



# MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

---

**First Published: October 30, 2005**

**Last Updated: June 19, 2006**

The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature provides the capability for Cisco IOS voice gateways to present analog and BRI phones to be controlled by the Cisco Unified CallManager and Cisco Unified CallManager Express (Cisco Unified CME) as though they were Cisco IP phones, enabling the following:

- Line-side support for the Multilevel Precedence and Preemption (MLPP) feature under Cisco CallManager control (Cisco Unified CME control is not supported)
- Cisco CallManager registration of analog and BRI endpoints (Cisco Unified CME control is not supported)
- Cisco CallManager endpoint autoconfiguration support
- Modem pass-through support
- Cisco Survivable Remote Site Telephony (SRST) support under Cisco CallManager control

## Feature History for the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature

Release	Modification
Cisco IOS Release 12.3(14)T	This feature was introduced.
Cisco IOS Release 12.4(9)T	This feature was implemented on the following platforms: Cisco 2801, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3825, Cisco 3845, Cisco VG 224.

## 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.



---

**Corporate Headquarters:**

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

Copyright © 2006 Cisco Systems, Inc. All rights reserved.

# Contents

- [Prerequisites for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 2](#)
- [Restrictions for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 2](#)
- [Information About MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 3](#)
- [How to Configure the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature, page 6](#)
- [Configuration Examples for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 24](#)
- [Additional References, page 28](#)
- [Command Reference, page 30](#)
- [Glossary, page 101](#)

## Prerequisites for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

Make sure that the following tasks have been completed before configuring the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature:

- Skinny Client Control Protocol (SCCP) version 4.0 or higher is enabled.
- Cisco CallManager 4.1 or a later release is running.
- Cisco IOS Release 12.4(9)T or a later release is running.

## Restrictions for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

- Unlike Cisco IP phones, which maintain active communications with all Cisco CallManagers, gateway-controlled endpoints communicate only with the active Cisco CallManager.
- MLPP is not supported in SRST environments or under Cisco Unified CME-control.
- Unlike Cisco IP phones, gateway-controlled phones preempted for reuse and remaining off-hook will not be auto-idled by Cisco CallManager.
- The following supplementary services are not supported:
  - Hold
  - Callback
  - Transfer
  - Call Forward
  - Redial
- Only a single ISDN B channel is supported for BRI voice calls.
- ISDN circuit-switched or packet-switched data is not supported on BRI calls.

- Fallback to other protocols, such as H.323 or Media Gateway Control Protocol (MGCP), is not supported.
- The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature is supported on the following ISDN switches only:
  - National ISDN-1 switches
  - NET3 ISDN switches, including Norway NET3, Australia NET3, and New Zealand NET3 switches (covers the Euro-ISDN E-DSS1 signaling system and is European Telecommunications Standards Institute (ETSI)-compliant).

## Information About MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

To configure the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature, you should understand the following concepts:

- [Benefits of MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature, page 3](#)
- [Feature Design of MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 3](#)

## Benefits of MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature

- Supports the use of existing customer premises equipment (CPE) in voice networks, by allowing legacy analog and BRI phones attached to a Cisco IOS gateway to be controlled by the Cisco CallManager.
- Facilitates replacement of public switched telephone network (PSTN) time-division multiplexing (TDM) infrastructure with VoIP.

## Feature Design of MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

Prior to the implementation of the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature, there was no Cisco CallManager support for existing CPE, that is, Cisco IOS gateway-controlled legacy analog and BRI telephony endpoints. There was no capability to translate call control messages between the CPE and Cisco CallManager for calls connecting over the VoIP network to the existing PSTN. The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature provides this support by presenting CPE devices to the Cisco CallManager as though they were Cisco IP phones. The new feature uses Cisco proprietary SCCP to communicate call control messages between the Cisco CallManager and gateway endpoints, which can be foreign exchange station (FXS) analog or BRI ISDN encryption-enabled phones. Cisco implements this support by the use of the SCCP Telephony Control Application (STCAPP) software that runs on existing line-side gateways. The line-side gateway translates call control messages between the Cisco CallManager SCCP and the Call Control Application Programming Interface (CCAPI), allowing the attached analog and BRI phones to be controlled by the Cisco CallManager in the same way that Cisco IP phones are controlled. To the Cisco CallManager, each gateway appears to be a collection of multiple SCCP-controlled IP phones.

The new feature supports the following capabilities:

- Basic call support (supplementary services are not supported).
- Cisco CallManager registration of analog and BRI endpoints.
- Autoconfiguration using TFTP download of extensible markup language (XML) dial-peer configuration files from the Cisco CallManager.
- MLPP support to analog and BRI endpoints. MLPP support enables the voice gateway to interoperate with other MLPP-capable networks for call preemption and precedence.
- Cisco Survivable Remote Site Telephony (SRST), enabling routers to provide call-handling support for Cisco IP phones, analog, and BRI endpoints when connections to remote primary, secondary, or tertiary Cisco CallManager installations are lost, or when the WAN connection is down. For more information on configuring SRST refer to the following:
  - [Cisco IOS Survivable Remote Site Telephony Version 3.3 System Administrator Guide](#)
  - [Cisco IOS Survivable Remote Site Telephony Version 3.3 Command Reference](#)
- Modem pass-through allowing secure telephones to use modem signaling to establish secure communications during a call.

## Supported Endpoints

The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature provides support for existing CPE, that is legacy analog and BRI telephony endpoints (phones). You may connect the following phone types in a VoIP network where the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature is enabled:

- Analog phones, either standard or encryption enabled.
- ISDN BRI phones, capable of carrying standard or encrypted phone calls.

All these phones appear to the Cisco CallManager as though they were Cisco IP phones, thereby enabling MLPP capability.

## Supported Call Types

In addition to basic voice calls, you may make the following calls in a VoIP network where the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature is enabled:

- MLPP calls: MLPP service allows properly validated users to place priority calls, and if necessary, users can preempt lower-priority phone calls. Precedence designates the priority level that is associated with a call. Preemption designates the process of terminating lower-precedence calls that are currently using the target device, (either trunks or stations), so a call of higher precedence can be extended to or through the device. An authenticated user can preempt calls either to targeted stations or through fully subscribed time division multiplexing (TDM) trunks. This capability assures high-ranking personnel of communication to critical organizations and personnel during network stress situations, such as a national emergency or degraded network situations.
- Modem pass-through calls. Modem pass-through, the transport of modem signals through a packet network using pulse code modulation (PCM) encoded packets, uses a Cisco proprietary protocol, Named Signaling Events (NSE), to signal events that are exchanged between gateways, using Real-time Transport Protocol (RTP) packets.

## Call Control Components

The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature uses a number of software components to achieve call control. In order for the Cisco CallManager to manage gateway endpoints, that is, phones that are connected to Cisco IOS gateways, the Cisco CallManager exchanges messages with gateways, and the gateways exchange messages with the phones. Call control message exchange involves the following processes:

- Cisco IP Phones communicate with the Cisco CallManager through SCCP, which comprises a messaging set between a skinny client (IP phone) and the Cisco CallManager call-processing server. With the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature enabled, SCCP is enabled on the gateway that sits between endpoints and the Cisco CallManager, thereby allowing endpoints to communicate with the Cisco CallManager as though they were Cisco IP phones.
- The MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature implements a new Cisco IOS application, the SCCP Telephony Control Application (STCAPP), to translate SCCP call control messages received from the Cisco CallManager. STCAPP processes messages received from the Cisco CallManager by using the Device Control API (DCAPI), a software layer that interfaces STCAPP with SCCP. STCAPP then interfaces with the physical analog and ISDN phones by way of the CCAPI software layer. CCAPI, which provides the interface between the software and the actual FXS and BRI ports on the line-side gateway, tells the physical devices to provide dial-tone, ringing, ringback, digit-collection, and so forth.

## Supported Gateways, Modules and Voice Interface Cards

Table 1 provides a list of supported gateways, modules, and voice interface cards (VICs).

**Table 1** Supported Gateways, Modules, and VICs for the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature

Supported Gateways	Supported Extension Modules	Supported Network Modules and Expansion Modules	Supported VICs
<ul style="list-style-type: none"> <li>• Cisco 2801</li> <li>• Cisco 2811</li> <li>• Cisco 2821</li> <li>• Cisco 2851</li> <li>• Cisco 3825</li> <li>• Cisco 3845</li> </ul>	—	<ul style="list-style-type: none"> <li>• NM-HD-1V</li> <li>• NM-HD-2V</li> <li>• NM-HD-2VE</li> </ul>	<ul style="list-style-type: none"> <li>• VIC2-2FXS</li> <li>• VIC-4FXS/DID</li> <li>• VIC2-2BRI-NT/TE</li> </ul>
<ul style="list-style-type: none"> <li>• Cisco 2801</li> <li>• Cisco 2821</li> <li>• Cisco 2851</li> <li>• Cisco 3825</li> <li>• Cisco 3845</li> </ul>	<ul style="list-style-type: none"> <li>• EVM-HD</li> </ul>	<ul style="list-style-type: none"> <li>• EVM-HD-8FXS/DID</li> <li>• EM-3FXS/4FXO</li> <li>• EM-HDA-8FXS</li> <li>• EM-4BRI-NT/TE</li> </ul>	—

Supported Gateways	Supported Extension Modules	Supported Network Modules and Expansion Modules	Supported VICs
<ul style="list-style-type: none"> <li>• Cisco 2801</li> <li>• Cisco 2811</li> <li>• Cisco 2821</li> <li>• Cisco 2851</li> <li>• Cisco 3825</li> <li>• Cisco 3845</li> </ul>	—	<ul style="list-style-type: none"> <li>• NM-HDV2</li> <li>• NM-HDV2-1T1/E1</li> <li>• NM-HDV2-2T1/E1</li> </ul>	<ul style="list-style-type: none"> <li>• VIC2-2FXS</li> <li>• VIC-4FXS/DID</li> <li>• VIC2-2BRI-NT/TE</li> </ul>
<ul style="list-style-type: none"> <li>• Cisco VG 224</li> </ul>	—	—	—

## How to Configure the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways Feature

This section contains the following procedures:

- [Configuring the SCCP Gateway and Gateway Endpoints on Cisco CallManager, page 6](#) (required)
- [Configuring Cisco CallManager Download on Cisco IOS Gateways, page 9](#) (required)
- [Configuring SCCP on Cisco IOS Gateways, page 10](#) (required)
- [Configuring STCAPP on Cisco IOS Gateways, page 11](#) (optional)
- [Configuring ISDN BRI Voice Calls, page 12](#) (optional)
- [Configuring Modem Pass-Through Calls, page 14](#) (optional)
- [Verifying MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways, page 15](#) (optional)

## Configuring the SCCP Gateway and Gateway Endpoints on Cisco CallManager

This task configures the SCCP gateway and SCCP gateway-controlled endpoints on the Cisco CallManager.

### Cisco CallManager Autoconfiguration


There are two methods of configuring the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature, either by using Cisco CallManager autoconfiguration or by manually configuring SCCP and STCAPP on the gateway. The first method allows you to configure the SCCP gateway and SCCP gateway controlled endpoints on the Cisco CallManager, then download XML configuration files for the endpoints to the Cisco IOS gateway. The second method requires you to manually enable STCAPP and manually configure gateway endpoints. We recommend the Cisco CallManager autoconfiguration method because it allows you to configure all devices in one place, eliminating endpoint configuration on the gateway.

Perform the following task to use Cisco CallManager autoconfiguration.

## SUMMARY STEPS

1. Choose **Add a New Device** or **Add a New Gateway** in the Cisco CallManager menu.
2. Choose the new SCCP gateway settings.
3. Configure the gateway MAC address, network modules, voice interface cards, and ports.
4. Save the configuration.
5. Verify the configuration.

## DETAILED STEPS

- 
- Step 1** In the drop-down list in Cisco CallManager, choose **Device > Add a New Device > Gateway** (from the **Device Type**) or choose **Device > Gateway > Add a New Gateway**.
- Step 2** Choose the appropriate settings for the SCCP gateway.
- a. Choose the gateway type.
  - b. Choose the SCCP option for device protocol.
- Step 3** Enter the appropriate SCCP gateway MAC address and configure the network modules, voice interface cards, and ports.
- a. Enter the last ten characters of the MAC address of the interface used to register with the Cisco CallManager. Use the **show interface** command on the SCCP configured interface on the gateway to determine the gateway MAC address. (This MAC address is the same as the address of the SCCP gateway local interface manually configured in Step 4 of the “[DETAILED STEPS](#)” section of “Configuring SCCP in Cisco IOS Gateways.”)
  - b. Enter the gateway name.
  - c. Enter the Cisco CallManager group number.
  - d. Configure the appropriate network modules, voice interface cards and ports.
-  **Note** Gateway VIC port and slot numbers are referred to as Endpoint Identifiers on the Cisco CallManager. For more information on Cisco CallManager gateway configuration, refer to the section “[Adding a Cisco IOS SCCP Gateway](#),” in the *Cisco CallManager Administration Guide*, Release 4.1(2).
- 
- Step 4** Click **Insert** to save a new gateway configuration, or click **Update** to save an existing gateway configuration.
- Step 5** Verify your configuration by displaying the list of configured analog and BRI phones. In the Cisco CallManager menu choose **Device > Phone > Find**. Analog phone device name begin with “AN” and BRI phone device names begin with “BR”.

## Configuring Cisco CallManager Download on Cisco IOS Gateways

This task configures automatic download capability of XML dial-peer configuration files from the Cisco CallManager and enables Cisco CallManager autoconfiguration.

**Note**

---

Although you may manually configure dial peers to work with the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature, we recommend that you use Cisco CallManager autoconfiguration.

