure terminal</b>  <b>Example:</b>  Router# configure terminal	Enters global configuration mode.

Command or Action	Purpose
<p><b>Step 3</b> <code>dial-peer voice tag pots</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 25 pots</pre>	Enters dial-peer configuration mode and for the specified POTS dial peer.
<p><b>Step 4</b> <code>application application-name</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# application transfer_app</pre>	Enables a specific application on a dial peer. The argument is as follows:
<p><b>Step 5</b> <code>destination-pattern [+]<i>string</i> [T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination-pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <code>+</code> --(Optional) Character that indicates an E.164 standard number.</li> <li>• <code>string</code> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li>• <code>T</code> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable-length dial string.</li> </ul>
<p><b>Step 6</b> <code>port slot / port</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# port 1/1</pre>	<p>Associates a dial peer with a voice slot number and a specific local voice port through which incoming VoIP calls are received.</p> <p><b>Note</b> To find the correct port argument for your router, see the <i>Cisco IOS Voice Command Reference</i>.</p>
<p><b>Step 7</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# exit</pre>	Exits the current mode.

## Configure SIP Call Transfer and Call Forwarding on a VoIP Dial Peer



### Note

To handle all call-transfer and call-forwarding situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer and call forwarding for a VoIP dial peer.

- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide*.

**Note**

To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:<http://www.cisco.com/web/psa/products/index.html> .

**Note**

- RLT on CAS or analog (FXS) ports is necessary for initiating IP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **dial-peer voice tag voip**
4. **application application-name**
5. **destination-pattern** [+]string [T]
6. **session target ipv4:** destination-address
7. **session protocol sipv2**
8. **exit**

**DETAILED STEPS**

Command or Action	Purpose
<b>Step 1 enable</b>  <b>Example:</b> <pre>Router&gt; enable</pre>	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
<b>Step 2 configure terminal</b>  <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.



Command or Action	Purpose
<p><b>Step 3</b> <code>dial-peer voice tag voip</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 29 voip</pre>	Enters dial-peer configuration mode for the specified dial peer.
<p><b>Step 4</b> <code>application application-name</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# application transfer_app</pre>	Enables a specific application on a dial peer. The argument is as follows:
<p><b>Step 5</b> <code>destination-pattern [+]<i>string</i> [T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination-pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li>• <code>+</code> --(Optional) Character that indicates an E.164 standard number.</li> <li>• <i>string</i> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li>• <code>T</code> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable-length dial string.</li> </ul>
<p><b>Step 6</b> <code>session target ipv4: destination-address</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# session target ipv4:10.10.1.3</pre>	<p>Specifies a network-specific address for a dial peer. The argument is as follows:</p> <ul style="list-style-type: none"> <li>• <i>destination address</i> --IP address of the dial peer, in this format: <code>xxx.xxx.xxx.xxx</code></li> </ul>
<p><b>Step 7</b> <code>session protocol sipv2</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# session protocol sipv2</pre>	<p>Configures the VoIP dial peer to use IETF SIP. The keyword is as follows:</p> <ul style="list-style-type: none"> <li>• <b>sipv2</b> --Causes the VoIP dial peer to use IETF SIP. Use this keyword with the SIP option.</li> </ul>
<p><b>Step 8</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# exit</pre>	Exits the current mode.

## Configure the SIP Call-Transfer and Call-Forwarding Session Target

**Note**

To configure a SIP server as a session target, follow this task. Although configuring a SIP server as a session target is not required, it is useful if there is a Cisco SIP proxy server (CSPS) present in the network. With a CSPS, you can configure the SIP server option and have the interested dial peers use the CSPS by default.

- To determine the call-transfer destination on the originator, check if there is a matching dial peer:
  - If yes, check the session target for the dial peer. If the session target is a SIP server, configure the SIP server as described in the task below. If the session target is not a SIP server, the session target configured in the VoIP dial peer is used.
  - If no, a TEL URL is sent.
- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide* .

**Note**

To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:<http://www.cisco.com/web/psa/products/index.html> .

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **sip-server dns: *host-name***
5. **exit**
6. **dial-peer voice *tag* voip**
7. **destination-pattern *[+]*string[T]**
8. **session target sip-server**
9. **exit**

**DETAILED STEPS**

Command or Action	Purpose
<b>Step 1</b> <b>enable</b>  <b>Example:</b>  Router> enable	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.

