



# Default Session Application Enhancements

---

The Default Session Application Enhancements feature provides support for Open Settlement Protocol (OSP), call transfer, and call forwarding. These features previously required use of a Tool Command Language interactive voice response (TCL IVR) application, such as `app_session.tcl`.

## Feature Specifications for Default Session Application Enhancements

---

### Feature History

Release	Modification
12.2(15)ZJ	This feature was introduced.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T.

---

### Supported Platforms

Cisco 1751-V, Cisco 1760, Cisco 1760-V, Cisco 2610XM, Cisco 2611XM, Cisco 2620XM, Cisco 2621XM, Cisco 2650, Cisco 2650XM, Cisco 2651XM-V, Cisco 2691, Cisco 3640, Cisco 3640A, Cisco 3660, Cisco 3725, Cisco 3745, and Cisco IAD2430 series.

**Note** The Cisco IAD2420 series is not supported in Cisco IOS Release 12.2(15)ZJ.

---

## Finding Support Information for Platforms and Cisco IOS Software Images

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

## Contents

- [Prerequisites for Default Session Application Enhancements, page 2](#)
- [Information About Default Session Application Enhancements, page 2](#)
- [How to Configure Default Session Application Enhancements, page 6](#)
- [Configuring a Global Application Example, page 10](#)
- [Additional References, page 11](#)
- [Command Reference, page 13](#)



---

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

Copyright © 2003 Cisco Systems, Inc. All rights reserved.

- [Glossary, page 25](#)

## Prerequisites for Default Session Application Enhancements

You must install Cisco IOS Release 12.3(4)T before using the Default Session Application Enhancements feature.

## Information About Default Session Application Enhancements

To configure the Default Session Application Enhancements feature, you must understand the following concepts:

- [New Version of the Default Session Application, page 2](#)
- [Global Default Application, page 2](#)
- [OSP, page 2](#)
- [Call Transfer and Call Forwarding, page 3](#)

## New Version of the Default Session Application



Note

---

The generic term *session application* refers to any application that is invoked when a call is placed through a dial peer. This generic term should not be confused with the specific TCL IVR 2.0 application named `session` (`app_session.tcl`), which is an example of a session application.

---

The default session application is typically used for one-stage or two-stage voice calls that do not require IVR treatment or a custom call flow. The new version of the default session application provides the same functionality as the old version. In addition, it provides call forward and call transfer capabilities for platforms that do not support the use of TCL IVR applications.

## Global Default Application

The new **`call application global`** command allows you to configure a default application for all dial peers with a single command. Prior to this feature, you could configure applications at the dial peer level only.

## OSP

The new version of the default session application provides support for OSP that was previously available only when using a TCL IVR application, such as `app_session.tcl`. With the new version of the default session application, you need not configure a TCL IVR 2.0 application on the plain old telephone service (POTS) dial peer in order to use OSP.

OSP is an ETSI standard, and it is the standard Cisco settlement protocol. OSP is a protocol based on Secure Socket Layer (SSL), which authenticates an IP session and authorizes the usage of network resources. OSP uses a combination of HTTP, Extensible Markup Language (XML), and SSL 3.0 to perform transfer pricing and authorization, and to indicate usage information.

The OSP implementation allows two gateways to use OSP to authorize and bill Public Switched Telephone Network (PSTN) calls routed over an IP network. With the settlement feature, calls always originate in the PSTN network, are authorized on an incoming gateway, and carry secure token information to an outgoing gateway.

**Note**

For more information on OSP, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2, “[Configuring Settlement Applications](#)” chapter.

## Call Transfer and Call Forwarding

The new version of the default session application provides support for H.450 and Session Initiation Protocol (SIP) call transfer and call forwarding. These features were previously available only when using a TCL IVR 2.0 application, such as `app_session.tcl`. With the new version of the default session application, you need not configure a TCL IVR 2.0 application on each incoming dial peer that will be involved in call transfer or call forwarding.

### H.450.2 Call Transfer Supplementary Service for H.323

H450.2 call transfer is a supplementary service that allows an endpoint to redirect an answered call to another endpoint. The ITU-T H.450.2 specification defines two variants of call transfer:

- **Transfer without consultation**—The transferring endpoint sends a transfer request to the transferred endpoint. This request includes the transferred-to number. The transferred endpoint initiates a call directly to the transferred-to endpoint without the involvement of the transferring endpoint.

Prior to Cisco IOS Release 12.2(11)YT, Cisco gateways supported H.450.2 call transfer as the transferred and transferred-to endpoint for call transfers without consultation.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways may also act as the transferring, transferred and transferred-to endpoint for call transfer without consultation when configured with a TCL IVR application that provides this functionality. One example is `app-h450-transfer.2.0.0.1.zip` (which can be found at <http://www.cisco.com/cgi-bin/tablebuild.pl/ip-iostsp>). This TCL IVR application allows the gateway to act as an H450.2 transferring endpoint for analog Foreign Exchange Station (FXS), T1 channel associated signaling (CAS), and Cisco IOS Telephony Service (ITS) IP phones.

- **Transfer with consultation**—The transferring endpoint places a consultation call to the transferred-to endpoint. During or after the consultation call, the transferring endpoint sends a transfer request to the transferred endpoint. This request includes the transfer-to number. The transferred endpoint initiates a call directly to the transferred-to endpoint without the involvement of the transferring endpoint.

Prior to Cisco IOS Release 12.2(11)YT, Cisco gateways did not support H.450.2 call transfer with consultation.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways may act as the transferring, transferred, and transferred-to endpoint for call transfer with consultation when configured with a TCL IVR application that provides this functionality. One example is `app-h450-transfer.2.0.0.1.zip` (which can be found at <http://www.cisco.com/cgi-bin/tablebuild.pl/ip-iostsp>). This TCL IVR application allows the gateway to act as an H450.2 transferring endpoint for analog FXS and Cisco ITS IP phones.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways support H.450 call transfer using gatekeeper routed signaling.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways support H.450 gatekeeper controlled (gatekeeper initiated) call transfer. Note that this functionality has not been interoperability tested with Cisco or any third-party gatekeepers.