---

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **ccm-manager sctp local** *interface-type interface-number*
4. **ccm-manager config** [**dialpeer-prefix** *prefix* | **server** {*ip-address* | *name*}]
5. **ccm-manager sctp**
6. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>ccm-manager sccp local interface-type interface-number</pre> <p><b>Example:</b> Router(config)# ccm-manager sccp local fastethernet0/0 </p>	<p>Selects the local interface that the SCCP application should use to register with Cisco CallManager. This interface must be specified before you enable the autoconfiguration process.</p> <ul style="list-style-type: none"> <li><i>interface-type</i>—Interface type that the SCCP application uses to register with Cisco CallManager.</li> <li><i>interface-number</i>—Interface number that the SCCP application uses to register with Cisco CallManager.</li> </ul>
Step 4	<pre>ccm-manager config [dialpeer-prefix prefix   server {ip-address   name}]</pre> <p><b>Example:</b> Router(config)# ccm-manager config dialpeer-prefix 888 </p>	<p>Enables the download of Cisco CallManager XML configuration files.</p> <ul style="list-style-type: none"> <li><b>dialpeer prefix</b> <i>prefix</i>—Configures the prefix to use for autogenerated dial peers. Range is 0 to 999999. The default is 999.</li> </ul> <p><b>Note</b> When manually adding a dial peer-prefix, select a prefix number other than the autoconfigured dial peer prefix (999 by default), to keep manually added dial peers from being deleted from the running configuration when the Cisco CallManager download happens in the gateway.</p> <ul style="list-style-type: none"> <li><b>server</b> <i>ip-address</i>—Specifies the IP address of the TFTP server from which to download the XML configuration files to the Cisco IOS gateway.</li> <li><b>server</b> <i>name</i>—Specifies the TFTP server name from which the Cisco IOS gateway downloads Cisco CallManager XML configuration files.</li> </ul>
Step 5	<pre>ccm-manager sccp</pre> <p><b>Example:</b> Router(config)# ccm-manager sccp </p>	<p>Enables the autoconfiguration process. This command immediately triggers the TFTP download of the XML configuration file.</p>
Step 6	<pre>exit</pre> <p><b>Example:</b> Router(config)# exit </p>	<p>Exits the current configuration mode.</p>

## Configuring SCCP on Cisco IOS Gateways

This task configures SCCP on the Cisco IOS gateway. SCCP messaging enables Cisco CallManager endpoint call control.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sccp ccm** {*ip-address* | *dns*} **identifier** *identifier-number* [**port** *port-number*][**version** *version-number*]
4. **sccp local** *interface-type interface-number*
5. **sccp ccm group** *group-number*
6. **associate ccm** *identifier-number* **priority** *priority-number*
7. **sccp**
8. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>sccp ccm</b> { <i>ip-address</i>   <i>dns</i> } <b>identifier</b> <i>identifier-number</i> [ <b>port</b> <i>port-number</i> ] [ <b>version</b> <i>version-number</i> ]  <b>Example:</b> Router(config)# sccp ccm 10.1.1.1 identifier 5	Adds a Cisco CallManager server to the list of available servers and sets various parameters. <ul style="list-style-type: none"> <li>• <i>IP address</i>— Specifies the IP address of the Cisco CallManager server.</li> <li>• <i>identifier-number</i>—Identifies the Cisco CallManager associated with the Cisco CallManager <i>group-number</i> configured in Step 5 . Valid entries are from 1 to 65535. There is no default value.</li> </ul>

	Command or Action	Purpose
Step 4	<p><code>sccp local interface-type interface-number</code></p> <p><b>Example:</b> Router(config)# sccp local fastethernet0/0</p>	<p>Selects the local interface that the SCCP application uses to register with Cisco CallManager. (This interface is the interface whose MAC address is specified for SCCP gateway registration using Cisco CallManager autoconfiguration in <a href="#">Step 3</a> of the “<a href="#">DETAILED STEPS</a>” section on page 7.)</p> <ul style="list-style-type: none"> <li><i>interface-type</i>—Specifies the interface type that the SCCP application uses to register with Cisco CallManager.</li> <li><i>interface-number</i>—Specifies the interface number that the SCCP application uses to register with Cisco CallManager.</li> </ul>
Step 5	<p><code>sccp ccm group group-number</code></p> <p><b>Example:</b> Router(config)# sccp ccm group 1</p>	<p>Creates a Cisco CallManager group.</p> <ul style="list-style-type: none"> <li><i>group-number</i>—Associates the Cisco CallManager group with the Cisco CallManager group <i>identifier</i> configured in Step 3. Range is 1 to 65535. There is no default value.</li> </ul>
Step 6	<p><code>associate ccm identifier-number priority priority-number</code></p> <p><b>Example:</b> Router(config)# associate ccm 5 priority 1</p>	<p>Associates a Cisco CallManager with a Cisco CallManager group.</p> <ul style="list-style-type: none"> <li><i>identifier-number</i>—Identifies the Cisco CallManager associated with the Cisco CallManager <i>group-number</i> configured in Step 5. Valid entries are from 1 to 65535. There is no default value.</li> <li><i>priority-number</i>—Priority of the Cisco CallManager within the Cisco CallManager group. Range is 1 to 4. There is no default value. The highest priority is 1.</li> </ul>
Step 7	<p><code>sccp</code></p> <p><b>Example:</b> Router(config)# sccp</p>	<p>Enables SCCP and its related applications.</p>
Step 8	<p><code>exit</code></p> <p><b>Example:</b> Router(config)# exit</p>	<p>Exits the current configuration mode.</p>

## Configuring STCAPP on Cisco IOS Gateways

This task manually configures STCAPP on the Cisco IOS gateway. STCAPP translates incoming SCCP call control messages received from the Cisco CallManager.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `stcapp ccm-group group-id`

4. **stcapp**
5. **stcapp timer roh** *seconds*
6. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>stcapp ccm-group</b> <i>group-id</i>  <b>Example:</b> Router(config)# stcapp ccm-group 1	Specifies the Cisco CallManager group number. <ul style="list-style-type: none"> <li>• The <b>ccm-group</b> <i>group-id</i> must match the Cisco CallManager identifier configured for the service- provider interface (SPI) using the <b>sccp ccm-group</b> [<i>group-id</i>] command.</li> </ul>
Step 4	<b>stcapp</b>  <b>Example:</b> Router(config)# stcapp	Enables the STCAPP.
Step 5	<b>stcapp timer roh</b> <i>seconds</i>  <b>Example:</b> Router(config)# stcapp timer roh 30	(Optional) Configures the STCAPP timer. <ul style="list-style-type: none"> <li>• <b>roh</b>—Configures the maximum duration a receiver off hook tone is played. The receiver off hook (ROH) tone signals the subscriber that the phone remains off hook when there is no active call.</li> <li>• <i>seconds</i>—Specifies the ROH tone timeout duration in seconds. Timeout must be a number in the range from 0 to 120. Default is 45 seconds.</li> </ul>
Step 6	<b>exit</b>  <b>Example:</b> Router(config)# exit	Exits the current configuration mode.

## Configuring ISDN BRI Voice Calls

This task configures ISDN BRI ports.


### SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **isdn switch-type** *switch-type*
4. **interface bri** *slot/port*
5. **isdn layer1-emulate network**
6. **isdn protocol-emulate** {*network* | *user*}
7. **isdn point-to-point-setup**
8. **isdn spid1** *spid*
9. **no shutdown**
10. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>isdn switch-type</b> <i>switch-type</i>  <b>Example:</b> Router(config)# isdn switch-type basic-ni	Configures the telephone-company ISDN switch type. The following ISDN switch types support the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature: <ul style="list-style-type: none"> <li><i>basic-ni</i>—Specifies national ISDN-1 switches.</li> <li><i>basic-net3</i>— Specifies NET3 ISDN: Norway NET3, Australia NET3, and New Zealand NET3 switches (covers the Euro-ISDN E-DSS1 signaling system and is ETSI-compliant)</li> </ul>
Step 4	<b>interface bri</b> <i>slot/port</i>  <b>Example:</b> Router(config)# interface bri 1/1	Enters interface configuration mode for the specified <i>slot/port</i> (location of voice network module and voice interface card).
Step 5	<b>isdn layer1-emulate network</b>  <b>Example:</b> Router(config-if)# isdn layer1-emulate network	Configures Layer 1 port mode emulation and clock status for the network.
Step 6	<b>isdn protocol-emulate</b> { <i>network</i>   <i>user</i> }	Configures Layer 2 and Layer 3 port mode emulation and clock status. <ul style="list-style-type: none"> <li><i>network</i>—Network termination (NT) side, that is, the clock slave.</li> </ul>

	Command or Action	Purpose
Step 7	<b>isdn point-to-point-setup</b>  <b>Example:</b> Router(config-if)# isdn point-to-point-setup	Configures the ISDN port to send SETUP messages on the static terminal endpoint identifier (TEI) point-to-point link.   <b>Note</b> Network-side BRI NI does not support point-to-multipoint topology.
Step 8	<b>isdn spid1 spid1</b>  <b>Example:</b> Router(config-if)# isdn spid1 40855522220101	Specifies the service-profile identifier (SPID).
Step 9	<b>no shutdown</b>  <b>Example:</b> Router(config-if)# no shutdown	Turns on the voice port.
Step 10	<b>exit</b>  <b>Example:</b> Router(config-if)# exit	Exits the current configuration mode.

## Configuring Modem Pass-Through Calls

This task configures modem pass-through operation on the gateway.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service { pots | voatm | vofr | voip }**
4. **modem passthrough nse [payload-type *number*] { codec { g711ulaw | g711alaw } } [redundancy [maximum-sessions *sessions*]]**
5. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>voice service {pots   voatm   vofr   voip}</b>  <b>Example:</b> Router(config)# voice service voip	Enters voice-service configuration mode and specifies a voice-encapsulation type. <ul style="list-style-type: none"> <li><b>voip</b>—Voice over IP (VoIP) encapsulation.</li> </ul>
Step 4	<b>modem passthrough nse [payload-type number] {codec {g711ulaw   g711alaw}} [redundancy [maximum-sessions sessions]]</b>  <b>Example:</b> Router(config-voi-serv)# modem passthrough nse codec g711ulaw	Enables fax or modem pass-through over VoIP globally for all dial peers. <ul style="list-style-type: none"> <li><b>nse</b>—Specifies that named signaling events (NSEs) are used to communicate codec switchover between gateways.</li> <li><b>codec</b>—Configures codec selections for upspeeding, the capability to automatically change the voice codec to G.711 if necessary and turn off echo cancellation (EC) and voice activity detection (VAD) for the duration of the call.</li> <li><b>g711ulaw</b>—Specifies codec G.711 mu-law, 64,000 bits per second for T1.</li> </ul>
Step 5	<b>exit</b>  <b>Example:</b> Router(config-voi-serv)# exit	Exits the current configuration mode.

## Verifying MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

This task verifies MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature configuration.

## SUMMARY STEPS

- show call active voice
- show call application voice stcapp
- show ccm-manager config- download
- show stcapp device voice-port *port-number*
- show stcapp device voice-port summary
- show stcapp statistics all

7. `show stcapp statistics voice-port port-number`

## DETAILED STEPS

**Step 1** Use the `show call active voice` command to display call information for voice calls in progress:

```
Router# show call active voice

Telephony call-legs: 2
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 2
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 4

GENERIC:
SetupTime=1557650 ms
Index=1
PeerAddress=
PeerSubAddress=
PeerId=999100
PeerIfIndex=14
LogicalIfIndex=10
ConnectTime=1562040 ms
CallDuration=00:01:01 sec
CallState=4
CallOrigin=2
ChargedUnits=0
InfoType=speech
TransmitPackets=3101
TransmitBytes=519564
ReceivePackets=3094
ReceiveBytes=494572
TELE:
ConnectionId=[0x11B1860C 0x22D711D7 0x8014E4D4 0x8FD15327]
IncomingConnectionId=[0x11B1860C 0x22D711D7 0x8014E4D4 0x8FD15327]
CallID=25
TxDuration=59670 ms
VoiceTxDuration=59670 ms
FaxTxDuration=0 ms
CoderTypeRate=g711ulaw
NoiseLevel=-12
ACOMLevel=22
OutSignalLevel=-12
InSignalLevel=-11
InfoActivity=1
ERLLevel=22
EchoCancellerMaxReflector=2
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
```

```
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
DSPIdentifier=1/1:1

    GENERIC:
SetupTime=1559430 ms
Index=1
PeerAddress=7702
PeerSubAddress=
PeerId=999100
PeerIfIndex=14
LogicalIfIndex=11
ConnectTime=1562020 ms
CallDuration=00:01:03 sec
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=3151
TransmitBytes=528900
ReceivePackets=3158
ReceiveBytes=503876
TELE:
ConnectionId=[0x0 0x0 0x0 0x0]
IncomingConnectionId=[0x0 0x0 0x0 0x0]
CallID=26
TxDuration=60815 ms
VoiceTxDuration=60815 ms
FaxTxDuration=0 ms
CoderTypeRate=g711ulaw
NoiseLevel=-12
ACOMLevel=28
OutSignalLevel=-12
InSignalLevel=-11
InfoActivity=1
ERLLevel=28
EchoCancellerMaxReflector=2
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
AlertTimepoint=1559430 ms
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=7701
TranslatedCallingOctet=0x0
TranslatedCalledNumber=7702
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwOutpulsedCalledNumber=7702
GwOutpulsedCalledOctet3=0x0
GwOutpulsedCallingNumber=7701
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x0
DSPIdentifier=1/1:2

    GENERIC:
SetupTime=1562040 ms
```

```

Index=1
PeerAddress=
PeerSubAddress=
PeerId=0
PeerIfIndex=0
LogicalIfIndex=0
ConnectTime=0 ms
CallDuration=00:00:00 sec
CallState=2
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=3215
TransmitBytes=512996
ReceivePackets=3208
ReceiveBytes=512812
VOIP:
ConnectionId[0x0 0x0 0x0 0x0]
IncomingConnectionId[0x0 0x0 0x0 0x0]
CallID=27
RemoteIPAddress=0.0.0.0
RemoteUDPPort=17718
RemoteSignallingIPAddress=0.0.0.0
RemoteSignallingPort=0
RemoteMediaIPAddress=10.2.6.10
RemoteMediaPort=17718
RoundTripDelay=0 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=FALSE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=other
ProtocolCallId=
SessionTarget=
OnTimeRvPlayout=60640
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=105 ms
LoWaterPlayoutDelay=105 ms
TxPakNumber=3040
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=60815
TxVoiceDuration=60815
RxPakNumber=3035
RxSignalPak=0
RxDuration=0
TxVoiceDuration=60690
VoiceRxDuration=60640
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=105
PlayDelayMin=105
PlayDelayMax=105
PlayDelayClockOffset=-1662143961

```

```
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0
PlayErrBufferOverflow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=-12
InSignalLevel=-11
LevelTxPowerMean=0
LevelRxPowerMean=-115
LevelBgNoise=0
ERLLevel=28
ACOMLevel=28
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
PlayoutMode = undefined
PlayoutInitialDelay=0 ms
ReceiveDelay=105 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = disabled
CoderTypeRate=g711ulaw
CodecBytes=160
Media Setting=flow-around

Modem passthrough signaling method is nse:
Buffer Fill Events = 0
Buffer Drain Events = 0
Percent Packet Loss = 0
Consecutive-packets-lost Events = 0
Corrected packet-loss Events = 0
Last Buffer Drain/Fill Event = 0sec
Time between Buffer Drain/Fills = Min 0sec Max 0sec

CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=

GENERIC:
SetupTime=1562040 ms
Index=2
PeerAddress=
PeerSubAddress=
PeerId=0
```

```

PeerIfIndex=0
LogicalIfIndex=0
ConnectTime=0 ms
CallDuration=00:00:00 sec
CallState=2
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=3380
TransmitBytes=540332
ReceivePackets=3386
ReceiveBytes=540356
VOIP:
ConnectionId[0x0 0x0 0x0 0x0]
IncomingConnectionId[0x0 0x0 0x0 0x0]
CallID=28
RemoteIPAddress=0.0.0.0
RemoteUDPPort=18630
RemoteSignallingIPAddress=0.0.0.0
RemoteSignallingPort=0
RemoteMediaIPAddress=10.2.6.10
RemoteMediaPort=18630
RoundTripDelay=0 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=FALSE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=other
ProtocolCallId=
SessionTarget=
OnTimeRvPayout=63120
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPayoutDelay=105 ms
LoWaterPayoutDelay=105 ms
TxPakNumber=3158
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=63165
TxVoiceDuration=63165
RxPakNumber=3164
RxSignalPak=0
RxDuration=0
TxVoiceDuration=63165
VoiceRxDuration=63120
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=105
PlayDelayMin=105
PlayDelayMax=105
PlayDelayClockOffset=957554296
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0