Command or Action	Purpose
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p><b>Step 3</b> <code>sip-ua</code></p> <p><b>Example:</b></p> <pre>Router(config)# sip-ua</pre>	Enters SIP user-agent configuration mode.
<p><b>Step 4</b> <code>sip-server dns: host-name</code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# sip-server dns:example.sip.com</pre>	<p>Sets the global SIP server interface to a DNS hostname. The argument is as follows:</p> <ul style="list-style-type: none"> <li><code>host-name</code> --DNS hostname. If you do not specify a hostname, the default DNS defined by the <code>ip name-server</code> command is used.</li> </ul>
<p><b>Step 5</b> <code>exit</code></p> <p><b>Example:</b></p> <pre>Router(config-sip-ua)# exit</pre>	Exits the current mode.
<p><b>Step 6</b> <code>dial-peer voice tag voip</code></p> <p><b>Example:</b></p> <pre>Router(config)# dial-peer voice 29 voip</pre>	Enters dial-peer configuration mode for the specified dial peer.
<p><b>Step 7</b> <code>destination-pattern [+]<i>string</i>[T]</code></p> <p><b>Example:</b></p> <pre>Router(config-dial-peer)# destination- pattern 7777</pre>	<p>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</p> <ul style="list-style-type: none"> <li><code>+</code> --(Optional) Character that indicates an E.164 standard number.</li> <li><code>string</code> --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</li> <li><code>T</code> --(Optional) Control character indicating that the <b>destination-pattern</b> value is a variable-length dial string.</li> </ul>

Command or Action	Purpose
<b>Step 8</b> <code>session target sip-server</code>  <b>Example:</b>  <pre>Router(config-dial-peer)# session target sip-server</pre>	<b>Instruct the dial-peer session target to use the global SIP server .</b> Doing so saves you from having to repeatedly enter the SIP server interface address for each dial peer.
<b>Step 9</b> <code>exit</code>  <b>Example:</b>  <pre>Router(config-dial-peer)# exit</pre>	Exits the current mode.

## Configure SIP Refer and Notify Message Settings

To configure SIP Refer and Notify message settings, perform the following steps.



### Note

The Refer request is initiated by the originating gateway and signals the start of call transfer. Once the outcome of the SIP Refer transaction is known, the recipient of the Refer request notifies the originating gateway of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify method.

- Custom scripting is necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.
- Configure the dial peers for correct functioning of the Refer method.



### Note

Dial-peer configuration steps are in "Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer".



**Note**

- Only RLT on CAS or analog (FXS) ports is supported with SIP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS and CAS phones are not supported.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.



**Note**

Custom scripting is necessary for ephones to initiate call forwarding. The standard configurations in this document work only when an ephone is the recipient or final-recipient.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers refer** *milliseconds*
5. **retry refer** *number*
6. **timers notify** *milliseconds*
7. **retry notify** *number*
8. **exit**

**DETAILED STEPS**

Command or Action	Purpose
<p><b>Step 1 enable</b></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.</p>
<p><b>Step 2 configure terminal</b></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>

Command or Action	Purpose
<b>Step 3 sip-ua</b>  <b>Example:</b> <pre>Router(config)# sip-ua</pre>	Enters SIP user-agent configuration mode.
<b>Step 4 timers refer <i>milliseconds</i></b>  <b>Example:</b> <pre>Router(config-sip-ua)# timers refer 500</pre>	Sets the amount time that the user agent waits before retransmitting the Refer request. The argument is as follows: <ul style="list-style-type: none"> <li>• <i>milliseconds</i> --Time, in ms. Range: 100 to 1000. Default: 500.</li> </ul>
<b>Step 5 retry refer <i>number</i></b>  <b>Example:</b> <pre>Router(config-sip-ua)# retry refer 10</pre>	Sets the number of times that the Refer request is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows: <ul style="list-style-type: none"> <li>• <i>number</i> --Number of Notify message retries. Range: 1 to 10. Default: 10.</li> </ul>
<b>Step 6 timers notify <i>milliseconds</i></b>  <b>Example:</b> <pre>Router(config-sip-ua)# timers notify 500</pre>	Sets the amount time that the user agent waits before retransmitting the Notify message. The argument is as follows: <ul style="list-style-type: none"> <li>• <i>milliseconds</i> --Time, in ms. Range: 100 to 1000. Default: 500.</li> </ul>
<b>Step 7 retry notify <i>number</i></b>  <b>Example:</b> <pre>Router(config-sip-ua)# retry notify 10</pre>	Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows: <ul style="list-style-type: none"> <li>• <i>number</i> --Number of notify message retries. Range: 1 to 10.</li> </ul>
<b>Step 8 exit</b>  <b>Example:</b> <pre>Router(config-sip-ua)# exit</pre>	Exits the current mode.