## SIP Call Transfer

SIP supports transfer without consultation (blind transfer) and transfer with consultation (attended transfer) 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. This is different from a consultative 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.

Release Line Trunking (RLT) functionality can be used to initiate blind transfers on Cisco IOS SIP gateways. The RLT functionality is used by SIP gateways to allow SIP call transfers to be triggered by CAS trunk signaling, which a 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. 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.

Cisco IOS SIP gateways can 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 number of the final recipient. 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.

## H.450.3 Call Diversion Supplementary Service for H.323

H450.3 call diversion is a supplementary service used during call establishment that allows an H.323 endpoint to divert the unanswered call to another H.323 endpoint. Four types of call diversion are specified in H450.3:

- Call forwarding unconditional (CFU)—Allows all incoming calls to a particular destination number to be diverted to a different destination.

Prior to Cisco IOS Release 12.2(11)YT, Cisco gateways supported CFU when acting as the diverted or diverted-to endpoint.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways may act as the diverting endpoint for Cisco ITS IP phones when configured with a TCL IVR application that provides this functionality. One example is `app-h450-transfer.2.0.0.1.zip` (which can be found at <http://www.cisco.com/cgi-bin/tablebuild.pl/ip-iostsp>).

- Call forwarding busy (CFB)—Allows incoming calls to a particular destination number to be diverted to a different destination if the original destination is busy.

Prior to Cisco IOS Release 12.2(11)YT, Cisco gateways supported CFB when acting as the diverted or diverted-to endpoint.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways may act as the diverting endpoint for Cisco ITS IP phones when configured with a TCL IVR application that provides this functionality. One example is `app-h450-transfer.2.0.0.1.zip` (which can be found at <http://www.cisco.com/cgi-bin/tablebuild.pl/ip-iostsp>).

- Call forwarding no reply (CFNR)—Allows incoming calls to a particular destination number to be diverted to a different destination if the original destination does not answer within a specified amount of time.

Prior to Cisco IOS Release 12.2(11)YT, Cisco gateways supported CFNR when acting as the diverted or diverted-to endpoint.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways may act as the diverting endpoint for Cisco ITS IP phones when configured with a TCL IVR application that provides this functionality. One example is `app-h450-transfer.2.0.0.1.zip` (which can be found at <http://www.cisco.com/cgi-bin/tablebuild.pl/ip-iostsp>).

- Call deflection (CD)—Allows incoming calls to a particular destination number to be diverted to a different destination. Call deflection differs from call forwarding in that the diverted-to destination is determined per call by the called user for call deflection.

Cisco gateways support H.450.3 call deflection as the diverting, diverted, and diverted-to endpoint. The Cisco gateway can act as the diverting endpoint upon receiving a QSIG reroute invoke request FACILITY message from the destination during call establishment. The deflecting gateway then uses the procedures outlined in the H.450.3 call deflection standard to transfer the call to another endpoint.



---

**Note** Call deflection cannot be invoked by using any other signaling type or switch type.

The initiation of call deflection using QSIG reroute invoke is valid only on calls that arrived as H.323 calls at the deflecting gateway. In other words, for calls that arrive at the gateway through a telephony interface (such as a hairpin call) or by using a non-H.323 IP protocol, the QSIG reroute invoke is ignored.

---

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways support call diversion using gatekeeper routed signaling.

Starting with Cisco IOS Release 12.2(11)YT, Cisco gateways support H.450 gatekeeper controlled (gatekeeper initiated) call transfer. Note that this functionality has not been interoperability tested with Cisco or any third-party gatekeepers.

## SIP Call Forwarding

SIP call forwarding is supported only on e-phones—IP phones that are registered with a Cisco IOS voice gateway operating in IOS Telephony Service (ITS) mode. FXS, Foreign Exchange Office (FXO), T1, E1, and CAS phones are not supported.

With e-phones, four different types of SIP call forwarding are 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. The Contact header 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 Voice over IP (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**

In Cisco IOS Release 12.2(11)YT, when SIP with e-phones is used, dual tone multifrequency (DTMF) is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for e-phones to initiate call forwarding.

**Note**

For more information on call transfer and call forwarding, refer to the Voice Configuration Library, Cisco IOS Release 12.3, “[Configuring H.323 Applications](#)” chapter and “[Configuring SIP for VoIP](#)” chapter, and [SIP Call Transfer and Call Forwarding Supplementary Services](#) documentation.

## How to Configure Default Session Application Enhancements

The default session application enhancements are in effect by default. The new version of the default session application is used automatically for incoming calls unless a different application is configured for specific dial peers using the **application** command or is globally configured for all inbound dial peers using the **call application global** command.

This section contains the following procedures.

- [Configuring a Global Application, page 6](#) (optional)
- [Verifying the Default Session Application, page 7](#) (optional)
- [Troubleshooting the Default Session Application, page 8](#) (optional)

### Configuring a Global Application

Perform this task to configure the application to use for incoming calls whose incoming dial peer does not have an explicit application configured.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call application global** *application-name*
4. **exit**
5. **show running-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"><li>Enter your password if prompted.</li></ul>
Step 2	<code>configure terminal</code>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<code>call application global application-name</code>  <b>Example:</b> Router(config)# call application global clid_authen_collect	Configures an application to be executed by every inbound dial peer that does not have a specific application configured. <ul style="list-style-type: none"><li>The <b>call application global</b> command can be overridden by configuring an application on the incoming dial peer using the <b>application</b> command.</li><li>The previous version of the default session application has been renamed default.c.old and can still be configured for specific dial peers through the <b>application</b> command or globally configured for all inbound dial peers through the <b>call application global</b> command.</li></ul>
Step 4	<code>exit</code>  <b>Example:</b> Router(config)# exit	Exits global configuration mode.
Step 5	<code>show running-config</code>  <b>Example:</b> Router# show running-config   begin application global	(Optional) Displays the contents of the currently running configuration file. <ul style="list-style-type: none"><li>Verifies that the new default session application has been configured for all inbound dial peers.</li></ul>

## Verifying the Default Session Application

Perform this task to verify that the new version of the default session application is executing.