```

```

PlayErrBufferOverflow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=-12
InSignalLevel=-11
LevelTxPowerMean=0
LevelRxPowerMean=-114
LevelBgNoise=0
ERLLevel=22
ACOMLevel=22
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
PlayoutMode = undefined
PlayoutInitialDelay=0 ms
ReceiveDelay=105 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = disabled
CoderTypeRate=g711ulaw
CodecBytes=160
Media Setting=flow-around

Modem passthrough signaling method is nse:
Buffer Fill Events = 0
Buffer Drain Events = 0
Percent Packet Loss = 0
Consecutive-packets-lost Events = 0
Corrected packet-loss Events = 0
Last Buffer Drain/Fill Event = 0sec
Time between Buffer Drain/Fills = Min 0sec Max 0sec

CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=
Telephony call-legs: 2
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 2
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 4

```

**Step 2** Use the **show call application voice stcapp** command to display general information about the STCAPP application:

```
Router# show call application voice stcapp
```

```
App Status:           Active
CCM Status:           UP
CCM Group:            1
Registration Mode:    CCM
Total Devices:        6
Total Calls in Progress: 0
Total Call Legs in Use: 0
ROH Timeout:          45
```

**Step 3** Use the **show ccm-manager config-download** command to display information about the status of the SCCP gateway download from the Cisco CallManager:

```
Router# show ccm-manager config-download
```

```
Configuration Auto-Download Information
=====
```

```
No configurations downloaded
Current state: Waiting for commands
Configuration Download statistics:
      Download Attempted           : 1
      Download Successful           : 1
      Download Failed               : 0
      Configuration Attempted      : 1
      Configuration Successful      : 0
      Configuration Failed(Parsing): 0
      Configuration Failed(config) : 1
Last config download command:
```

```
SCCP auto-configuration status
```

```
=====
Registered with Call Manager: No
Local interface: FastEthernet0/0 (000c.8522.6910)
Current version-id: {D3A886A2-9BC9-41F8-9DB2-0E565CF51E5A}
Current config applied at: 04:44:45 EST Jan 9 2003
Gateway downloads succeeded: 1
Gateway download attempts: 1
Last gateway download attempt: 04:44:45 EST Jan 9 2003
Last successful gateway download: 04:44:45 EST Jan 9 2003
Current TFTP server: 10.2.6.101
Gateway resets: 0
Gateway restarts: 0
Managed endpoints: 6
Endpoint downloads succeeded: 6
Endpoint download attempts: 6
Last endpoint download attempt: 04:44:45 EST Jan 9 2003
Last successful endpoint download: 04:44:45 EST Jan 9 2003
Endpoint resets: 0
Endpoint restarts: 0
```

```
Configuration Error History:
sccp ccm CCM-PUB7 identifier 1
end
```

```
controller T1 2/0no shut
```

```
controller T1 2/0no shut
```

```
controller T1 2/0no shut
```

```
isdn switch-type primary-ni
end
```

- Step 4** Use the **show stcapp device voice-port** *port number* command to display detailed device-level information:

```
Router# show stcapp device voice-port 1/0/0

Port Identifier: 1/0/0
Device Type:    ALG
Device Id:      4
Device Name:    AN0C85226910200
Device State:   IS
Call State:     ACTIVE
Diagnostic:     None
Directory Number: 7701
Dial Peer(s):  999100
Last Event:     STCAPP_CC_EV_CALL_DIGIT_END_TIMEOUT
Active Call Info:
  Call Reference: 33554606
  Local IP Addr: 10.2.6.10
  Local IP Port: 17718
  Remote IP Addr: 10.2.6.10
  Remote IP Port: 18630
  Calling Number: 7701
  Called Number: 7702
  Codec:         g711ulaw
```

- Step 5** Use the **show stcapp device summary** command to display a summary of devices:

```
Router# show stcapp device summary

Total Devices:        6
Total Calls in Progress: 2
Total Call Legs in Use: 4

Port      Device      Device  Call      Dev  Directory
Identifier Name      State   State     Type  Number
-----
1/0/0     AN0C85226910200 IS      ACTIVE    ALG   7701
1/0/1     AN0C85226910201 IS      ACTIVE    ALG   7702
1/0/3     AN0C85226910203 IS      IDLE      ALG   7704
1/1/0.1   BR0C85226910280 OOS     UNREGISTERED BRI
1/1/1.1   BR0C85226910281 OOS     UNREGISTERED BRI
1/0/2     AN0C85226910202 IS      IDLE      ALG   7703
```

- Step 6** Use the **show stcapp statistics all** command to display STCAPP call statistics:

```
Router# show stcapp statistics all

STCAPP Device/Call Statistics
  OA = Origination Attempts, TA = Termination Attempts
  Err = Call Errors, PE = Call PreEmptions
Port      DevErr  CallOA  CallTA  CallErr  CallPE
-----
1/0/0.1   0        5        6        0        4
1/1/0     0        0        0        0        0
1/1/1     0        0        0        0        0
2/0/0     0        1        0        0        0
2/0/1     0        0        1        0        0
```

- Step 7** Use the **show stcapp statistics voice-port *port number*** command to display STCAPP call statistics for an individual port:

```
Router# show stcapp statistics all

STCAPP Device/Call Statistics
  OA = Origination Attempts, TA = Termination Attempts
  Err = Call Errors, PE = Call PreEmptions
Port      DevErr  Cal1OA  CallTA  CallErr  CallPE
-----
1/0/0.1      0      5      6      0      4
```

## Troubleshooting Tips

- Use the **debug ccm-manager config-download** command to display details about the Cisco CallManager download.
- Use the following commands to troubleshoot the STCAPP:
  - **debug voip application stcapp all**
  - **debug voip application stcapp error**
  - **debug voip application stcapp events**
  - **debug voip application stcapp functions**
  - **debug voip application stcapp port**
- Use the following commands to troubleshoot the SCCP protocol:
  - **debug sccp all**

### Note

Using the **debug sccp all** command generates a large volume of output. We recommend logging this information to the buffered log instead of to the console.

- **debug sccp error**
- **debug sccp events**
- **debug sccp packets**
- **debug sccp parser**

# Configuration Examples for MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways

This section provides the following configuration example:

- [MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways: Example, page 24](#)

## MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways: Example

The following example shows how to configure the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature:

```
Router# show running-config
```

```
*Jan  9 08:51:42.763: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...
```

```
Current configuration : 3502 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging buffered 25000000 debugging
!
username lab password 0 lab
no aaa new-model
!
resource manager
!
clock timezone EST -5
no network-clock-participate slot 1
no network-clock-participate slot 2
ip subnet-zero
no ip cef
!
!
no ip dhcp use vrf connected
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.18.138.14
no ip ips deny-action ips-interface
!
```

The following lines show that the STCAPP is enabled. The Cisco CallManager group number must match the Cisco CallManager identifier configured for the SPI using the **sccp ccm-group** command.

```
stcapp ccm-group 1
stcapp
!
no ftp-server write-enable
```

The next line shows the ISDN switch type configuration for BRI voice ports.

```
isdn switch-type basic-ni
voice-card 1
no dspfarm
!
voice-card 2
dspfarm
!
!
voice service voip
```

The next line shows modem pass-through configuration:

```
modem passthrough nse codec g711ulaw
!
!
interface FastEthernet0/0
```

```

ip address 10.2.6.10 255.255.255.0
duplex auto
speed auto
no clns route-cache
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
no clns route-cache
!

```

The following lines show the required configuration for the ISDN BRI interface:

```

interface BRI1/0
no ip address
isdn switch-type basic-ni

```

The following **isdn timer** command is not user-configured. The command is automatically generated based on the presence of the **isdn switch-type basic ni** and **isdn protocol-emulate network** commands:

```

isdn timer t309 30000
isdn overlap-receiving T302 16000
isdn protocol-emulate network
isdn point-to-point-setup
isdn layer1-emulate network
isdn spid1 7705
isdn incoming-voice voice

```

The next line shows automatically enabled support for ISDN Calling Name display information.

```

isdn supp-service name calling
isdn sending-complete

```

The following **isdn skipsend-idverify** command is not user-configured. The command is automatically generated based on the presence of the **isdn switch-type basic ni** and **isdn protocol-emulate network** commands:

```

isdn skipsend-idverify
line-power
no clns route-cache
!
interface BRI1/1
no ip address
isdn switch-type basic-net3
isdn overlap-receiving T302 16000
isdn protocol-emulate network
isdn layer1-emulate network
isdn incoming-voice voice
isdn supp-service name calling
isdn skipsend-idverify
no clns route-cache
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
!
!
ip http server
no ip http secure-server
!
ip access-list extended jeff
!
access-list 1 permit 10.6.6.31
access-list 107 deny ip host 10.6.6.20 any log
access-list 107 permit ip any any

```

```

!
!
control-plane
!
!
!
voice-port 1/0/0
timeouts initial 60
timeouts interdigit 60
!
voice-port 1/0/1
timeouts initial 60
!
voice-port 1/0/2
timeouts initial 60
timeouts interdigit 60
!
voice-port 1/0/3
timeouts initial 60
!
voice-port 1/1/0
timeouts initial 60
!
voice-port 1/1/1
compand-type a-law
timeouts initial 60
bearer-cap Speech
!
ccm-manager music-on-hold

```

The following lines show the Cisco CallManager configuration for automatic download capability of XML dial-peer configuration files. The IP address should match the address of the TFTP server from which to download XML files. The SCCP interface should match the interface specified for Cisco CallManager registration.

```

ccm-manager config server 10.6.6.31
ccm-manager sccp local FastEthernet0/0
ccm-manager sccp
!
!

```

The following lines show SCCP configuration. The IP address should match the address of the Cisco CallManager server.

```

sccp local FastEthernet0/0
sccp ccm 10.2.6.101 identifier 2
sccp ccm 10.2.6.100 identifier 1
sccp ip precedence 1
sccp

```

The following SCCP Cisco CallManager group number must match the STCAPP Cisco CallManager group number specified using the **stcapp ccm-group** command:

```

sccp ccm group 1
associate ccm 2 priority 2
associate ccm 1 priority 1
!
!

```

The following lines show STCAPP-controlled dial peers. By default, Cisco CallManager autoconfigured STCAPP dial peers begin with the prefix 999.

```

dial-peer voice 999100 pots
service stcapp
port 1/0/0
!

```

## Additional References

```

dial-peer voice 999101 pots
service stcapp
port 1/0/1
!
dial-peer voice 999103 pots
service stcapp
port 1/0/3
!
dial-peer voice 999110 pots
service stcapp
port 1/1/0
!
dial-peer voice 999111 pots
service stcapp
port 1/1/1
!
dial-peer voice 999102 pots
service stcapp
port 1/0/2
!
!
line con 0
exec-timeout 0 0
length 0
line aux 0
line vty 0 4
password cisco
login
!
!
end

```

## Additional References

The following sections provide references related to the MLPP for Analog and BRI Endpoints on Cisco IOS Voice Gateways feature.

## Related Documents

Related Topic	Document Title
Cisco CallManager interoperability	<a href="#">Cisco Unified CallManager and Cisco IOS Interoperability Guide</a>
Cisco CallManager administration	<a href="#">Cisco CallManager Administration Guide</a> , Release 4.1(2)
Cisco CallManager security configuration	<a href="#">Cisco CallManager Security Guide</a> , Release 4.1.2
Cisco IOS SRST	<a href="#">Cisco IOS Survivable Remote Site Telephony Version 3.3 System Administrator Guide</a> <a href="#">Cisco IOS Survivable Remote Site Telephony Version 3.3 Command Reference</a>
ISDN configuration	<a href="#">Cisco IOS ISDN Voice Configuration Guide</a> , Release 12.3, “Basic ISDN Voice-Interface Configuration” chapter
Cisco IOS voice configuration	<a href="#">Cisco IOS Voice Configuration Library</a>
Cisco IOS voice command reference	<a href="#">Cisco IOS Voice Command Reference</a> , Release 12.4T

Related Topic	Document Title
STC application and Cisco CallManager controlled secure endpoints	<a href="#">Secure Communication Between IP-STE Endpoint and Line-Side STE Endpoint</a>
SCCP gateway controlled supplementary features	<a href="#">SCCP Controlled Analog (FXS) Ports with Supplementary Features in Cisco IOS Gateways</a>

## Standards

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

## MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

## RFCs

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

## Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	<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 only.

- [ccm-manager scep](#)
- [ccm-manager scep local](#)
- [debug ccm-manager](#)
- [debug voip application stcapp all](#)
- [debug voip application stcapp error](#)
- [debug voip application stcapp events](#)
- [debug voip application stcapp functions](#)
- [debug voip application stcapp port](#)
- [debug voip dcapi](#)
- [show call active voice](#)
- [show call application voice](#)
- [show ccm-manager](#)
- [show stcapp device](#)
- [show stcapp statistics](#)
- [stcapp](#)
- [stcapp ccm-group](#)
- [stcapp timer](#)

## ccm-manager sccp

To enable Cisco CallManager auto configuration of the Cisco IOS gateway, use the **ccm manager sccp** command in global configuration mode. To disable auto configuration, use the **no** form of this command.

**ccm-manager sccp**

**no ccm-manager sccp**

**Syntax Description** This command has no arguments or keywords.

**Defaults** Auto configuration is disabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use this command to trigger TFTP download of the eXtensible Markup Language (XML) configuration file. Issuing this command immediately triggers the download, and also enables the Skinny Client Control Protocol (SCCP) and SCCP Telephony Control Application (STCAPP), applications that enable Cisco CallManager control of gateway-connected telephony endpoints.

**Examples** The following example enables autoconfiguration of gateway-connected endpoints:

```
Router(config)# ccm-manager sccp
```

Related Commands	Command	Description
	<b>ccm-manager config</b>	Specifies the TFTP server from which the Cisco IOS gateway downloads Cisco CallManager XML configuration files.
	<b>ccm-manager sccp local</b>	Selects the local interface for SCCP application use for Cisco CallManager registration.
	<b>show ccm-manager config-download</b>	Displays information about the status of the Cisco IOS gateway configuration download.

## ccm-manager sccp local

To select the local interface that the Skinny Client Control Protocol (SCCP) application uses to register with Cisco CallManager, use the **ccm- manager sccp local** command in global configuration mode. To deselect the interface, use the **no** form of this command.