## Verifying SIP Call Transfer

To verify SIP configurations, perform the following steps as appropriate (commands are listed in alphabetical order).

**SUMMARY STEPS**

1. **show dial-peer voice**
2. **show ephone**
3. **show ephone-dn**
4. **show running-configuration**
5. **show sip-ua retry**
6. **show sip-ua statistics**
7. **show sip-ua timers**
8. **show telephony-service ephone-dn**
9. **show telephony-service voice-port**
10. **show voice port**

**DETAILED STEPS**

- 
- Step 1**     **show dial-peer voice**  
Use this command to display configuration information about voice dial peers. Use with the **summary** keyword to display a summary of all voice dial peers.
- Step 2**     **show ephone**  
Use this command to display IP-phone output. Use with the **summary** keyword to display a summary of all IP phones.
- Step 3**     **show ephone-dn**  
Use this command to display the IP-phone destination number. Use with the **summary** keyword to display a summary of all IP-phone destination numbers.
- Step 4**     **show running-configuration**  
Use this command to verify your configuration.
- Step 5**     **show sip-ua retry**  
Use this command to display SIP retry statistics including Notify responses.

**Example:**

```
Router# show sip-ua retry
SIP UA Retry Values
invite retry count = 6 response retry count = 1
bye retry count = 1 cancel retry count = 1
prack retry count = 10 comet retry count = 10
reliable lxx count = 6 notify retry count = 10
```

- Step 6**     **show sip-ua statistics**  
Use this command to display response, traffic, and retry statistics for the SIP user agent.

The following applies to the example below.

Field	Meaning
OkNotify1/0	Successful response to the Notify request.
202Accepted 0/1	Successful response to the Refer request.

Field	Meaning
Notify 0/1	Status.
Refer 1/0	Status.
Notify 0/1	No Notify requests were received from the gateway. One request was sent.
Refer 1/0	One request was received. No requests were sent.
Notify 0 under Retry Statistics	The Notify request was not retransmitted.

**Example:**

```

Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
  Informational:
    Trying 0/0, Ringing 0/0,
    Forwarded 0/0, Queued 0/0,
    SessionProgress 0/0
  Success:
    OkInvite 0/0, OkBye 0/0,
    OkCancel 0/0, OkOptions 0/0,
    OkPrack 0/0, OkPreconditionMet 0/0,
    OKSubscribe 0/0, OkNotify 0/0,
    202Accepted 0/0
  Redirection (Inbound only):
    MultipleChoice 0, MovedPermanently 0,
    MovedTemporarily 0, UseProxy 0,
    AlternateService 0
  Client Error:
    BadRequest 0/0, Unauthorized 0/0,
    PaymentRequired 0/0, Forbidden 0/0,
    NotFound 0/0, MethodNotAllowed 0/0,
    NotAcceptable 0/0, ProxyAuthReqd 0/0,
    ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
  ReqEntityTooLarge 0/0, ReqURITooLarge 0/0,
  UnsupportedMediaType 0/0, BadExtension 0/0,
  TempNotAvailable 0/0, CallLegNonExistent 0/0,
  LoopDetected 0/0, TooManyHops 0/0,
  AddrIncomplete 0/0, Ambiguous 0/0,
  BusyHere 0/0, RequestCancel 0/0
  NotAcceptableMedia 0/0, BadEvent 0/0
  Server Error:
    InternalError 0/0, NotImplemented 0/0,
    BadGateway 0/0, ServiceUnavail 0/0,
    GatewayTimeout 0/0, BadSipVer 0/0,
    PreCondFailure 0/0
  Global Failure:
    BusyEverywhere 0/0, Decline 0/0,
    NotExistAnywhere 0/0, NotAcceptable 0/0
  Miscellaneous counters:
    RedirectResponseMappedToClientError 1,
SIP Total Traffic Statistics (Inbound/Outbound)
  Invite 0/0, Ack 0/0, Bye 0/0,
  Cancel 0/0, Options 0/0,
  Prack 0/0, Comet 0/0,
  Subscribe 0/0, Notify 0/0,
  Refer 0/0
  Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliablelxx 0, Notify 0
SDP application statistics:
  Parses: 0, Builds 0

```



```
Invalid token order: 0, Invalid param: 0
Not SDP desc: 0, No resource: 0
```

**Tip** To reset counters for the `show sip-ua statistics` command, use the `clear sip-ua statistics` command.

**Step 7** `show sip-ua timers`

Use this command to display the current settings for SIP user-agent timers, including Notify responses.