## SUMMARY STEPS

1. `enable`
2. `debug voip application session`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> <code>enable</code>	Enables higher privilege levels, such as privileged EXEC mode.  • Enter your password if prompted.
Step 2	<code>debug voip application session</code>  <b>Example:</b> Router# <code>debug voip application session</code>	(Optional) Displays session application debug messages.

## Troubleshooting the Default Session Application

The Default Session Application Enhancements feature introduces a new EXEC mode command to enable diagnostic output concerning various events relating to the operation of the default session application to be displayed on a console. The `debug voip application` command is intended only for troubleshooting purposes because the volume of output generated by the software can result in severe performance degradation on the router. Perform this task to minimize the impact of using the `debug voip application` command.

## SUMMARY STEPS

1. Attach a console directly to a router running the Cisco IOS Release 12.2(15)ZJ or a later release.
2. `enable`
3. `configure terminal`
4. `no logging console`
5. Use Telnet to access a router port and repeat Steps 2 and 3.
6. `terminal monitor`
7. `end`
8. `debug voip application`
9. `configure terminal`
10. `no terminal monitor`
11. `end`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	Attach a console directly to a router running the Cisco IOS Release 12.2(15)ZJ or a later release.	—
Step 2	<code>enable</code>  <b>Example:</b> Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 3	<code>configure terminal</code>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 4	<code>no logging console</code>  <b>Example:</b> Router(config)# no logging console	Disables all logging to the console terminal. <ul style="list-style-type: none"> <li>To reenable logging to the console, use the <b>logging console</b> command in global configuration mode.</li> </ul>
Step 5	Use Telnet to access a router port and repeat Steps 2 and 3.	Enters global configuration mode in a recursive Telnet session, which allows the output to be redirected away from the console port.
Step 6	<code>terminal monitor</code>  <b>Example:</b> Router(config)# terminal monitor	Enables logging output on the virtual terminal.
Step 7	<code>end</code>  <b>Example:</b> Router(config)# end	Exits to privileged EXEC mode.
Step 8	<code>debug voip application</code>  <b>Example:</b> Router# debug voip application all	Display debug messages for Application Framework Session application interactions. <ul style="list-style-type: none"> <li>Use the <b>all</b> or <b>session</b> option of the <b>debug voip application</b> command to display all application library debugs or session application debugs.</li> </ul>
Step 9	<code>configure terminal</code>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 10	<code>no terminal monitor</code>  <b>Example:</b> Router(config)# no terminal monitor	Disables logging on the virtual terminal.
Step 11	<code>end</code>  <b>Example:</b> Router(config)# end	Exits to privileged EXEC mode.

# Configuration Examples for Default Session Application Enhancements

This section provides configuration examples to match the identified configuration tasks in the previous section:

- [Configuring a Global Application Example, page 10](#)
- [Verifying the Default Session Application Example, page 10](#)

## Configuring a Global Application Example

In the following example, `clid_authen_collect` is configured as the global application for all inbound dial peers that do not have a specific application configured. The default session application is configured on dial peer 111. This dial peer configuration overrides the global application configuration.

```
call application global clid_authen_collect

dial-peer voice 110 pots
  port 1:D

dial-peer voice 111 pots
  application default
  port 0:D
```

When a call comes into dial peer 110, the `clid_authen_collect` application handles the call because dial peer 110 has no application configured, and the `clid_authen_collect` application is the globally configured application.

When a call comes into dial peer 111, the new version of the default session application handles the call because it is configured on the dial peer, and the dial peer application configuration overrides the globally configured application.

## Verifying the Default Session Application Example

In the following example, the output is displayed for each command in the task.