**ccm-manager sccp local** *interface-type interface-number*

**no ccm-manager sccp local** *interface-type interface-number*

### Syntax Description

<i>interface-type</i>	Interface type that the SCCP application uses for Cisco CallManager registration.
<i>interface-number</i>	Interface number that the SCCP application uses for Cisco CallManager registration.

### Defaults

No local interface is selected.

### Command Modes

Global configuration

### Command History

Release	Modification
12.3(14)T	This command was introduced.

### Usage Guidelines

You must specify this interface before enabling the Cisco CallManager auto configuration process. The MAC address of this interface is used to identify gateway endpoints.

### Examples

The following example configures a FastEthernet interface for SCCP application use for Cisco CallManager registration:

```
Router(config)# ccm-manager sccp local fastethernet0/0
```

### Related Commands

Command	Description
<b>show ccm-manager config-download</b>	Displays information about the status of the Cisco IOS gateway configuration download.

# debug ccm-manager

To display debugging information about the Cisco CallManager, use the **debug ccm-manager** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ccm-manager {backhaul {events | errors} | config-download {all | errors | events |
packets | xml} | errors | events | music-on-hold {errors | events | packets} | packets}
```

```
no debug ccm-manager
```

## Syntax Description

<b>backhaul</b>	(Optional) Enables debugging of the Cisco CallManager backhaul. The keywords are as follows: <ul style="list-style-type: none"> <li><b>events</b>—Displays Cisco CallManager backhaul events.</li> <li><b>errors</b>—Displays Cisco CallManager backhaul errors.</li> </ul>
<b>config-download</b>	Enables debugging of the Cisco CallManager configuration download. The keywords are as follows: <ul style="list-style-type: none"> <li><b>all</b>—Displays all Cisco CallManager configuration parameters.</li> <li><b>errors</b>—Displays Cisco CallManager configuration errors.</li> <li><b>events</b>—Displays Cisco CallManager configuration events.</li> <li><b>packets</b>—Displays Cisco CallManager configuration packets.</li> <li><b>xml</b>—Displays the Cisco CallManager configuration eXtensible Markup Language (XML) parser.</li> </ul>
<b>errors</b>	(Optional) Displays errors related to Cisco CallManager.
<b>events</b>	(Optional) Displays Cisco CallManager events, such as when the primary Cisco CallManager server fails and control is switched to the backup Cisco CallManager server.
<b>music-on-hold</b>	(Optional) Displays music-on-hold (MOH). The keywords are as follows: <ul style="list-style-type: none"> <li><b>errors</b>—Displays MOH errors.</li> <li><b>events</b>—Displays MOH events.</li> <li><b>packets</b>—Displays MOH packets.</li> </ul>
<b>packets</b>	(Optional) Displays Cisco CallManager packets.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(3)T	This command was introduced for Cisco CallManager Version 3.0 and the Cisco Voice Gateway 200 (Cisco VG200).
12.2(2)XA	This command was implemented on Cisco 2600 series and Cisco 3600 series routers.

Release	Modification
12.2(2)XN	Support for enhanced Media Gateway Control Protocol (MGCP) voice gateway interoperability was added to Cisco CallManager Version 3.1 for the Cisco 2600 series routers, Cisco 3600 series routers, and the Cisco Voice Gateway 200 (Cisco VG200).
12.2(11)T	This command was implemented on the Cisco IAD2420 series.
12.3(14)T	New output was added relating to the SCCP protocol.

## Examples

The following is sample output from the **debug ccm-manager events** command:

```
Router# debug ccm-manager events

*Feb 28 22:56:05.873: cmapp_mgcpapp_go_down: Setting mgc status to NO_RESPONSE
*Feb 28 22:56:05.873: cmapp_host_fsm: New state DOWN for host 0 (172.20.71.38)
*Feb 28 22:56:05.873: cmapp_mgr_process_ev_active_host_failed: Active host 0
(172.20.71.38) failed
*Feb 28 22:56:05.873: cmapp_mgr_check_hostlist: Active host is 0 (172.20.71.38)
*Feb 28 22:56:05.877: cmapp_mgr_switchover: New actv host will be 1 (172.20.71.44)
*Feb 28 22:56:05.877: cmapp_host_fsm: Processing event GO_STANDBY for host 0
(172.20.71.38) in state DOWN
*Feb 28 22:56:05.877: cmapp_open_new_link: Open link for [0]:172.20.71.38
*Feb 28 22:56:05.877: cmbh_open_tcp_link: Opening TCP link with Rem IP 172.20.71.38, Local
IP 172.20.71.19, port 2428
*Feb 28 22:56:05.881: cmapp_open_new_link: Open initiated OK: Host 0 (172.20.71.38),
session_id=8186DEE4
*Feb 28 22:56:05.881: cmapp_start_open_link_tmr: Host 0 (172.20.71.38), tmr 0
*Feb 28 22:56:05.881: cmapp_host_fsm: New state STANDBY_OPENING for host 0 (172.20.71.38)
*Feb 28 22:56:05.881: cmapp_host_fsm: Processing event GO_ACTIVE for host 1 (172.20.71.44)
in state STANDBY_READY
*Feb 28 22:56:05.885: cmapp_mgr_send_rehome: new addr=172.20.71.44,port=2427
*Feb 28 22:56:05.885: cmapp_host_fsm: New state REGISTERING for host 1 (172.20.71.44)
```

The following is sample output from the **debug ccm-manager config-download all** command for an error case in which the configuration file cannot be accessed for SCCP download:

```
*Jan 9 07:28:33.499: cmapp_xml_process_timer:
*Jan 9 07:28:33.499: cmapp_xml_find_ep_by_name: Checking for ep_name [*]
*Jan 9 07:28:33.499: cmapp_xml_exec_fsm: Endpoint is [*]
*Jan 9 07:28:33.499: cmapp_xml_exec_fsm: endpoint = * state = CMAPP_XML_FILE_DNLD, event
= CMAPP_XML_EVT_FILE_DNLD_TIMER
*Jan 9 07:28:33.499: cmapp_xml_file_retry_timer_expired: state = CMAPP_XML_FILE_DNLD,
event = CMAPP_XML_EVT_FILE_DNLD_TIMER
*Jan 9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp://10.6.6.31/Router.cisco.com.cnf.xml, rc=-2
*Jan 9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp://10.6.6.31/Router.cisco.com.cnf.xml, len = 0
*Jan 9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp:///Router.cisco.com.cnf.xml, rc=-2
*Jan 9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp:///Router.cisco.com.cnf.xml, len = 0
*Jan 9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp:///Router.cisco.com.cnf.xml, rc=-2
*Jan 9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp:///Router.cisco.com.cnf.xml, len = 0
*Jan 9 07:29:14.499: cmapp_xml_exec_fsm: New state = CMAPP_XML_FILE_DNLD, ep = 6544CFA8
```

The following is sample output from the **debug ccm-manager config-download all** command for a successful SCCP download:

```

*Jan  9 09:44:45.543: cmapp_sccp_config:
*Jan  9 09:44:45.543: cmapp_sccp_reset_curcfg:
*Jan  9 09:44:45.543: cmapp_sccp_init_curcfg:
*Jan  9 09:44:45.543: cmapp_sccp_download_gw_config_file:
*Jan  9 09:44:45.543: cmapp_sccp_get_gw_name:
*Jan  9 09:44:45.543: cmapp_sccp_get_gw_name: XML file name
generated->SKIGW0C85226910.cnf.xml
*Jan  9 09:44:45.543: cmapp_sccp_get_xml_file_via_tftp:
*Jan  9 09:44:45.543: cmapp_sccp_tftp_download_file:
*Jan  9 09:44:45.543: cmapp_sccp_tftp_get_file_size:
*Jan  9 09:44:45.563: cmapp_sccp_get_buffer:
*Jan  9 09:44:45.575: cmapp_sccp_tftp_download_file: File
(tftp://10.2.6.101/SKIGW0C85226910.cnf.xml) read 8162 bytes
*Jan  9 09:44:45.575: cmapp_sccp_get_xml_file_via_tftp: Read file
tftp://10.2.6.101/SKIGW0C85226910.cnf.xml, len = 8162
*Jan  9 09:44:45.575: cmapp_parse_gw_xml_file:
*Jan  9 09:44:45.579: cmapp_sccp_gw_chardata_handler: ccm found, priority=0

```

The following lines show the conversion of XML data into router configuration information for the endpoint:

```

*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Unit has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Subunit has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 2
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 3
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Subunit has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Unit has been set to 2

```

Table 2 describes the significant fields shown in the displays.

**Table 2** *debug ccm-manager Field Descriptions*

Field	Description
<i>nn:nn:nn:</i>	Time stamp time in hours (military format), minutes, and seconds that indicates when the Cisco CallManager event occurred.
<i>cmapp_ error message:</i>	The Cisco CallManager routine in which the error event occurred.

#### Related Commands

Command	Description
<b>show ccm-manager</b>	Displays a list of Cisco CallManager servers, their current status, and their availability.

# debug voip application stcapp all

To display debugging information for the components of the SCCP Telephony Control Application (STCAPP), use the **debug voip application stcapp all** command in privileged EXEC mode. To disable debugging display, use the **no** form of this command.

**debug voip application stcapp all**

**no debug voip application stcapp all**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.
	12.4(4)T	Command output was enhanced to display codec capabilities for modem transport.
	12.4(6)XE	Command output was enhanced to display fax relay, RFC 2833 DTMF digit relay, dial tone after remote onhook, call control feature mode and visual message waiting indicator (VMWI) information for SCCP analog ports.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

**Usage Guidelines** The **debug voip application stcapp all** command provides debugging output for all the STCAPP debug commands compiled into one display.

**Examples** The following is sample output from the **debug voip application stcapp all** command for a Cisco VG 224 voice gateway in Cisco IOS Release 12.4(6)XE showing call control feature mode messages for the drop last active call feature. Port 2/0 calls port 2/1, performs a hook flash to get dial tone while port 2/1 is on hold, and calls port 2/3. Ports 2/0 and 2/3 are active, while port 2/1 is on hold.

```
Router# debug voip application stcapp all
```

Port 2/0 performs a hook flash to activate the drop last call feature.

```
Mar 3 20:41:07.022: 2/0 : stcapp_screen_api_event
Mar 3 20:41:07.022: 2/0 : event:STCAPP_CC_EV_CALL_FEATURE_HOOKFLASH received.
Mar 3 20:41:07.022: 2/0 : stcapp_screen_call_feature_hookflash
Mar 3 20:41:07.022: 2/0 : lcb->num_ccbs=2, lcb->mode=CALL_TRANSFER(1),
lcb->state=ACTIVE (8)
Mar 3 20:41:07.022: 2/0 : lcb->fm_mode=CALL_FM_NONE(0),
Mar 3 20:41:07.022: 2/0 : ACTIVE call_state=5, lcb->mode=CALL_TRANSFER(1),
lcb->state=8
Mar 3 20:41:07.022: 2/0 : STANDBY call_state=8, lcb->mode=CALL_TRANSFER(1),
lcb->state=8
Mar 3 20:41:07.022: 2/0 : enter FM: set event id to STCAPP_EV_FEATURE_MODE(143)
Mar 3 20:41:07.022: 2/0 : ==> Received event:STCAPP_EV_FEATURE_MODE for CallId: 63
Mar 3 20:41:07.022: 2/0 : Call State:ACTIVE
```

```
Mar 3 20:41:07.022: 2/0 : stcapp_feature_mode_ah
```

The following line shows a message to generate feature tone.

```
Mar 3 20:41:07.022: 2/0 : Sending ccGenerateTone(2048(0x800)):FEATURE tone
Mar 3 20:41:07.022: 2/0 : set lcb->fm_mode = CALL_FM_TRANSF_FT_ON (4)
Mar 3 20:41:07.022: 2/0 : Sending ccCallReportDigits
Mar 3 20:41:07.022: 2/0 : New State = FM_DIGIT_COLLECT
```

The following lines show port 2/0 dialing the feature access code (FAC) #1 to drop the last active call.

```
Mar 3 20:41:11.234: htsp_digit_ready(2/0): digit = #
Mar 3 20:41:11.234: STCAPP:Receive CC event:: call_id=63, ccb=0x64A5059C
Mar 3 20:41:11.234: 2/0 : ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END for CallId: 63
Mar 3 20:41:11.234: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 20:41:11.234: 2/0 : stcapp_fm_dc_digit_end_ah
Mar 3 20:41:11.234: 2/0 : Digit received is (#)
Mar 3 20:41:11.234: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_ON(4)
Mar 3 20:41:11.234: 2/0 : Sending ccGenerateTone(0x0)
Mar 3 20:41:11.234: 2/0 : set lcb->fm_mode to CALL_FM_TRANSF_FT_OFF (5)
Mar 3 20:41:11.234: 2/0 : So far the fm feature code =#, fm_string_idx=1
Mar 3 20:41:11.238: 2/0 : No state change
Mar 3 20:41:12.346: htsp_digit_ready(2/0): digit = 1
Mar 3 20:41:12.346: STCAPP:Receive CC event:: call_id=63, ccb=0x64A5059C
Mar 3 20:41:12.346: 2/0 : ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END for CallId: 63
Mar 3 20:41:12.346: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 20:41:12.346: 2/0 : stcapp_fm_dc_digit_end_ah
Mar 3 20:41:12.346: 2/0 : Digit received is (1)
Mar 3 20:41:12.346: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_OFF(5)
Mar 3 20:41:12.346: 2/0 : So far the fm feature code =#1, fm_string_idx=2
```

The following lines show the SCCP gateway processing FAC #1 messages.