**Example:**

```
Router# show sip-ua timers
SIP UA Timer Values (milliseconds)
trying 500, expires 150000, connect 500, disconnect 500
comet 500, prack 500, rellxx 500, notify 500
```

**Step 8** `show telephony-service ephone-dn`

Use this command to display the Cisco-IP-phone destination number of the Cisco IOS telephony-service router.

**Step 9** `show telephony-service voice-port`

Use this command to display the virtual voice-port configuration.

**Step 10** `show voice port`

Use this command to display configuration information about a specific voice port.

## Troubleshooting Tips

For general troubleshooting tips and a list of important **debug** commands, see the “General Troubleshooting Tips” section of the “Basic SIP Configuration” document.

## Configuration Examples for SIP Call-Transfer Features

- [SIP Call Transfer Using the Refer Method Examples, page 41](#)
- [SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Examples, page 42](#)

## SIP Call Transfer Using the Refer Method Examples



**Note**

Note that the `application session` command is set on all involved gateway dial peers. You must set the correct Cisco IOS session for call transfer.

```
Router# show running-config
Building configuration...
Current configuration : 4192 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
```

```

no service password-encryption
service internal
service udp-small-servers
!
voice-card 2
!
ip subnet-zero
!
controller T1 2/0
framing esf
linecode b8zs
ds0-group 0 timeslots 1-24 type e&m-wink-start
!
interface FastEthernet3/0
ip address 172.18.200.36 255.255.255.0
speed 10
half-duplex
no shut
ip rsvp bandwidth 7500 7500
!
voice-port 2/0:0
timing hookflash-in 1500
!
dial-peer voice 3660110 voip
application session
incoming called-number 3660110
destination-pattern 3660110
session protocol sipv2
session target ipv4:172.18.200.24
codec g711ulaw
!
dial-peer voice 3640110 pots
application session
destination-pattern 3640110
direct-inward-dial
port 2/0:0
!
sip-ua
retry bye 1
retry refer 3
timers notify 400
timers refer 567
no oli
sip-server ipv4:172.18.200.21
!
line con 0
line aux 0
line vty 0 4
login
!
end

```

## SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Examples

This section provides an end-to-end call-transfer configuration example.



### Note

---

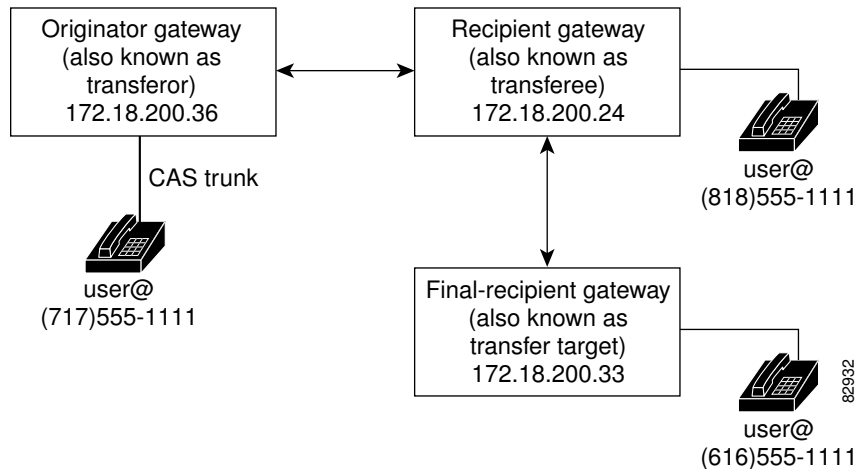
IP addresses and hostnames in examples are fictitious.

---

## Blind Call Transfer

The figure below shows the relationship of the gateways in the blind call transfer.

**Figure 10** *Blind Call Transfer*



The following scenario is an example of a blind call transfer.

- 1 User at (818) 555-0111 calls user at (717) 555-0111, and they are in a conversation.
- 2 User at (717) 555-0111 decides to transfer user at (818) 555-0111 to user at (616) 555-0111.