Sample Output for the debug voip application session Command

In the following example, the function names that begin with `afs` indicate that the default session application is executing:

```
Router# debug voip application session
applib session debugging is on
*Apr  4 23:57:08.054://-1//APPL:HN04B2BC78:LG35:/AFS_CALLSETUPIND:Calling #(4153788854),
Called #(52984), peer_tag(1)
*Apr  4 23:57:08.054://-1//APPL:HN04B2BC78:LG35:/afsSetupCall:Called #(52984)
*Apr  4
23:57:08.058://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLINIT) -- (CC_EV_CALL_SETUP_IND) -
->> (CONTACTINGDEST)
*Apr  4 23:57:08.466://-1//APPL:HN04B2BC78:LG36:/AFS_ContactingDest_ALERT:
*Apr  4 23:57:08.470://-1//APPL:HN04B2BC78:LG36:/AFS_ContactingDest_ALERT:inID(35),
outID(36), outbnd peer_tag(6), prog_ind(8)
*Apr  4
23:57:08.470://-1//APPL:HN04B2BC78:LG36:/afsMsgHandler:(CONTACTINGDEST) -- (CC_EV_CALL_ALERT)
)-->> (CONFINGALERT)
```

```

*Apr 4 23:57:08.470://-1//APPL:HN04B2BC78:CN11:/AFS_ConfingAlert_CREATEDONE:
{HAN[AFS_HAND][NULL ]( LEG[35 ] [AFS_HAND][LEG_INCALERTING(4)] [Cause(0)] [UC=1
] HAN[CS_HAND][AFS_HAND][FALSE] [UC=1 ] LEG[36 ] [CS_HAND
] [LEG_OUTINIT(6)] [Cause(0)] [UC=1 ] CON[11 ] [AFS_HAND][CONNECTION_CONFED(2)] [UC=1 ]}
*Apr 4
23:57:08.470://-1//APPL:HN04B2BC78:CN11:/afsMsgHandler:(CONFINGALERT)--(CC_EV_CONF_CREATE_
DONE)-->(CONFEDALERT)
*Apr 4
23:57:08.478://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CONFEDALERT)--(CC_EV_VOICE_MODE_DO
NE)-->(CONFEDALERT)
*Apr 4 23:57:24.162://-1//APPL:HN04B2BC78:HN04B2BC78:/AFS_ConfedAlert_SETUPDONE:
*Apr 4 23:57:24.162://-1//APPL:HN04B2BC78:HN04B2BC78:/afsAppHandlerCleanup:CS_HAND
*Apr 4
23:57:24.162://-1//APPL:HN04B2BC78:/afsMsgHandler:(CONFEDALERT)--(APP_EV_CALLSETUP_DONE)--
->(CALLACTIVE)
*Apr 4
23:57:24.182://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLACTIVE)--(CC_EV_VOICE_MODE_DON
E)-->(CALLACTIVE)
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:LG35:/AFS_DISCONNECT:
*Apr 4
23:57:34.838://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLACTIVE)--(CC_EV_CALL_DISCONNEC
TED)-->(CALLDISCONNECT)
*Apr 4 23:57:34.838://-1//APPL:/afsCallProcess:[HANDLERDONE_EVENT_END]
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:/afsHNDCleanup:Terminate TRUE Terminated
FALSE{HAN[AFS_HAND][NULL ]( LEG[35 ] [AFS_HAND][LEG_INCCONNECTED(5)] [Cause(16)] [UC=1
] LEG[36 ] [AFS_HAND][LEG_OUTCONNECTED(8)] [Cause(0)] [UC=1 ] CON[11
] [AFS_HAND][CONNECTION_CONFED(2)] [UC=1 ]}
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:CN11:/afsMsgHandler:(CC_EV_CONF_DESTROY_DONE)
*Apr 4
23:57:34.854://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CC_EV_CALL_DISCONNECT_DONE)
*Apr 4
23:57:34.862://-1//APPL:HN04B2BC78:LG36:/afsMsgHandler:(CC_EV_CALL_DISCONNECT_DONE)
*Apr 4 23:57:34.862://-1//APPL:/afsCallProcess:[HANDLERDONE_EVENT_END]
*Apr 4 23:57:34.862://-1//APPL:HN04B2BC78:/afsHNDCleanup:Terminate TRUE Terminated
TRUE{HAN[AFS_HAND][NULL ]( )}
*Apr 4 23:57:34.862://-1//APPL:HN04B32530:/afsFreeHND:Hndlr returned to the free queue

```

## Additional References

The following sections provide additional references related to the Default Session Application Enhancements feature:

- [Related Documents, page 12](#)
- [Standards, page 12](#)
- [MIBs, page 12](#)
- [RFCs, page 12](#)
- [Technical Assistance, page 13](#)

## Related Documents

Related Topic	Document Title
Cisco H.323 and SIP functionality	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Voice, Video, and Fax Configuration Guide</a>, Release 12.2</li> <li>• <a href="#">Cisco IOS Voice, Video, and Fax Command Reference</a>, Release 12.2 T</li> <li>• <a href="#">Call Transfer Capabilities Using the Refer Method</a>, Release 12.2(8)T</li> <li>• <a href="#">SIP Call Transfer and Call Forwarding Supplementary Services</a>, Release 12.2(11)YT</li> </ul>
Cisco VoiceXML	<a href="#">Cisco IOS TCL and VoiceXML Application Guide</a>
Cisco TCL IVR API	<a href="#">TCL IVR API Version 2.0 Programmer's Guide</a>

## Standards

Standards <sup>1</sup>	Title
ITU-T H.450.2	<i>Call transfer supplementary service for H.323</i>
ITU-T H.450.3	<i>Call diversion supplementary service for H.323</i>

1. Not all supported standards are listed.

## MIBs

MIBs <sup>1</sup>	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:  <a href="http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a>

1. Not all supported MIBs are listed.

## RFCs

RFCs <sup>1</sup>	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

1. Not all supported RFCs are listed.

## Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	<a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a>

## Command Reference

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

### New Commands

- [call application global](#)
- [call application voice default disc-prog-ind-at-connect](#)
- [debug voip application](#)

### Modified Commands

- [application](#)

# application

To enable a specific interactive voice response (IVR) application on a dial peer, use the **application** command in dial peer voice configuration mode. To remove the application from the dial peer, use the **no** form of this command.

**application** *application-name* [**out-bound**]

**no application** *application-name* [**out-bound**]

## Syntax Description

<i>application-name</i>	Character string that indicates the name of the predefined application you want to enable on the dial peer. For H.323 networks, the application is defined by a Tool Command Language interactive voice response (TCL IVR) filename and location. Incoming calls using plain old telephone service (POTS) dial peers and outgoing calls using Multimedia Mail over IP (MMoIP) dial peers are handed off to this application. For Media Gateway Control Protocol (MGCP) or Simple Gateway Control Protocol (SGCP) networks, see the “Usage Guidelines” section for valid application names.
<b>out-bound</b>	(Optional) The named application will handle the MMoIP dial peer in the outgoing mode.

## Defaults

No default behavior or values.

## Command Modes

Dial peer voice configuration

## Command History

Release	Modification
11.3(6)NA2	This command was introduced on the Cisco 2500 series, 3600 series, and Cisco AS5300.
12.0(5)T	The SGCPAPP application was supported initially on the Cisco AS5300 universal access server in a private release that was not generally available.
12.0(7)XK	Support for the SGCPAPP application was extended to the Cisco MC3810 multiservice concentrator and the Cisco 3600 series routers (except for the Cisco 3620) in a private release that was not generally available.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(3)T	The MGCPAPP application was supported initially on the Cisco AS5300 universal access server.
12.1(3)XI	The <b>out-bound</b> keyword was added for the store-and-forward fax feature on the Cisco AS5300 universal access server.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Release	Modification
12.2(15)ZJ	The session application referred to by the <b>default</b> value of the <i>application-name</i> argument was updated to include support for Open Settlement Protocol (OSP), call transfer, and call forwarding. The version of the session application referred to by <b>default</b> in Cisco IOS Release 12.2(13)T and earlier releases was renamed default.c.old.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

### Usage Guidelines

Use this command to associate a predefined session application with an incoming POTS dial peer or an outgoing MMoIP dial peer. Calls using this incoming POTS dial peer or this outgoing MMoIP dial peer will be handed to the predefined specified session application.