```
Mar 3 20:41:12.346: 2/0 : stcapp_handle_fm_feature_id
Mar 3 20:41:12.346: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_OFF(5),
Mar 3 20:41:12.346: 2/0 : STANDBY call_state=8
Mar 3 20:41:12.346: 2/0 : stcapp_send_softkey_event
Mar 3 20:41:12.346: 2/0 : Sending dcDeviceSoftKeyEvent(EndCall: event=9) for device
id: 1, call_ref: 16810780
Mar 3 20:41:12.346: 2/0 : stcapp_send_softkey_event
Mar 3 20:41:12.346: 2/0 : Sending dcDeviceSoftKeyEvent(Resume: event=10) for device
id: 1, call_ref: 16810777
Mar 3 20:41:12.346: 2/0 : set lcb->fm_mode to CALL_FM_NONE (0)
Mar 3 20:41:12.346: 2/0 : No state change
Mar 3 20:41:12.366: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
Mar 3 20:41:12.366: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 20:41:12.366: 2/0 : stcapp_close_rcv_chnl_ah
Mar 3 20:41:12.366: 2/0 : stcapp_disconnect_call_leg
Mar 3 20:41:12.366: 2/0 : Sending ccCallDisconnect for VoIP_LEG with call id:70
Mar 3 20:41:12.366: 2/0 : No state change
Mar 3 20:41:12.370: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
Mar 3 20:41:12.370: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 20:41:12.370: 2/0 : stcapp_active_close_xmt_chnl_ah
Mar 3 20:41:12.370: 2/0 : New State = ONHOOK_PEND
```

The following is sample output from the **debug voip application stcapp all** command for a Cisco VG 224 voice gateway in Cisco IOS Release 12.4(6)XE showing call control feature mode messages for the call transfer feature:

```
Router# debug voip application stcapp all
```

```
Mar 3 21:00:56.014: 2/0 : stcapp_handle_fm_feature_id
```

## debug voip application stcapp all

```

Mar 3 21:00:56.014: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_OFF(5),
Mar 3 21:00:56.014: 2/0 : set lcb->mode to CALL_TRANSFER (1)
Mar 3 21:00:56.014: 2/0 : stcapp_send_softkey_event
Mar 3 21:00:56.014: 2/0 : Sending dcDeviceSoftKeyEvent(Transfer: event=4) for
device id: 1, call_ref: 16810789
Mar 3 21:00:56.014: 2/0 : set lcb->fm_mode to CALL_FM_NONE (0)
Mar 3 21:00:56.014: 2/0 : No state change
Mar 3 21:00:56.034: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
Mar 3 21:00:56.034: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:00:56.038: 2/0 : stcapp_close_rcv_chnl_eh
Mar 3 21:00:56.038: 2/0 : stcapp_disconnect_call_leg
Mar 3 21:00:56.038: 2/0 : No state change
Mar 3 21:00:56.042: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
Mar 3 21:00:56.042: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:00:56.042: 2/0 : stcapp_active_close_xmt_chnl_eh
Mar 3 21:00:56.042: 2/0 : New State = ONHOOK_PEND

```

The following is sample output in Cisco IOS 12.4(6)XE from the **debug voip application stcapp all** command showing call control feature mode messages for the call conference feature:

```
Router# debug voip application stcapp all
```

```

Mar 3 21:18:54.258: 2/0 : stcapp_handle_fm_feature_id
Mar 3 21:18:54.258: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_OFF(5),
Mar 3 21:18:54.258: 2/0 : stcapp_send_softkey_event
Mar 3 21:18:54.258: 2/0 : Sending dcDeviceSoftKeyEvent(Conference: event=13) for
device id: 1, call_ref: 16810798
Mar 3 21:18:54.258: 2/0 : set lcb->fm_mode to CALL_FM_NONE (0)
Mar 3 21:18:54.258: 2/0 : No state change
Mar 3 21:18:54.298: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
Mar 3 21:18:54.298: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:18:54.298: 2/0 : stcapp_close_rcv_chnl_eh
Mar 3 21:18:54.298: 2/0 : stcapp_disconnect_call_leg
Mar 3 21:18:54.298: 2/0 : No state change
Mar 3 21:18:54.302: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
Mar 3 21:18:54.302: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:18:54.302: 2/0 : stcapp_active_close_xmt_chnl_eh
Mar 3 21:18:54.302: 2/0 : New State = ONHOOK_PEND

```

The following is sample output in Cisco IOS 12.4(6)XE from the **debug voip application stcapp all** command showing call control feature mode messages for the drop last conferee feature:

```
Router# debug voip application stcapp all
```

```

Mar 3 21:27:05.170: 2/0 : stcapp_handle_fm_feature_id
Mar 3 21:27:05.170: 2/0 : lcb->fm_mode = CALL_FM_CONF_FT_OFF(7),
Mar 3 21:27:05.170: 2/0 : stcapp_send_softkey_event
Mar 3 21:27:05.170: 2/0 : Sending dcDeviceSoftKeyEvent(DropLastConferee: event=19)
for device id: 1, call_ref: 16810795
Mar 3 21:27:05.170: 2/0 : set lcb->fm_mode to CALL_FM_NONE (0)
Mar 3 21:27:05.170: 2/0 : No state change
Mar 3 21:27:05.194: 2/0 : ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
Mar 3 21:27:05.194: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:27:05.194: 2/0 : stcapp_conn_call_info_eh
Mar 3 21:27:05.194: 2/0 : No state change
Mar 3 21:27:05.194: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
Mar 3 21:27:05.194: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:27:05.194: 2/0 : stcapp_close_rcv_chnl_eh
Mar 3 21:27:05.198: 2/0 : stcapp_disconnect_call_leg
Mar 3 21:27:05.198: 2/0 : No state change
Mar 3 21:27:05.198: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
Mar 3 21:27:05.198: 2/0 : Call State:FM_DIGIT_COLLECT

```

```
Mar 3 21:27:05.202: 2/0 : stcapp_active_close_xmt_chnl_eh
Mar 3 21:27:05.202: 2/0 : New State = ONHOOK_PEND
```

The following is sample output in Cisco IOS 12.4(6)XE from the **debug voip application stcapp all** command showing call control feature mode messages for the toggle feature:

```
Router# debug voip application stcapp all
```

```
Mar 3 21:37:11.650: 2/0 : stcapp_handle_fm_feature_id
Mar 3 21:37:11.650: 2/0 : lcb->fm_mode = CALL_FM_TRANSF_FT_OFF(5),
Mar 3 21:37:11.650: 2/0 : STANDBY call_state=8
Mar 3 21:37:11.650: 2/0 : stcapp_send_softkey_event
Mar 3 21:37:11.650: 2/0 : Sending dcDeviceSoftKeyEvent(Hold: event=3) for device
id: 1, call_ref: 16810811
Mar 3 21:37:11.650: 2/0 : stcapp_send_softkey_event
Mar 3 21:37:11.650: 2/0 : Sending dcDeviceSoftKeyEvent(Resume: event=10) for device
id: 1, call_ref: 16810808
Mar 3 21:37:11.650: 2/0 : set new lcb->fm_mode=CALL_FM_NONE (0)
Mar 3 21:37:11.650: 2/0 : in toggle: (1)
Mar 3 21:37:11.650: 2/0 : No state change
Mar 3 21:37:11.654: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
Mar 3 21:37:11.658: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:37:11.658: 2/0 : stcapp_close_rcv_chnl_eh
Mar 3 21:37:11.658: 2/0 : stcapp_disconnect_call_leg
Mar 3 21:37:11.658: 2/0 : No state change
Mar 3 21:37:11.674: 2/0 : ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
Mar 3 21:37:11.674: 2/0 : Call State:FM_DIGIT_COLLECT
Mar 3 21:37:11.674: 2/0 : stcapp_active_close_xmt_chnl_eh
Mar 3 21:37:11.674: 2/0 : New State = ONHOOK_PEND
```

The following is sample output from the **debug voip application stcapp all** command for a Cisco VG 224 voice gateway in Cisco IOS Release 12.4(6)XE showing T.38 fax relay messages:

```
Router# debug voip application stcapp all
```

```
01:10:40: //68/xxxxxxxxxxxxx/CCAPI/cc_api_set_fax_mode:
Destination Interface=0x464EDDFC, Destination Call Id=68, Source Call Id=67
01:10:40: [1185752428]CNFSM: new_container:fax_t38_container
01:10:40: [1185752428]CNFSM: next_state:S_DSMP_GW_FAX_T38_CONNECTED
01:10:40: STCAPP:Receive CC event:: call_id=68, ccb=0x46B264C4
01:10:40: 1/0/0: ==> Received event:STCAPP_CC_EV_CALL_FEATURE_T38_CODEEC for CallId: 68
01:10:40: 1/0/0: Call State:ACTIVE
01:10:40: 1/0/0: stcapp_t38_remote_codec_dnld_done_eh
01:10:40: //68/xxxxxxxxxxxxx/CCAPI/ccCallFeature:
Feature Type=39, Call Id=68
01:10:40: //67/xxxxxxxxxxxxx/CCAPI/cc_api_remote_codec_dnld_done:
Destination Interface=0x4631148C, Destination Call Id=67, Source Call Id=68, Xmit
Function=0x4230E4D4

01:10:40: //-1/xxxxxxxxxxxxx/DSM:():-1/dsp_stream_mgr_remote_dnld_done:
```

The following is sample output from the **debug voip application stcapp all** command for a voice gateway in Cisco IOS Release 12.4(4)T showing device modem transport capability:

```
Router# debug voip application stcapp all
```

```
01:10:40: //68/xxxxxxxxxxxxx/CCAPI/cc_api_set_fax_mode:
Destination Interface=0x464EDDFC, Destination Call Id=68, Source Call Id=67
01:10:40: [1185752428]CNFSM: new_container:fax_t38_container
01:10:40: [1185752428]CNFSM: next_state:S_DSMP_GW_FAX_T38_CONNECTED
01:10:40: STCAPP:Receive CC event:: call_id=68, ccb=0x46B264C4
01:10:40: 1/0/0: ==> Received event:STCAPP_CC_EV_CALL_FEATURE_T38_CODEEC for CallId: 68
```

## debug voip application stcapp all

```

01:10:40: 1/0/0:      Call State:ACTIVE
01:10:40: 1/0/0:      stcapp_t38_remote_codec_dnld_done_eh
01:10:40: //68/xxxxxxxxxxxx/CCAPI/ccCallFeature:
      Feature Type=39, Call Id=68
01:10:40: //67/xxxxxxxxxxxx/CCAPI/cc_api_remote_codec_dnld_done:
      Destination Interface=0x4631148C, Destination Call Id=67, Source Call Id=68, Xmit
      Function=0x4230E4D4

01:10:40: //-1/xxxxxxxxxxxx/DSM:():-1/dsp_stream_mgr_remote_dnld_done:

```

The following is sample output from the **debug voip application stcapp all** command for a voice gateway in Cisco IOS Release 12.4(4)T showing modem transport capability:

```

Router# debug voip application stcapp all

*Jan 11 12:24:18.443: stcapp_start
*Jan 11 12:24:18.443:      stcapp process started
*Jan 11 12:24:18.443: stcapp_init_symphony
*Jan 11 12:24:18.443:      CCAPI successfully initialized
*Jan 11 12:24:18.443: stcapp_init_rtp
*Jan 11 12:24:18.443: stcapp_vp_shut
*Jan 11 12:24:18.443: stcapp_port_up_down
*Jan 11 12:24:18.443:      RTP successfully brought in service
*Jan 11 12:24:18.443: stcapp_create_dcbs_from_dialpeers
*Jan 11 12:24:18.447: 1/1/0: stcapp_create_device
*Jan 11 12:24:18.447: 1/1/0:      Endpoint base name generated->AN0D65D8DD40280
*Jan 11 12:24:18.447: 1/1/0:      New dialpeer id: 999110
*Jan 11 12:24:18.447: 1/1/0:      Analog device is ready to be registered

```

The following lines show the codec subtype, which indicates the modem transport method; 0=None, 1=V.150.1 (modem relay), and 2=VBD (modem pass-through):

```

*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=5 (g711ulaw) subtype=2
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=1 (g729ar8) subtype=2
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=5 (g711ulaw) subtype=1
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=1 (g729ar8) subtype=1
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=5 (g711ulaw) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=6 (g711alaw) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=1 (g729ar8) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=2 (g726r16) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=3 (g726r24) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=4 (g726r32) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=7 (g728) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=8 (g723r63) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=9 (g723r53) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=12 (g729br8) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=14 (g723ar63) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      reg caps including codec=15 (g723ar53) subtype=0
*Jan 11 12:24:18.447: 1/1/0:      Device: AN0D65D8DD40280 Id: 7 successfully registered
with CM
*Jan 11 12:24:18.455: ==> Received event:STCAPP_DC_EV_DEVICE_REGISTER_DONE
*Jan 11 12:24:18.455: 1/1/0:      Device State:OOS
*Jan 11 12:24:18.455: 1/1/0: stcapp_dev_default_eh
*Jan 11 12:24:18.455: 1/1/0:      New State = INIT
*Jan 11 12:24:18.455: ==> Received event:STCAPP_DC_EV_DEVICE_CAP_REQ
*Jan 11 12:24:18.455: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.455: 1/1/0: stcapp_cap_req_eh
*Jan 11 12:24:18.455: 1/1/0:      Sending dcDeviceHeadsetStatus for devID:7
*Jan 11 12:24:18.455: 1/1/0:      Sending dcDeviceButtonTemplateReq for devID:7
*Jan 11 12:24:18.455: 1/1/0:      No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_BUTTON_TEMP_RES
*Jan 11 12:24:18.647: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.647: 1/1/0: stcapp_button_templ_res_eh

```

```

*Jan 11 12:24:18.647: 1/1/0:      Sending dcDeviceLineStatReq for devID:7
*Jan 11 12:24:18.647: 1/1/0:      No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_FORWARD_STAT_RES
*Jan 11 12:24:18.647: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.647: 1/1/0:      stcapp_forward_stat_res_eh
*Jan 11 12:24:18.647: 1/1/0:      lineNumber: 1
*Jan 11 12:24:18.647: 1/1/0:      forwardAllActive: 0
*Jan 11 12:24:18.647: 1/1/0:      forwardBusyActive: 0
*Jan 11 12:24:18.647: 1/1/0:      forwardNoAnswerActive: 0
*Jan 11 12:24:18.651: 1/1/0:      ForwardAllDirNumber:
*Jan 11 12:24:18.651: 1/1/0:      No state change
*Jan 11 12:24:18.651: ==> Received event:STCAPP_DC_EV_DEVICE_LINE_STAT_RES
*Jan 11 12:24:18.651: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.455: 1/1/0:      stcapp_cap_req_eh
*Jan 11 12:24:18.455: 1/1/0:      Sending dcDeviceHeadsetStatus for devID:7
*Jan 11 12:24:18.455: 1/1/0:      Sending dcDeviceButtonTemplateReq for devID:7
*Jan 11 12:24:18.455: 1/1/0:      No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_BUTTON_TEMP_RES
*Jan 11 12:24:18.647: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.647: 1/1/0:      stcapp_button_templ_res_eh
*Jan 11 12:24:18.647: 1/1/0:      Sending dcDeviceLineStatReq for devID:7
*Jan 11 12:24:18.647: 1/1/0:      No state change
*Jan 11 12:24:18.647: ==> Received event:STCAPP_DC_EV_DEVICE_FORWARD_STAT_RES
*Jan 11 12:24:18.647: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.647: 1/1/0:      stcapp_forward_stat_res_eh
*Jan 11 12:24:18.647: 1/1/0:      lineNumber: 1
*Jan 11 12:24:18.647: 1/1/0:      forwardAllActive: 0
*Jan 11 12:24:18.647: 1/1/0:      forwardBusyActive: 0
*Jan 11 12:24:18.647: 1/1/0:      forwardNoAnswerActive: 0
*Jan 11 12:24:18.651: 1/1/0:      ForwardAllDirNumber:
*Jan 11 12:24:18.651: 1/1/0:      No state change
*Jan 11 12:24:18.651: ==> Received event:STCAPP_DC_EV_DEVICE_LINE_STAT_RES
*Jan 11 12:24:18.651: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.651: 1/1/0:      stcapp_line_stat_eh
*Jan 11 12:24:18.651: 1/1/0:      lineNumber: 1
*Jan 11 12:24:18.651: 1/1/0:      lineDirNumber: 5902
*Jan 11 12:24:18.651: 1/1/0:      display name: 5902
*Jan 11 12:24:18.651: 1/1/0:      Sending dcDeviceRegAvailableLines for devID:7
*Jan 11 12:24:18.651: 1/1/0:      Sending dcDeviceDateTimeReq for devID:7
*Jan 11 12:24:18.651: 1/1/0:      No state change
*Jan 11 12:24:18.823: ==> Received event:STCAPP_DC_EV_DEVICE_DEFINE_DATE_TIME_RES
*Jan 11 12:24:18.827: 1/1/0:      Device State:INIT
*Jan 11 12:24:18.827: 1/1/0:      stcapp_define_date_time_eh
*Jan 11 12:24:18.827: 1/1/0:      New State = IS
*Jan 11 12:24:18.827: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:24:18.827: 1/1/0:      Device State:IS
*Jan 11 12:24:18.827: 1/1/0:      stcapp_display_prompt_status_eh
*Jan 11 12:24:18.827: 1/1/0:      lineNumber: 0
*Jan 11 12:24:18.827: 1/1/0:      call reference: 0
*Jan 11 12:24:18.827: 1/1/0:      promptStatus: Your current options
*Jan 11 12:24:18.827: 1/1/0:      device control type: 3
*Jan 11 12:24:18.827: 1/1/0:      No state change

```

The following is sample output from the **debug voip application stcapp all** command for a voice gateway during call setup in Cisco IOS Release 12.3(14)T.

The following lines show the voice gateway beginning call setup:

```

*Jan 9 06:48:06.947: ==> Received event:STCAPP_CC_EV_CALL_SETUP_IND
(evId:CC_EV_CALL_SETUP_IND) for CallId: 5
*Jan 9 06:48:06.947: 1/0/0:      Call State:IDLE
*Jan 9 06:48:06.947: 1/0/0:      stcapp_setup_ind_eh
*Jan 9 06:48:06.947: 1/0/0:      Acquired CCB 0x66F12558 for device id:4

```

## debug voip application stcapp all

```
*Jan 9 06:48:06.947: 1/0/0: Voice Setup: callID:5, vdb_ptr:66CA57B4
```

The voice gateway notifies the Cisco CallManager of the endpoint device (phone) in the offhook condition:

```
*Jan 9 06:48:06.947: 1/0/0: Sending StationOffHook to CallManager
*Jan 9 06:48:06.947: 1/0/0: Sending ccCallSetupAck to Symphony for voice call id:5
*Jan 9 06:48:06.947: 1/0/0: New State = OFFHOOK
*Jan 9 06:48:06.955: 1/0/0: No line (line=0) found... most likely old Call Ref: event
STCAPP_DC_EV_DEVICE_SET_RINGER
```

The following lines show that the Cisco CallManager acknowledged the offhook condition:

```
*Jan 9 06:48:06.955: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_OFFHOOK
(evID:DC_EV_DEVICE_CALL_STATE_OFFHOOK)
*Jan 9 06:48:06.955: 1/0/0: Call State:OFFHOOK
*Jan 9 06:48:06.955: 1/0/0: stcapp_cs_offhook_eh
*Jan 9 06:48:06.955: 1/0/0: No state change
```

The voice gateway receives the Cisco CallManager notification to send dial tone to the phone:

```
*Jan 9 06:48:06.955: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 9 06:48:06.955: 1/0/0: Call State:OFFHOOK
*Jan 9 06:48:06.955: 1/0/0: stcapp_start_tone_eh
```

The voice gateway generates dial tone and prepares to collect dialed digits:

```
*Jan 9 06:48:06.955: 1/0/0: Sending ccGenerateTone(8(0x8))
*Jan 9 06:48:06.955: 1/0/0: Sending ccCallReportDigits
*Jan 9 06:48:06.955: 1/0/0: No state change
```

### Related Commands

Command	Description
<b>debug voip application stcapp error</b>	Displays STCAPP error log contents.
<b>debug voip application stcapp events</b>	Traces STCAPP call flow events.
<b>debug voip application stcapp functions</b>	Displays STCAPP entry and exit function calls for all voice ports.
<b>debug voip application stcapp port</b>	Displays debugging information for the components of the STCAPP for a specified port.

# debug voip application stcapp error

To troubleshoot the SCCP Telephony Control Application (STCAPP) error log contents, use the **debug voip application stcapp error** command in privileged EXEC mode. To disable STCAPP error debugging, use the **no** form of this command.

**debug voip application stcapp error**

**no debug voip application stcapp error**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** The **debug voip application stcapp error** command traces application error logs. STCAPP error logs are generated during normal call processing, when there are insufficient resources, or when there are problems in the underlying application code. This **debug** command shows error events or unexpected behavior in system software. Usually no events will be generated.

**Examples** The following example shows the error log contents when STCAPP debugging is enabled:

```
Router# debug voip application stcapp error

STCAPP error debugging is on
Router#
*Jan 9 06:54:07.583:      stcapp_process_queue_events:ERROR:STCAPP_DCB_ACCESS_ERR from
state machine
```

Related Commands	Command	Description
	<b>debug voip application stcapp all</b>	Displays debug output for all the <b>debug</b> commands for the STCAPP compiled into one display.
	<b>debug voip application stcapp events</b>	Traces SCCP Telephony Control Application (STCAPP) call flow events.
	<b>debug voip application stcapp functions</b>	Displays STCAPP entry and exit function calls for all voice ports.
	<b>debug voip application stcapp port</b>	Displays debug information for the components of the SCCP Telephony Control Application for a specified port.

# debug voip application stcapp events

To trace SCCP Telephony Control Application (STCAPP) call flow events, use the **debug voip application stcapp events** command in privileged EXEC mode. To disable STCAPP event call traces, use the **no** form of this command.

**debug voip application stcapp events**

**no debug voip application stcapp events**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use this command to debug call flow events for all ports controlled by the STCAPP.

**Examples** The following example displays call teardown and disconnect events:

```
Router# debug voip application stcapp events
```

The following lines show the application running on the voice gateway receiving notice to stop dial tone generation, following the onhook condition of the endpoint device (phone):

```
*Jan 9 06:48:55.011: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 9 06:48:55.011: 1/0/1: Call State:REM_ONHOOK_PEND
*Jan 9 06:48:55.011: 1/0/1: stcapp_stop_tone_ah
*Jan 9 06:48:55.011: 1/0/1: Sending ccGenerateTone(NULL)
*Jan 9 06:48:55.015: 1/0/1: No state change
```

The application takes no additional action to process the uninteresting event.

```
*Jan 9 06:48:55.015: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 9 06:48:55.015: 1/0/1: Call State:REM_ONHOOK_PEND
*Jan 9 06:48:55.015: 1/0/1: Uninteresting event
```

The application receives the call disconnect notice and proceeds to tear down the telephony call leg.

```
*Jan 9 06:48:58.903: ==> Received event:STCAPP_CC_EV_CALL_DISCONNECTED
(evID:CC_EV_CALL_DISCONNECTED) for CallId: 6
*Jan 9 06:48:58.903: 1/0/1: Call State:REM_ONHOOK_PEND
*Jan 9 06:48:58.903: 1/0/1: stcapp_loc_onhook_ah
*Jan 9 06:48:58.903: 1/0/1: Sending StationOnHook to CallManager
*Jan 9 06:48:58.903: 1/0/1: stcapp_call_cleanup
*Jan 9 06:48:58.903: 1/0/1: stcapp_set_ring_mode
*Jan 9 06:48:58.903: 1/0/1: SCCP ring mode:1
*Jan 9 06:48:58.903: 1/0/1: Invoking Feature:33. Mode:0 for callid:6
```

```
*Jan 9 06:48:58.903: 1/0/1: stcapp_disconnect_call_leg
*Jan 9 06:48:58.903: 1/0/1:      Sending ccCallDisconnect for call id:6
*Jan 9 06:48:58.903: 1/0/1:      CCB 0x65CF3EC4 unlinked
*Jan 9 06:48:58.903: 1/0/1:      New State = IDLE
```

**Related Commands**

Command	Description
<b>debug voip application stcapp all</b>	Displays debug output for all the <b>debug</b> commands for the STCAPP compiled into one display.
<b>debug voip application stcapp error</b>	Displays SCCP Telephony Control Application (STCAPP) error log contents.
<b>debug voip application stcapp functions</b>	Displays STCAPP entry and exit function calls for all voice ports.
<b>debug voip application stcapp port</b>	Displays debug information for the components of the SCCP Telephony Control Application for a specified port.

# debug voip application stcapp functions

To debug SCCP Telephony Control Application (STCAPP) functions, use the **debug voip application stcapp functions** command in privileged EXEC mode. To disable STCAPP function debugging, use the **no** form of this command.

**debug voip application stcapp functions**

**no debug voip application stcapp functions**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

**Usage Guidelines** Use this command to display STCAPP entry and exit function calls for all voice ports.

**Examples** The following example displays function calls for STCAPP ports 1/0/0 and 1/0/1:

```
Router# debug voip application stcapp functions

STCAPP function debugging is on
*Jan  9 06:55:27.583: 1/0/0: stcapp_setup_ind_eh
*Jan  9 06:55:27.591: 1/0/0: stcapp_cs_offhook_eh
*Jan  9 06:55:27.591: 1/0/0: stcapp_start_tone_eh
*Jan  9 06:55:27.591: 1/0/0: stcapp_report_digits_done_eh
*Jan  9 06:55:28.923: 1/0/0: stcapp_digit_end_eh
*Jan  9 06:55:28.927: 1/0/0: stcapp_stop_tone_eh
*Jan  9 06:55:29.063: 1/0/0: stcapp_digit_end_eh
*Jan  9 06:55:29.203: 1/0/0: stcapp_digit_end_eh
*Jan  9 06:55:29.343: 1/0/0: stcapp_digit_end_eh
*Jan  9 06:55:29.355: 1/0/0: stcapp_cs_proceed_eh
*Jan  9 06:55:29.359: 1/0/0: stcapp_proceed_call_info_eh
*Jan  9 06:55:29.359: 1/0/0: stcapp_start_tone_eh
*Jan  9 06:55:29.359: 1/0/0: stcapp_proceed_call_info_eh
*Jan  9 06:55:29.359: 1/0/1: stcapp_cs_ringin_eh
*Jan  9 06:55:29.359: 1/0/1: stcapp_call_info_eh
```

Related Commands	Command	Description
	<b>debug voip application stcapp all</b>	Displays debug output for all the <b>debug</b> commands for the SCCP Telephony Control Application (STCAPP) compiled into one display.
	<b>debug voip application stcapp error</b>	Displays SCCP Telephony Control Application (STCAPP) error log contents.

Command	Description
<b>debug voip application stcapp events</b>	Traces SCCP Telephony Control Application (STCAPP) call flow events.
<b>debug voip application stcapp port</b>	Displays debug information for the components of the SCCP Telephony Control Application (STCAPP) for a specified port.

# debug voip application stcapp port

To enable SCCP Telephony Control Application (STCAPP) debugging for a specific port, use the **debug voip application stcapp port** command in privileged EXEC mode. To disable specific STCAPP port debugging, use the **no** form of this command.

**debug voip application stcapp port** *port-number*

**no debug voip application stcapp port** *port-number*

## Syntax Description

<i>port-number</i>	Number of the port on the interface. See the appropriate platform manual or online help for port numbers on your networking device.
--------------------	---

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(4)T	Command output was enhanced to display the modem transport method.
12.4(6)XE	Command output was enhanced to display fax relay, RFC 2833 DTMF digit relay, dial tone after remote onhook, call control feature mode and visual message waiting indicator (VMWI) information for SCCP analog ports.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

## Usage Guidelines

Use this command to display debugging information for the components of the STCAPP for a specific port.

## Examples

The following example displays RFC 2833 DTMF digits messages sent and received on a voice gateway in Cisco IOS Release 12.4(6)XE:

```
Router# debug voip application stcapp port 2/3
```

The following lines show the SCCP gateway receiving the RFC payload.

```
Mar 4 00:23:31.166: 2/3 : ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_RCV_CHNL
*Mar 4 00:23:31.166: 2/3 : Call State:PROCEEDING
*Mar 4 00:23:31.166: 2/3 : stcapp_open_rcv_chnl_eh
*Mar 4 00:23:31.166: 2/3 : call_ref=20797703
*Mar 4 00:23:31.166: 2/3 : stcapp_get_ccb_ptr
*Mar 4 00:23:31.166: 2/3 : received ORC: rcv payload=101
*Mar 4 00:23:31.166: 2/3 : stcapp_set_up_voip_leg
*Mar 4 00:23:31.166: 2/3 : stcapp_get_ccb_ptr
*Mar 4 00:23:31.166: 2/3 : stcapp_set_up_modem_parms
```

The following lines show the SCCP gateway sending the RFC payload.

```
*Mar 4 00:23:31.174: 2/3 : ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_XMT_CHNL
```

```
*Mar 4 00:23:31.174: 2/3 : Call State:CONNECTING
*Mar 4 00:23:31.174: 2/3 : stcapp_start_media_ah
*Mar 4 00:23:31.174: 2/3 : call_ref=20797703
*Mar 4 00:23:31.174: 2/3 : stcapp_get_ccb_ptr
*Mar 4 00:23:31.174: 2/3 : lcb->mode 0, lcb->conf_status 0
*Mar 4 00:23:31.174: 2/3 : received XMT: send payload=101
*Mar 4 00:23:31.174: 2/3 : Start media: CCB Count:1 Call Leg Count:2
*Mar 4 00:23:31.174: 2/3 : New State = ACTIVE_PENDING
```

The following example displays VMWI lamp state messages between Cisco CallManager and a voice gateway in Cisco IOS Release 12.4(6)XE:

```
Router# debug voip application stcapp port 2/4
```

```
*Mar 1 01:41:58.395: 2/0 : No state change... call remaining
*Mar 16 21:47:14.045: 2/4 : stcapp_screen_api_event
```

The following lines show the gateway receiving messages from Cisco callManager to activate the VMWI lamp.