Transfer takes place by the user at (717) 555-0111 going on-hook over the CAS trunk and dialing (616) 555-0111.

- 1 Call transfer is initiated from the originating gateway to the recipient gateway, and the originator releases the CAS trunk to (717) 555-0111.
- 2 Recipient gateway releases the call leg to the originator and initiates a new call to the final-recipient-- user at (616) 555-0111.
- 3 Call transfer is complete, and user at (818) 555-0111 and user at (616) 555-0111 are in a conversation.

## Originating Gateway

The following example shows a configuration of the originating gateway.

```

Router# show running-config
Building configuration...
Current configuration : 4192 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
voice-card 2
!
ip subnet-zero
!
controller T1 2/0
framing esf
  
```

```

linecode b8zs
ds0-group 0 timeslots 1-24 type e&m-wink-start
!
interface FastEthernet3/0
ip address 172.18.200.36 255.255.255.0
speed 10
half-duplex
no shut
ip rsvp bandwidth 7500 7500
!
voice-port 2/0:0
timing hookflash-in 1500
!
call application voice sample_RLT tftp://sample_RLT.tcl
call application voice sample_RLT uid-len 4
call application voice sample_RLT language 1 en
call application voice sample_RLT set-location en 0 tftp://prompts/en/
!
dial-peer voice 2 voip
application sample_rlt
destination-pattern 8183821111
session protocol sipv2
session target ipv4:172.18.200.24
codec g711ulaw
!
dial-peer voice 3 pots
destination-pattern 7173721111
direct-inward-dial
port 2/0:0
prefix 7173721111
!
dial-peer voice 3621111 voip
application sample_rlt
destination-pattern 6163621111
session protocol sipv2
session target sip-server
codec g711ulaw
!
sip-ua
retry bye 1
retry refer 3
timers notify 400
timers refer 567
no oli
sip-server ipv4:172.18.200.21
!
line con 0
line aux 0
line vty 0 4
login
!
end

```

### Recipient Gateway

The following example shows a configuration of the recipient gateway.

```

Router# show running-config
Building configuration...
Current configuration : 2791 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
interface FastEthernet2/0

```

```

ip address 172.18.200.24 255.255.255.0
duplex auto
no shut
speed 10
ip rsvp bandwidth 7500 7500
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 8183821111
port 1/1/1
!
dial-peer voice 3 voip
application session
destination-pattern 7173721111
session protocol sipv2
session target ipv4:172.18.200.36
codec g711ulaw
!
dial-peer voice 4 voip
application session
destination-pattern 6163621111
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
sip-ua
!
line con 0
line aux 0
line vty 0 4
login
!
end

```

### Final-Recipient

The following example shows a configuration of the final-recipient gateway.

```

Router# show running-config
!
version 12.2
no parser cache
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
no logging buffered
!
clock timezone GMT 0
aaa new-model
!
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
!
ip domain name example.com
ip dhcp smart-relay
!
!
voice class codec 1
codec preference 2 g711alaw

```

```

codec preference 3 g711ulaw
codec preference 5 g726r16
codec preference 6 g726r24
codec preference 7 g726r32
codec preference 8 g723ar53
codec preference 9 g723ar63
codec preference 10 g729r8
!
interface Ethernet0/0
ip address 172.18.200.33 255.255.255.0
no shut
half-duplex
ip rsvp bandwidth 7500 7500
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dial-peer voice 1 pots
application session
destination-pattern 6163621111
port 1/1/1
!
ip classless
no ip http server
ip pim bidir-enable
!
gateway
!
sip-ua
!
rtr responder
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
password password1
line vty 5 15
!
end

```

## Additional References

### General SIP References

- “Overview of SIP” --Describes underlying SIP technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance.

### References Mentioned in This Chapter (listed alphabetically)

- "Call Transfer Capabilities Using the Refer Method".
- *Cisco IOS Dial Technologies Configuration Guide*.

**Note**

---

To locate a release-specific configuration guide for your Cisco IOS software release, select the Cisco IOS and NX-OS Software category at the following Product Support page and navigate accordingly:<http://www.cisco.com/web/psa/products/index.html>.

---

- *Cisco IOS Voice Command Reference*.
- "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms".
- "CDR Accounting for Cisco IOS Voice Gateways" guide.
- "Tcl IVR API Version 2.0 Programmer's Guide".

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2012 Cisco Systems, Inc. All rights reserved.