#### Note

In Cisco IOS Release 12.3(4)T and later releases, the application name default refers to the new version of the default session application that supports OSP, call transfer, and call forwarding. The default session application in Cisco IOS Release 12.2(13)T and earlier releases has been renamed default.old.c and can still be configured for specific dial peers through the **application** command or globally configured for all inbound dial peers through the **call application global** command.

### SGCP Networks

For SGCP networks, enter **SGCPAPP** in uppercase characters. This application can be applied only to POTS dial peers. Note that SGCP dial peers do not use dial peer hunting.



#### Note

In Cisco IOS Release 12.2, you cannot mix SGCP and non-SGCP endpoints in the same T1 controller. You also cannot mix SGCP and non-SGCP endpoints in the same DS0 group.

### MGCP Networks

For MGCP networks, enter **MGCPAPP** in upper-case characters. This application can be applied only to POTS dial peers. Note that MGCP dial peers do not use dial peer hunting.

### Examples

The following example shows how to define an application and how to apply it to an outbound MMoIP dial peer for the fax onramp operation:

```
call application voice fax_on_vfc_onramp http://santa/username/clid_4digits_npw_3.tcl
dial-peer voice 3 mmoip
  application fax_on_vfc_onramp out-bound
  destination-pattern 57108..
  session target mailto:$d$@mail-server.cisco.com
```

The following example shows how to apply the MGCP application to a dial peer:

```
dial-peer voice 1 pots
  application MGCPAPP
```

Related Commands	Command	Description
	<b>call application voice</b>	Defines the name to be used for an application and indicates the location of the appropriate IVR script to be used with this application.
	<b>mgcp</b>	Starts the MGCP daemon.
	<b>sgcp</b>	Starts and allocates resources for the SCGP daemon.
	<b>sgcp call-agent</b>	Defines the IP address of the default SGCP call agent.

# call application global

To configure an application to use for incoming calls whose incoming dial peer does not have an explicit application configured, use the **call application global** command in global configuration mode. To remove the application, use the **no** form of this command.

**call application global** *application-name*

**no call application global** *application-name*

## Syntax Description

<i>application-name</i>	Character string that defines the name of the application.
-------------------------	--

## Defaults

The default application is **default** for all dial peers.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(15)ZJ	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

## Usage Guidelines

The application defined in the dial peer always takes precedence over the global application configured with the **call application global** command. The application configured with this command executes only when a dial peer has no application configured.

The application you configure with this command can be an application other than the default session application, but it must be included with the Cisco IOS software or be loaded onto the gateway with the **call application voice** command before using this command. If the application does not exist in Cisco IOS software or has not been loaded onto the gateway, this command will have no effect.



### Note

In Cisco IOS Release 12.3(4)T and later releases, the application name default refers to the new version of the default session application that supports Open Settlement Protocol (OSP), call transfer, and call forwarding. The default session application in Cisco IOS Release 12.2(13)T and earlier releases has been renamed default.old.c and can still be configured for specific dial peers through the **application** command or globally configured for all inbound dial peers through the **call application global** command.

## Examples

In the following example, the `clid_authen_collect` application is configured as the global application for all inbound dial peers that do not have a specific application configured:

```
call application global clid_authen_collect
```

Related Commands	Command	Description
	application	Enables a specific IVR application on a dial peer.
	call application voice	Defines the name to be used for an application and indicates the location of the appropriate IVR script to be used with this application.

# call application voice default disc-prog-ind-at-connect

To convert a DISCONNECT message with Progress Indicator set to PROG\_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state, use the **call application voice default disc-prog-ind-at-connect** command in global configuration mode. To revert to a DISCONNECT message with Progress Indicator set to PROG\_INBAND (PI=8) when the call is in the active state, use the **no** form of this command.

**call application voice default disc-prog-ind-at-connect [1 | 0]**

**no call application voice default disc-prog-ind-at-connect [1 | 0]**

## Syntax Description

<b>1</b>	(Optional) Convert a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state.
<b>0</b>	(Optional) Revert to a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) when the call is in the active state.

## Defaults

The DISCONNECT message has Progress Indicator set to PROG\_INBAND (PI=8) when the call is in the active state.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(15)ZJ	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

## Usage Guidelines

This command has no effect if the call is not in the active state.

This command is available for the default voice application. It may not be available when using some TCL IVR applications.

The Cisco IOS command-line interface command completion and help features do not work with this command. The command must be typed correctly and in its entirety.



### Note

This command may be replaced in future releases, or its syntax may change in a way that is not backward compatible.

## Examples

In the following example, a DISCONNECT message with Progress Indicator set to PROG\_INBAND (PI=8) is converted to a regular DISCONNECT message when the call is in the active state:

```
call application voice default disc-prog-ind-at-connect 1
```

# debug voip application

To display all application library debug messages, use the **debug voip application** command in privileged EXEC mode. To disable the debug output, use the **no** form of this command.

**debug voip application** [**core** | **all** | **callsetup** | **digitcollect** | **error** | **redirect** | **session** | **supplementary-services**]

**no debug voip application** [**core** | **all** | **callsetup** | **digitcollect** | **error** | **redirect** | **session** | **supplementary-services**]

Syntax Description		
	<b>all</b>	Displays all application library debugs.
	<b>callsetup</b>	Displays call setup being processed.
	<b>core</b>	Displays debug messages for the Application Framework library
	<b>digitcollect</b>	Displays digits collected during the call.
	<b>error</b>	Displays Application Framework Session application errors.
	<b>redirect</b>	Displays call redirection handler debugs.
	<b>session</b>	Displays session application debugs.
	<b>supplementary-services</b>	Provides application layer tracing related to the processing of supplementary services requests.

**Defaults** Debugging is not enabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(15)ZJT	This command was introduced.
	12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

**Usage Guidelines** With no options specified, the **debug voip application** command displays application program interface (API) libraries being processed.