```
*Mar 16 21:47:14.045: 2/4 : event:STCAPP_DC_EV_DEVICE_SET_LAMP received.
*Mar 16 21:47:14.049: 2/4 : msg_mwi 1, mwi 0, vmwi 0
*Mar 16 21:47:14.049: 2/4 : event STCAPP_DC_EV_DEVICE_SET_LAMP_PROCESS_VMWI
created.
*Mar 16 21:47:14.049: 2/4 : New State = VMWI_DSP_SETUP
*Mar 16 21:47:14.053: 2/4 : ==> Received event:STCAPP_CC_EV_CALL_PROCEEDING for
CallId: 229
*Mar 16 21:47:14.053: 2/4 : Call State:VMWI_DSP_SETUP
*Mar 16 21:47:14.053: 2/4 : stcapp_vmwi_call_proceed_ah
*Mar 16 21:47:14.057: 2/4 : No state change
*Mar 16 21:47:14.057: 2/4 : ==> Received event:STCAPP_CC_EV_CALL_MODIFY_DONE for
CallId: 229
*Mar 16 21:47:14.057: 2/4 : Call State:VMWI_DSP_SETUP
*Mar 16 21:47:14.057: 2/4 : stcapp_vmwi_call_modify_done_ah
*Mar 16 21:47:14.057: 2/4 : Sending ccCallFeature (vmwi = on)
*Mar 16 21:47:14.057: 2/4 : New State = VMWI_PENDING
```

The following lines show the VMWI activation is completed and call-tear down is beginning.

```
*Mar 16 21:47:15.237: 2/4 : ==> Received event:STCAPP_CC_EV_VBD_XMIT_DONE for CallId:229
*Mar 16 21:47:15.237: 2/4 : Call State:VMWI_PENDING
*Mar 16 21:47:15.237: 2/4 : stcapp_vmwi_fsk_gen_done_ah
*Mar 16 21:47:15.237: 2/4 : stcapp_get_ccb_ptr
*Mar 16 21:47:15.237: 2/4 : disconnect voice call leg
*Mar 16 21:47:15.237: 2/4 : stcapp_disconnect_call_leg
*Mar 16 21:47:15.237: 2/4 : Sending ccCallDisconnect for VOICE_LEG with call id:229
```

The following example displays information on a call between ports 1/1/0 and 1/1/1 after port 1/1/1 hangs up and the automatic dial tone generation after remote onhook feature is enabled:

```
Router# debug voip application stcapp port 1/1/0
```

```
Jan 7 00:41:37.484: 1/1/0: ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_RCV_CHNL
*Jan 7 00:41:37.484: 1/1/0: Call State:ACTIVE
```

The following lines show the call being disconnected.

```
*Jan 7 00:41:37.484: 1/1/0: stcapp_close_rcv_chnl_ah
*Jan 7 00:41:37.484: 1/1/0: call_ref=209
*Jan 7 00:41:37.484: 1/1/0: stcapp_get_ccb_ptr
*Jan 7 00:41:37.484: 1/1/0: Sending ccConferenceDestroy
*Jan 7 00:41:37.484: 1/1/0: Sending ccCallDisconnect for voip call id:44
```

```

*Jan 7 00:41:37.484: 1/1/0: stcapp_disconnect_call_leg
*Jan 7 00:41:37.484: 1/1/0: Sending ccCallDisconnect for VoIP_LEG with call id:44
*Jan 7 00:41:37.484: 1/1/0: No state change
*Jan 7 00:41:37.488: 1/1/0: ==> Received event:STCAPP_DC_EV_MEDIA_CLOSE_XMT_CHNL
*Jan 7 00:41:37.488: 1/1/0: Call State:ACTIVE
*Jan 7 00:41:37.488: 1/1/0: stcapp_active_close_xmt_chnl_eh
*Jan 7 00:41:37.488: 1/1/0: lcb->mode 0 lcb->conf_status 0
*Jan 7 00:41:37.488: 1/1/0: New State = ONHOOK_PEND
*Jan 7 00:41:37.488: 1/1/0: stcapp_cs_onhook_eh
*Jan 7 00:41:37.488: 1/1/0: stcapp_get_ccb_ptr
*Jan 7 00:41:37.488: 1/1/0: call_ref=209, ccb=0x4662B31C, lcb->num_ccbs=1
*Jan 7 00:41:37.488: 1/1/0: stcapp_process_cs_onhook
*Jan 7 00:41:37.488: 1/1/0: lcb->mode=CALL_BASIC (0)

```

The following lines show power denial-based supervisory disconnect signal being sent.

```

*Jan 7 00:41:37.488: 1/1/0: Sending power denial signal to device 9
*Jan 7 00:41:37.488: 1/1/0: stcapp_update_dialtone_gen_trigger
*Jan 7 00:41:37.488: 1/1/0: stcapp_send_softkey_event
*Jan 7 00:41:37.488: 1/1/0: Sending dcDeviceSoftKeyEvent(NewCall: event=2) for device
id: 9, call_ref: 0
*Jan 7 00:41:37.488: 1/1/0: New State = REM_ONHOOK_PEND
*Jan 7 00:41:37.488: 1/1/0: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
*Jan 7 00:41:37.488: 1/1/0: Call State:REM_ONHOOK_PEND
*Jan 7 00:41:37.488: 1/1/0: stcapp_stop_tone_eh
*Jan 7 00:41:37.488: 1/1/0: call_ref=209
*Jan 7 00:41:37.488: 1/1/0: stcapp_get_ccb_ptr
*Jan 7 00:41:37.488: 1/1/0: Sending ccGenerateTone(NULL)
*Jan 7 00:41:37.488: 1/1/0: No state change
*Jan 7 00:41:37.492: 1/1/0: ==> Received event:STCAPP_CC_EV_CONF_DESTROY_DONE for CallId:
41
*Jan 7 00:41:37.492: 1/1/0: Call State:REM_ONHOOK_PEND
*Jan 7 00:41:37.492: 1/1/0: Uninteresting event
*Jan 7 00:41:37.492: 1/1/0: stcapp_screen_api_event
*Jan 7 00:41:37.492: 1/1/0: event:STCAPP_CC_EV_CALL_DISCONNECT_DONE received.
*Jan 7 00:41:37.492: STCAPP:Receive CC event:: call_id=44, ccb=0x4662B31C
*Jan 7 00:41:37.492: 1/1/0: Received event:CC_EV_CALL_DISCONNECT_DONE for CallId: 44
*Jan 7 00:41:37.492: 1/1/0: stcapp_process_disconnect_done
*Jan 7 00:41:37.492: 1/1/0: stcapp_reset_call_leg
*Jan 7 00:41:37.492: 1/1/0: ccb(0x4662B31C): voice/voip call_id=41/44, reset
call_id=44
*Jan 7 00:41:37.492: 1/1/0: stcapp_conn_db_delete_ccb
*Jan 7 00:41:37.492: 1/1/0: ccb=0x4662B31C
*Jan 7 00:41:37.492: 1/1/0: Disconnect Done: CCB Count:1 Call Leg Count:1
*Jan 7 00:41:37.496: 1/1/0: stcapp_get_dcb_and_lcb

```

The following lines show the call transitioning to off hook.

```

*Jan 7 00:41:37.496: 1/1/0: stcapp_screen_api_event
*Jan 7 00:41:37.496: 1/1/0: event:STCAPP_DC_EV_DEVICE_CALL_STATE_OFFHOOK received.
*Jan 7 00:41:37.496: 1/1/0: Create new event
STCAPP_DC_EV_DEV_CS_OFFHOOK_DIALTONE_GEN*Jan 7 00:41:37.496: 1/1/0: ==> Received
event:STCAPP_DC_EV_DEVICE_CALL_STATE_OFFHOOK_DIALTONE_GEN
*Jan 7 00:41:37.496: 1/1/0: Call State:REM_ONHOOK_PEND
*Jan 7 00:41:37.496: 1/1/0: stcapp_cs_offhook_dialtone_gen_eh
*Jan 7 00:41:37.496: 1/1/0: call_ref=211
*Jan 7 00:41:37.496: 1/1/0: New State = OFFHOOK
*Jan 7 00:41:37.496: 1/1/0: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 7 00:41:37.496: 1/1/0: Device State:IS
*Jan 7 00:41:37.496: 1/1/0: stcapp_display_prompt_status_eh
*Jan 7 00:41:37.496: 1/1/0: lineNumber: 1
*Jan 7 00:41:37.496: 1/1/0: call reference: 211
*Jan 7 00:41:37.496: 1/1/0: promptStatus: ' '
*Jan 7 00:41:37.496: 1/1/0: No state change

```

The following lines show the message to generate dial tone for the new call.

```
*Jan 7 00:41:37.496: 1/1/0: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
*Jan 7 00:41:37.496: 1/1/0: Call State:OFFHOOK
*Jan 7 00:41:37.496: 1/1/0: stcapp_start_tone_eh
*Jan 7 00:41:37.496: 1/1/0: stcapp_get_ccb_ptr
*Jan 7 00:41:37.496: 1/1/0: call_ref=211, ccb=0x4662B31C, tone=8(0x8), dir=1
*Jan 7 00:41:37.496: 1/1/0: plar enable (0), hookflash (0)
*Jan 7 00:41:37.496: 1/1/0: plar enable (0), tone (0x8), hookflash (0)
*Jan 7 00:41:37.496: 1/1/0: Sending ccGenerateTone(8(0x8)) ? produce dial tone
*Jan 7 00:41:37.496: 1/1/0: Sending ccCallReportDigits
*Jan 7 00:41:37.496: 1/1/0: No state change
```

The following example displays information for a modem-relay call on a voice gateway in Cisco IOS Release 12.4(4)T:

Router# **debug voip application stcapp port 1/1/0**

```
*Jan 11 12:37:48.631: ==> Received event:STCAPP_CC_EV_CALL_SETUP_IND
(evId:CC_EV_CALL_SETUP_IND) for CallId: 326
*Jan 11 12:37:48.631: 1/1/0: Call State:IDLE
*Jan 11 12:37:48.631: 1/1/0: stcapp_setup_ind_eh
*Jan 11 12:37:48.631: 1/1/0: stcapp_get_ccb
*Jan 11 12:37:48.631: 1/1/0: dcb->lcb[line_inst - 1].num_ccbs=0
*Jan 11 12:37:48.631: 1/1/0: Acquired CCB 0x65D932B8 for device id:7
*Jan 11 12:37:48.631: 1/1/0: num_ccbs++, num_ccbs=1
*Jan 11 12:37:48.631: 1/1/0: Voice Setup: callID:326, vdb_ptr:666581AC
*Jan 11 12:37:48.631: 1/1/0: Sending StationOffHook to CallManager
*Jan 11 12:37:48.631: 1/1/0: Sending ccCallSetupAck to Symphony for voice call id:326
*Jan 11 12:37:48.631: 1/1/0: New State = OFFHOOK
*Jan 11 12:37:48.643: 1/1/0: No line (line=0) found... most likely old Call Ref: event
STCAPP_DC_EV_DEVICE_SET_RINGER
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_OFFHOOK
(evID:DC_EV_DEVICE_CALL_STATE_OFFHOOK)
*Jan 11 12:37:48.643: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:48.643: 1/1/0: stcapp_cs_offhook_eh
*Jan 11 12:37:48.643: 1/1/0: call_ref=16777250
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0: Using call_ref 0 to get ccb=0x65D932B8
*Jan 11 12:37:48.643: 1/1/0: No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:48.643: 1/1/0: Device State:IS
*Jan 11 12:37:48.643: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:48.643: 1/1/0: lineNumber: 1
*Jan 11 12:37:48.643: 1/1/0: call reference: 16777250
*Jan 11 12:37:48.643: 1/1/0: promptStatus: Enter Number
*Jan 11 12:37:48.643: 1/1/0: No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 11 12:37:48.643: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:48.643: 1/1/0: stcapp_start_tone_eh
*Jan 11 12:37:48.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:48.643: 1/1/0: call_ref=16777250, ccb=0x65D932B8, tone=8(0x8)
*Jan 11 12:37:48.643: 1/1/0: Sending ccGenerateTone(8(0x8))
*Jan 11 12:37:48.643: 1/1/0: Sending ccCallReportDigits
*Jan 11 12:37:48.643: 1/1/0: No state change
*Jan 11 12:37:48.643: ==> Received event:STCAPP_CC_EV_CALL_REPORT_DIGITS_DONE
(evId:CC_EV_CALL_REPORT_DIGITS_DONE) for CallId: 326
*Jan 11 12:37:48.647: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:48.647: 1/1/0: stcapp_report_digits_done_eh
*Jan 11 12:37:48.647: 1/1/0: No state change
*Jan 11 12:37:52.643: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evId:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
```

```

*Jan 11 12:37:52.643: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.643: 1/1/0: Uninteresting event
*Jan 11 12:37:52.683: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evID:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.683: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.683: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.683: 1/1/0: Digit received is (5)
*Jan 11 12:37:52.683: 1/1/0: Sending StationKeypadButton(5) to CallManager
*Jan 11 12:37:52.683: 1/1/0: No state change
*Jan 11 12:37:52.687: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:52.687: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.687: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:52.687: 1/1/0: call_ref=16777250
*Jan 11 12:37:52.687: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:52.687: 1/1/0: Sending ccGenerateTone(NULL)
*Jan 11 12:37:52.687: 1/1/0: No state change
*Jan 11 12:37:52.775: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evID:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:52.775: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.775: 1/1/0: Uninteresting event
*Jan 11 12:37:52.823: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evID:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.823: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.823: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.823: 1/1/0: Digit received is (8)
*Jan 11 12:37:52.823: 1/1/0: Sending StationKeypadButton(8) to CallManager
*Jan 11 12:37:52.823: 1/1/0: No state change
*Jan 11 12:37:52.923: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evID:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:52.923: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.923: 1/1/0: Uninteresting event
*Jan 11 12:37:52.963: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evID:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:52.963: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:52.963: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:52.963: 1/1/0: Digit received is (0)
*Jan 11 12:37:52.963: 1/1/0: Sending StationKeypadButton(0) to CallManager
*Jan 11 12:37:52.963: 1/1/0: No state change
*Jan 11 12:37:53.063: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_BEGIN
(evID:CC_EV_CALL_DIGIT_BEGIN) for CallId: 326
*Jan 11 12:37:53.063: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:53.063: 1/1/0: Uninteresting event
*Jan 11 12:37:53.103: ==> Received event:STCAPP_CC_EV_CALL_DIGIT_END
(evID:CC_EV_CALL_DIGIT_END) for CallId: 326
*Jan 11 12:37:53.103: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:53.103: 1/1/0: stcapp_digit_end_eh
*Jan 11 12:37:53.103: 1/1/0: Digit received is (2)
*Jan 11 12:37:53.103: 1/1/0: Sending StationKeypadButton(2) to CallManager
*Jan 11 12:37:53.103: 1/1/0: No state change
*Jan 11 12:37:53.235: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_PROCEED
(evID:DC_EV_DEVICE_CALL_STATE_PROCEED)
*Jan 11 12:37:53.235: 1/1/0: Call State:OFFHOOK
*Jan 11 12:37:53.235: 1/1/0: stcapp_cs_proceed_eh
*Jan 11 12:37:53.235: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.235: 1/1/0: Sending ccCallProceeding for voice call id:326
*Jan 11 12:37:53.235: 1/1/0: Stopping the initial and inter digit timer!
*Jan 11 12:37:53.235: 1/1/0: New State = PROCEEDING
*Jan 11 12:37:53.235: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:53.235: 1/1/0: Call State:PROCEEDING
*Jan 11 12:37:53.235: 1/1/0: stcapp_proceed_call_info_eh
*Jan 11 12:37:53.235: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0: No state change

```