The **debug voip application all** command differs from the **debug voip ivr all** command in that the former enables all application framework debugs, and the latter enables both the Application Framework Session debugs and the interactive voice response (IVR) debugs.

[Table 1](#) lists those commands that have been replaced in Cisco IOS Release 12.2(15)ZJ:

**Table 1** Replaced debug voip ivr Commands

Command in Cisco IOS Release 12.2(13)T	Replacement Command in Cisco IOS Release 12.3(4)T
<b>debug voip ivr applib</b>	<b>debug voip application core</b>
<b>debug voip ivr callsetup</b>	<b>debug voip application callsetup</b>

**Table 1** Replaced debug voip ivr Commands (continued)

Command in Cisco IOS Release 12.2(13)T	Replacement Command in Cisco IOS Release 12.3(4)T
<b>debug voip ivr digitcollect</b>	<b>debug voip application digitcollect</b>
<b>debug voip ivr redirect</b>	<b>debug voip application redirect</b>
<b>debug voip ivr supplementary-services</b>	<b>debug voip application supplementary-services</b>

**Examples**

The following examples are from the code for Cisco IOS Release 12.2(15)ZJ.

The following output is displayed when the **debug voip application callsetup** command is entered:

```
Router# debug voip application callsetup
```

```
ivr call setup debugging is on
Router#
*Mar 7 22:08:40.032://7//APPL:/afsSettlementValidateCall:target=, tokenp=0x0
*Mar 7 22:08:41.864://-1//PCM :LP:HN23A698CC:HN23A691A4:/InitiateCallSetup:Mode 1
RedirectMode 6 Incoming leg[-1] AlertTime -1 Destinations(1) [ 405 ]
*Mar 7 22:08:41.868://-1//PCM :HN23A698D0:/InitiateCallSetup:Destination 0 guid
:231D511B.1A5F11CC.800BB191.E9DE175D
*Mar 7 22:08:41.868: incoming_guid :00000000.00000000.00000000.00000000
*Mar 7 22:08:41.868://-1//PCM :HN23A698D0:/DNInitiate:Destination[405]
*Mar 7 22:08:41.868://-1//PCM :HN23A698D0:/DNMatchDialPeer:
*Mar 7 22:08:41.868: src carrier id:, tgt carrier id:
*Mar 7 22:08:41.868://-1//PCM :HN23A698D0:/DNQueuePeers:Matched peers(1)
*Mar 7 22:08:41.868://-1//PCM :HN23A698D0:/DNSSetupPeer: Destination 0x6221092C
*Mar 7 22:08:41.872://-1//PCM :HN23A698D0:/DNSSetupPeer:dialpeer tags for Rotary = 400
*Mar 7 22:08:41.872://-1//PCM :HN23A698D0:/DNSSetupPeer:
*Mar 7 22:08:41.872:Destination SetupPeer cid(-1), destPat(405), match(2), prefix(),
peer(630D95B0)
*Mar 7 22:08:41.872://-1//PCM :HN23A698D0:/DNSSettlementMatrixCheck:retcode=1 cid(-1)
trans=0x0, provider=0 No settle-call present
*Mar 7 22:08:41.940://8//PCM
:/DNHandler:(DN_SETTING[2])--(CC_EV_CALL_PROCEEDING[25])--IGNORED-->>(DN_SETTING[2])
*Mar 7 22:08:41.940://8//PCM :/CS_Setting_PROCEED:
*Mar 7 22:08:41.940://8//PCM :/CSPopLegAndWait:
*Mar 7 22:08:41.940://8//PCM :/CallSetupHandler:(CS_SETTING[0])
----- (CS_EV_PROCEEDING[3]) ----->>>(CS_SETTING[0])
*Mar 7 22:08:41.948://-1//PCM :HN23A698CC:/CSInterceptEvent:CallSetup[0x6320B998]
handlercount=1 Waits=1 #Objects=2 (CS_SETTING)
*Mar 7 22:08:41.948://8//PCM :/CSInterceptEvent:(CS_SETTING[0]) intercepting
CS_EV_PROGRESS leg 8 (Mask=12)
*Mar 7 22:08:41.948://-1//PCM :HN23A698CC:/CSInterceptEvent:CallSetup[0x6320B998]
handlercount=1 Waits=1 #Objects=2 (CS_SETTING)
*Mar 7 22:08:41.952://-1//PCM :HN23A698CC:/CallSetupContinueEvent:CallSetup[0x6320B998]
handlercount=1 Waits=1 #Objects=2 (CS_SETTING)
*Mar 7 22:08:41.956://8//PCM :/CS_CutProgress:
*Mar 7 22:08:41.956://8//PCM :/CSPopLegAndWait:
*Mar 7 22:08:41.956://8//PCM :/CallSetupContinueEvent:(CS_SETTING[0])
----- (CS_EV_PROGRESS[15]) ----->>>(CS_CONFEDALERT[5])
*Mar 7 22:08:41.956://-1//PCM :HN23A698CC:/CallSetupHandleQueueEvents:
*Mar 7 22:08:41.956://8//PCM :/CallSetupContinueEvent: *** Leaving function
***CallSetup[0x6320B998] handlercount=1 Waits=1 #Objects=2 (CS_CONFEDALERT)
*Mar 7 22:08:43.864://8//PCM :/CS_ConfedAlert_CONNECTED:no of Destinations:1
*Mar 7 22:08:43.864://-1//PCM :HN23A698CC:/CSDiscReturnAndEmptyLegALL:
*Mar 7 22:08:43.864://8//PCM :/CSPopLegAndWait:
*Mar 7 22:08:43.864://-1//PCM :HN23A698CC:/CSReturnIFDone:CallSetup[0x6320B998]
handlercount=1 Waits=1 #Objects=2 (CS_CONFED)
*Mar 7 22:08:43.864:CallSetupDump:CallSetup[0x6320B998] State:CS_CONFED[3] #Handler=1
#Waits=1#Objects=2
```

```

*Mar 7 22:08:43.864:All Destinations:
*Mar 7 22:08:43.864:DestinationDump:Destination[0x6221092C]:DN_SETTING[2] Holding[0]
Leg[8]
*Mar 7 22:08:43.864:settlement_in_use:0 settlement_transaction:0x0 settlement_provider:0
settlement_type:0 settlement_callvalid:1 busyRotary:0
*Mar 7 22:08:43.864:WaitList of Destinations:{HAN[DN_HAND ][CS_HAND ]( )}
*Mar 7 22:08:43.868:Handler Tree{HAN[CS_HAND ][AFS_HAND ]( HAN[DN_HAND ][CS_HAND ][FALSE]
[UC=1 ]LEG[8 ][CS_HAND ][LEG_OUTCONNECTED(8)] [Cause(0)] [UC=1 ])}
*Mar 7 22:08:43.868:{HAN[DN_HAND ][CS_HAND ]( )}
*Mar 7 22:08:43.868:
*Mar 7 22:08:43.868:Handler Tree Trace
*Mar 7 22:08:43.868://-1//PCM :HN23A698D0:/DNCleanup:Terminate=TRUE Status DN_SUCCESS
Leg[8]
*Mar 7 22:08:43.868://-1//PCM :HN23A698D0:/DNSettlementCleanup:cid(-1) trans=0x0,
provider=0
*Mar 7 22:08:43.868://-1//PCM :HN23A698D0:/DNSetFree:
*Mar 7 22:08:43.868://-1//PCM :HN23A698CC:/CSReturnIFDone:CallSetup[0x6320B998]
handlercount=0 #Waits=0 #Objects=1 (CS_CONFED)
*Mar 7 22:08:43.872://-1//PCM :HN23A698CC:/CSReturnIFDone:decoupled extern connection
*Mar 7 22:08:43.872://-1//PCM :HN23A698CC:/CSReturnIFDone:CallSetup Returning(ls_000
Status CS_ACTIVE)
*Mar 7 22:08:43.872://8//PCM :/CallSetupHandler:(CS_CONFEDALERT[5])
----- (CS_EV_CONNECTED[5]) ----->>> (CS_CONFED[3])
*Mar 7 22:08:43.872://-1//PCM :HN23A698CC:/CallSetupCleanup:Terminate=FALSE
*Mar 7 22:08:43.876://-1//PCM :HN23A698CC:/CallSetupCleanup:State CS_CONFED

```

The following output is displayed when the **debug voip application digitcollect** command is entered:

```

Router# debug voip application digitcollect

ivr digit collect debugging is on
Router#
*Mar 7 22:09:08.108://9//DCM :/DigitCollect:DialPlan=TRUE AbortKey= TermKey=# NumPatts=0
Enable=FALSE InterruptPrompt=FALSE maxDigits=0 DialPlanTerm=FALSE
*Mar 7 22:09:08.108://9//APPL:/AppTypeAheadGetDigit:no chars in buffer.
*Mar 7 22:09:08.112://9//DCM :/act_DCRunning_RDone:callid=9 Enable succeeded.enable=0
matchDialplan=1 numPatterns=0matchDialplanTerm=0
*Mar 7 22:09:11.428://9//APPL:/AppVcrControlEvent:VCR Control, not enabled.---
*Mar 7 22:09:11.428://9//APPL:/AppTypeAheadEvent:Passing, not enabled.---
*Mar 7 22:09:11.428://9//DCM :/act_DCRunning_Digit::pLeg 9 Digit 4 Tone Mode 0
*Mar 7 22:09:11.428://9//DCM :/DCTreatDigit:
*Mar 7 22:09:11.428://-1//DCM :HN23A6FF50:/DCTreatDigit:
*Mar 7 22:09:11.428: src carrier id:, tgt carrier id:
*Mar 7 22:09:11.428://-1//DCM :HN23A6FF50:/DCTreatDigit:Match single infotype
*Mar 7 22:09:11.676://9//APPL:/AppVcrControlEvent:VCR Control, not enabled.---
*Mar 7 22:09:11.676://9//APPL:/AppTypeAheadEvent:Passing, not enabled.---
*Mar 7 22:09:11.676://9//DCM :/act_DCRunning_Digit::pLeg 9 Digit 0 Tone Mode 0
*Mar 7 22:09:11.676://9//DCM :/DCTreatDigit:
*Mar 7 22:09:11.680://-1//DCM :HN23A6FF50:/DCTreatDigit:
*Mar 7 22:09:11.680: src carrier id:, tgt carrier id:
*Mar 7 22:09:11.680://-1//DCM :HN23A6FF50:/DCTreatDigit:Match single infotype
*Mar 7 22:09:11.908://9//APPL:/AppVcrControlEvent:VCR Control, not enabled.---
*Mar 7 22:09:11.908://9//APPL:/AppTypeAheadEvent:Passing, not enabled.---
*Mar 7 22:09:11.908://9//DCM :/act_DCRunning_Digit::pLeg 9 Digit 5 Tone Mode 0
*Mar 7 22:09:11.908://9//DCM :/DCTreatDigit:
*Mar 7 22:09:11.908://-1//DCM :HN23A6FF50:/DCTreatDigit:
*Mar 7 22:09:11.908: src carrier id:, tgt carrier id:
*Mar 7 22:09:11.908://-1//DCM :HN23A6FF50:/DCTreatDigit:Match single infotype
*Mar 7 22:09:11.912://9//DCM :/act_DCRunning_RDone:callid=9 Reporting disabled.
*Mar 7 22:09:11.912://-1//DCM :HN23A6FF50:/DigitCollectComplete:Status
4=DC_MATCHED_DIALPLAN. Digits=405
*Mar 7 22:09:11.916://-1//DCM :HN23A6FF50:/DCHandlerCleanup:

```

The following output is displayed when the **debug voip application session** command is entered:

```

Router# debug voip application session

applib session debugging is on
*Apr 4 23:57:08.054://-1//APPL:HN04B2BC78:LG35:/AFS_CALLSETUPIND:Calling #(4153788854),
Called #(52984), peer_tag(1)
*Apr 4 23:57:08.054://-1//APPL:HN04B2BC78:LG35:/afsSetupCall:Called #(52984)
*Apr 4
23:57:08.058://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLINIT)--(CC_EV_CALL_SETUP_IND)-
->>(CONTACTINGDEST)
*Apr 4 23:57:08.466://-1//APPL:HN04B2BC78:LG36:/AFS_ContactingDest_ALERT:
*Apr 4 23:57:08.470://-1//APPL:HN04B2BC78:LG36:/AFS_ContactingDest_ALERT:inID(35),
outID(36), outbnd peer_tag(6), prog_ind(8)
*Apr 4
23:57:08.470://-1//APPL:HN04B2BC78:LG36:/afsMsgHandler:(CONTACTINGDEST)--(CC_EV_CALL_ALERT
)-->>(CONFINGALERT)
*Apr 4 23:57:08.470://-1//APPL:HN04B2BC78:CN11:/AFS_ConfingAlert_CREATEDONE:
{HAN[AFS_HAND][NULL ]( LEG[35 ])[AFS_HAND][LEG_INCALERTING(4)][Cause(0)][UC=1
]HAN[CS_HAND][AFS_HAND][FALSE][UC=1 ]LEG[36 ]][CS_HAND
][LEG_OUTINIT(6)][Cause(0)][UC=1 ]CON[11 ]][AFS_HAND][CONNECTION_CONFED(2)][UC=1 ]}
*Apr 4
23:57:08.470://-1//APPL:HN04B2BC78:CN11:/afsMsgHandler:(CONFINGALERT)--(CC_EV_CONF_CREATE
DONE)-->>(CONFEDALERT)
*Apr 4
23:57:08.478://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CONFEDALERT)--(CC_EV_VOICE_MODE_DO
NE)-->>(CONFEDALERT)
*Apr 4 23:57:24.162://-1//APPL:HN04B2BC78:HN04B2BC78:/AFS_ConfedAlert_SETUPDONE:
*Apr 4 23:57:24.162://-1//APPL:HN04B2BC78:HN04B2BC78:/afsAppHandlerCleanup:CS_HAND
*Apr 4
23:57:24.162://-1//APPL:HN04B2BC78:/afsMsgHandler:(CONFEDALERT)--(APP_EV_CALLSETUP_DONE)--
>>(CALLACTIVE)
*Apr 4
23:57:24.182://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLACTIVE)--(CC_EV_VOICE_MODE_DON
E)-->>(CALLACTIVE)
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:LG35:/AFS_DISCONNECT:
*Apr 4
23:57:34.838://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CALLACTIVE)--(CC_EV_CALL_DISCONNEC
TED)-->>(CALLDISCONNECT)
*Apr 4 23:57:34.838://-1//APPL:/afsCallProcess:[HANDLERDONE_EVENT_END]
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:/afsHNDCleanup:Terminate TRUE Terminated
FALSE{HAN[AFS_HAND][NULL ]( LEG[35 ])[AFS_HAND][LEG_INCCONNECTED(5)][Cause(16)][UC=1
]LEG[36 ]][AFS_HAND][LEG_OUTCONNECTED(8)][Cause(0)][UC=1 ]CON[11
][AFS_HAND][CONNECTION_CONFED(2)][UC=1 ]}
*Apr 4 23:57:34.838://-1//APPL:HN04B2BC78:CN11:/afsMsgHandler:(CC_EV_CONF_DESTROY_DONE)
*Apr 4
23:57:34.854://-1//APPL:HN04B2BC78:LG35:/afsMsgHandler:(CC_EV_CALL_DISCONNECT_DONE)
*Apr 4
23:57:34.862://-1//APPL:HN04B2BC78:LG36:/afsMsgHandler:(CC_EV_CALL_DISCONNECT_DONE)
*Apr 4 23:57:34.862://-1//APPL:/afsCallProcess:[HANDLERDONE_EVENT_END]
*Apr 4 23:57:34.862://-1//APPL:HN04B2BC78:/afsHNDCleanup:Terminate TRUE Terminated
TRUE{HAN[AFS_HAND][NULL ]( )}
*Apr 4 23:57:34.862://-1//APPL:HN04B32530:/afsFreeHND:Hndlr returned to the free queue

```

Table 2 describes the significant fields shown in the display.

**Table 2** debug voip application Field Descriptions

Field	Description
Called #	Called # may not appear in the initial /AFS_CALLSETUPIND message, but it appears in later in the /afsSetupCall message.

*Table 2 debug voip application Field Descriptions (continued)*

Field	Description
peer_tag	Dial peer tag.
/afsFreeHND	Verifies that the application completed properly.

**Related Commands**

Command	Description
<b>debug voip ivr all</b>	Displays all IVR and application framework messages.

# Glossary

**e-phone**—IP phone that is registered with a Cisco IOS voice gateway operating in IOS Telephony Service mode.

**H.323**—H.323 allows dissimilar communication devices to communicate with each other by using a standardized communication protocol. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.

**H.450.2**—Call transfer supplementary service for H.323.

**H.450.3**—Call diversion supplementary service for H.323.

**IVR**—interactive voice response. Term used to describe systems that provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words or, more commonly, DTMF signaling. Examples include banks that allow you to check your balance from any telephone and automated stock quote systems.

**OSP**—Open Settlement Protocol. Client/server protocol defined by the ETSI TIPHON to establish authenticated connections between gateways, and to allow gateways and servers to transfer accounting and routing information securely. OSP allows service providers to roll out VoIP services without establishing direct peering agreements with other ITSPs.

**SIP**—Session Initiation Protocol. Protocol developed by the IETF MMUSIC Working Group as an alternative to H.323. SIP features are compliant with IETF RFC 2543, published in March 1999. SIP equips platforms to signal the setup of voice and multimedia calls over IP networks.

**TCL**—Tool Command Language. A scripting language used for gateway products both internally and externally to Cisco IOS software code.



Note

---

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

---