```

*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_START_TONE
(evID:DC_EV_DEVICE_START_TONE)
*Jan 11 12:37:53.239: 1/1/0:      Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_start_tone_eh
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:      call_ref=16777250, ccb=0x65D932B8, tone=1(0x1)
*Jan 11 12:37:53.239: 1/1/0:      Sending ccCallAlert(signal:1) for voice call id:326
*Jan 11 12:37:53.239: 1/1/0:      No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_RINGOUT
(evID:DC_EV_DEVICE_CALL_STATE_RINGOUT)
*Jan 11 12:37:53.239: 1/1/0:      Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_set_call_state_eh
*Jan 11 12:37:53.239: 1/1/0:      call_ref=16777250, call_state=2
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:      No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:53.239: 1/1/0:      Device State:IS
*Jan 11 12:37:53.239: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:53.239: 1/1/0:      lineNumber: 1
*Jan 11 12:37:53.239: 1/1/0:      call reference: 16777250
*Jan 11 12:37:53.239: 1/1/0:      promptStatus: Ring Out
*Jan 11 12:37:53.239: 1/1/0:      No state change
*Jan 11 12:37:53.239: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:53.239: 1/1/0:      Call State:PROCEEDING
*Jan 11 12:37:53.239: 1/1/0: stcapp_proceed_call_info_eh
*Jan 11 12:37:53.239: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:53.239: 1/1/0:      No state change
*Jan 11 12:37:56.635: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:56.635: 1/1/0:      Call State:PROCEEDING
*Jan 11 12:37:56.635: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:56.635: 1/1/0:      call_ref=16777250
*Jan 11 12:37:56.635: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0:      Sending ccGenerateTone(NULL)
*Jan 11 12:37:56.639: 1/1/0:      No state change
*Jan 11 12:37:56.639: ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_RCV_CHNL
(evID:DC_EV_MEDIA_OPEN_RCV_CHNL)
*Jan 11 12:37:56.639: 1/1/0:      Call State:PROCEEDING
*Jan 11 12:37:56.639: 1/1/0: stcapp_open_rcv_chnl_eh
*Jan 11 12:37:56.639: 1/1/0:      call_ref=16777250
*Jan 11 12:37:56.639: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0: stcapp_set_up_void_leg
*Jan 11 12:37:56.639: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.639: 1/1/0:      Codec: 5 ptime :20, codecbytes: 160

```

The following line indicates the modem transport method that will be used:

```
*Jan 11 12:37:56.639: 1/1/0:      CCM directive -> enabling modem relay
```

The following lines show modem-relay parameters:

```

*Jan 11 12:37:56.639: 1/1/0:      MR parms: sprt_retries=10, sprt_latency=250,
sprt_rx_v14_pb_hold_time=32, sprt_tx_v14_hold_time=12, sprt_tx_v14_hold_count=22,
gw_xid=1, dictsize=1024, stringlen=16, compressdir=3, sse_red_interval=16,
sse_red_pkt_count=2, sse_tl=2100, sse_retries=5
*Jan 11 12:37:56.639: 1/1/0:      Info provided to RTPSPI - sess_mode 2, desired_qos 0,
codec 5, pkt_period 20, lr_port 17180
*Jan 11 12:37:56.639: 1/1/0:      Sending ccIFCallSetupRequest for voip leg
*Jan 11 12:37:56.639: 1/1/0:      ccIFCallSetRequest returned voip call id:327
*Jan 11 12:37:56.639: 1/1/0:      Sending dcDeviceOpenReceiveChannelAck
*Jan 11 12:37:56.639: 1/1/0:      ORChnlAck Info: codec:5, loc_port:17180, chnl_id:16777521
*Jan 11 12:37:56.639: 1/1/0:      New State = CONNECTING

```

```

*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_CONNECTED
(evID:DC_EV_DEVICE_CALL_STATE_CONNECTED)
*Jan 11 12:37:56.643: 1/1/0:      Call State:CONNECTING
*Jan 11 12:37:56.643: 1/1/0: stcapp_set_call_state_eh
*Jan 11 12:37:56.643: 1/1/0:      call_ref=16777250, call_state=6
*Jan 11 12:37:56.643: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.643: 1/1/0:      No state change
*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_DISPLAY_PROMPT_STATUS
*Jan 11 12:37:56.643: 1/1/0:      Device State:IS
*Jan 11 12:37:56.643: 1/1/0: stcapp_display_prompt_status_eh
*Jan 11 12:37:56.643: 1/1/0:      lineNumber: 1
*Jan 11 12:37:56.643: 1/1/0:      call reference: 16777250
*Jan 11 12:37:56.643: 1/1/0:      promptStatus: Connected
*Jan 11 12:37:56.643: 1/1/0:      No state change
*Jan 11 12:37:56.643: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan 11 12:37:56.643: 1/1/0:      Call State:CONNECTING
*Jan 11 12:37:56.643: 1/1/0: stcapp_conn_call_info_eh
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:      stcapp_call_info_eh::caller_name=
*Jan 11 12:37:56.647: 1/1/0:      Irrelevant CALL_INFO message is ignore!
*Jan 11 12:37:56.647: 1/1/0:      No state change
*Jan 11 12:37:56.647: ==> Received event:STCAPP_DC_EV_DEVICE_STOP_TONE
(evID:DC_EV_DEVICE_STOP_TONE)
*Jan 11 12:37:56.647: 1/1/0:      Call State:CONNECTING
*Jan 11 12:37:56.647: 1/1/0: stcapp_stop_tone_eh
*Jan 11 12:37:56.647: 1/1/0:      call_ref=16777250
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:      Sending ccGenerateTone(NULL)
*Jan 11 12:37:56.647: 1/1/0:      No state change
*Jan 11 12:37:56.647: ==> Received event:STCAPP_DC_EV_MEDIA_OPEN_XMT_CHNL
(evID:DC_EV_MEDIA_OPEN_XMT_CHNL)
*Jan 11 12:37:56.647: 1/1/0:      Call State:CONNECTING
*Jan 11 12:37:56.647: 1/1/0: stcapp_start_media_eh
*Jan 11 12:37:56.647: 1/1/0:      call_ref=16777250
*Jan 11 12:37:56.647: 1/1/0: stcapp_get_ccb_ptr
*Jan 11 12:37:56.647: 1/1/0:      New State = ACTIVE_PENDING
*Jan 11 12:37:56.647: ==> Received event:STCAPP_CC_EV_CALL_CONNECTED
(evID:CC_EV_CALL_CONNECTED) for CallId: 327
*Jan 11 12:37:56.647: 1/1/0:      Call State:ACTIVE_PENDING
*Jan 11 12:37:56.647: 1/1/0: stcapp_call_connected_eh
*Jan 11 12:37:56.647: 1/1/0: stcapp_create_conference
*Jan 11 12:37:56.647: 1/1/0:      Sending ccConferenceCreate to Symphony
*Jan 11 12:37:56.651: 1/1/0:      Conference created. voice call id:326, voip call id:327
*Jan 11 12:37:56.651: 1/1/0:      No state change
*Jan 11 12:37:56.651: ==> Received event:STCAPP_CC_EV_CONF_CREATE_DONE
(evID:CC_EV_CONF_CREATE_DONE) for CallId: 326
*Jan 11 12:37:56.651: 1/1/0:      Call State:ACTIVE_PENDING
*Jan 11 12:37:56.651: 1/1/0: stcapp_active_pending_eh
*Jan 11 12:37:56.651: 1/1/0:      Sending ccCallModify for voice call id:326
*Jan 11 12:37:56.651: 1/1/0:      codec=5, vad=0
*Jan 11 12:37:56.651: 1/1/0:      Stopping the initial and inter digit timer!
*Jan 11 12:37:56.651: 1/1/0:      Sending ccCallModify for voip call id:327
*Jan 11 12:37:56.651: 1/1/0:      Updated SMT info to RTPSPI - sess_mode:3,desired_qos:0,
codec:5, pkt_period:20,rem_port:18968 vad:0 ip_tos:4
*Jan 11 12:37:56.655: 1/1/0:      No state change
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_VOICE_MODE_DONE
(evID:CC_EV_VOICE_MODE_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:      Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0:      Uninteresting event
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_REPORT_DIGITS_DONE
(evID:CC_EV_CALL_REPORT_DIGITS_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:      Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0:      Uninteresting event

```

```

*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_MODIFY_DONE
(evId:CC_EV_CALL_MODIFY_DONE) for CallId: 326
*Jan 11 12:37:56.655: 1/1/0:      Call State:ACTIVE_PENDING
*Jan 11 12:37:56.655: 1/1/0: stcapp_default_eh
*Jan 11 12:37:56.655: 1/1/0:      call_ref=0, call_state=0
*Jan 11 12:37:56.655: 1/1/0:      New State = ACTIVE
*Jan 11 12:37:56.655: ==> Received event:STCAPP_CC_EV_CALL_MODIFY_DONE
(evId:CC_EV_CALL_MODIFY_DONE) for CallId: 327
*Jan 11 12:37:56.655: 1/1/0:      Call State:ACTIVE
*Jan 11 12:37:56.655: 1/1/0:      Uninteresting event
*Jan 11 12:37:59.963: ==> Received event:STCAPP_CC_EV_CALL_FEATURE_OFFHOOK
(evId:CC_EV_CALL_FEATURE) for CallId: 326
*Jan 11 12:37:59.963: 1/1/0:      Call State:ACTIVE
*Jan 11 12:37:59.963: 1/1/0: stcapp_call_feature_eh
*Jan 11 12:37:59.963: 1/1/0:      lcb->num_ccbs = 1
*Jan 11 12:37:59.963: 1/1/0:      No CC_FEATURE match!
*Jan 11 12:37:59.967: 1/1/0:      No state change... call remaining

```

The following example displays information on STCAPP controlled foreign exchange station (FXS) port 1/0/1 during call setup for a voice gateway in Cisco IOS 12.3(14)T:

```
Router# debug voip application stcapp port 1/0/1
```

```
stcapp port debugging is on
```

The following lines show the voice gateway receiving notification from the Cisco CallManager of an incoming call:

```

*Jan  9 06:57:24.403: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_STATE_RINGIN
(evID:DC_EV_DEVICE_CALL_STATE_RINGIN)
*Jan  9 06:57:24.403: 1/0/1:      Call State:IDLE
*Jan  9 06:57:24.403: 1/0/1: stcapp_cs_ringin_eh
*Jan  9 06:57:24.407: 1/0/1:      Acquired CCB 0x66C0A428 for device id:3

```

The next lines show the new call processing state for the port:

```

*Jan  9 06:57:24.407: 1/0/1:      New State = RINGIN
*Jan  9 06:57:24.407: ==> Received event:STCAPP_DC_EV_DEVICE_CALL_INFO
(evID:DC_EV_DEVICE_CALL_INFO)
*Jan  9 06:57:24.407: 1/0/1:      Call State:RINGIN
*Jan  9 06:57:24.407: 1/0/1: stcapp_call_info_eh

```

The next lines show the application sending a call setup request for the telephony leg:

```

*Jan  9 06:57:24.407: 1/0/1: stcapp_set_up_voice_leg
*Jan  9 06:57:24.407: 1/0/1:      Sending ccIFCallSetupRequest for voice leg
*Jan  9 06:57:24.407: 1/0/1:      ccIFCallSetRequest returned voice call id:22.
CdPN:7702CgPN:7701
*Jan  9 06:57:24.407: 1/0/1:      No state change

```

The next lines show the application invoking the ringing state:

```

*Jan  9 06:57:24.407: ==> Received event:STCAPP_DC_EV_DEVICE_SET_RINGER
(evID:DC_EV_DEVICE_SET_RINGER)
*Jan  9 06:57:24.407: 1/0/1:      Call State:RINGIN
*Jan  9 06:57:24.407: 1/0/1: stcapp_set_ringer_eh
*Jan  9 06:57:24.407: 1/0/1: stcapp_set_ring_mode
*Jan  9 06:57:24.407: 1/0/1:      SCCP ring mode:2
*Jan  9 06:57:24.407: 1/0/1:      Invoking Feature:12. Mode:0 for callid:22
*Jan  9 06:57:24.407: 1/0/1:      No state change

```

**debug voip application stcapp port**

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>debug voip application stcapp all</b>	Displays debug output for all the debug commands for the STCAPP compiled into one display.
	<b>debug voip application stcapp error</b>	Displays STCAPP error log contents.
	<b>debug voip application stcapp events</b>	Traces STCAPP call flow events.
	<b>debug voip application stcapp functions</b>	Displays STCAPP entry and exit function calls for all voice ports.

# debug voip dcapi

To debug the device control application interface (DCAPI), use the **debug voip dcapi** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug voip dcapi [error | inout]**

**no debug voip dcapi [error | inout]**

## Syntax Description

<b>error</b>	(Optional) Displays error logs in the DCAPI.
<b>inout</b>	(Optional) Displays the execution path through the DCAPI.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.3(14)T	This command was introduced.

## Usage Guidelines

The **debug voip dcapi error** command traces the error logs in the device control API (DCAPI), which is the software layer that interfaces the SCCP Telephony Control Application (STCAPP) with the Cisco CallManager using the Skinny Client Control Protocol (SCCP). Error logs are generated during normal call processing, when there are insufficient resources, or when there are problems in the device control API itself. This **debug** command shows error events or unexpected behavior in system software.

The **debug voip dcapi inout** command shows how a call executes through the software. This command traces the execution path through the DCAPI during communications with the SCCP service provider interface (SPI) and the call control API (CCAPI) that controls the physical voice port. You can use the output from this command to understand how devices are being handled by the APIs and to see the call setup and teardown operations performed on the telephony call leg.

## Examples

The following is sample output from the **debug voip dcapi inout** command during call setup:

```
Router# debug voip dcapi inout

*Jan 27 16:26:23.957: dc_api_device_set_ringer_res: Set Ringer message success
*Jan 27 16:26:23.957: //-1/xxxxxxxxxxxx/CCAPI/dc_api_device_stop_tone_res:

*Jan 27 16:26:23.957: dc_api_device_stop_tone_res: Stop Tone message success
*Jan 27 16:26:23.957: //-1/xxxxxxxxxxxx/CCAPI/dc_api_media_open_rcv_channel:

*Jan 27 16:26:23.957: dc_api_media_open_rcv_channel: evt DC_EV_MEDIA_OPEN_RCV_CHNL is
successfully enqueued to app
*Jan 27 16:26:23.957: //-1/xxxxxxxxxxxx/CCAPI/dc_api_device_stop_tone_res:

*Jan 27 16:26:23.957: dc_api_device_stop_tone_res: Stop Tone message success
*Jan 27 16:26:23.957: //-1/xxxxxxxxxxxx/CCAPI/dc_api_device_call_state_res:

*Jan 27 16:26:23.957: dc_api_device_call_state_res: Call State message success
*Jan 27 16:26:23.957: //-1/xxxxxxxxxxxx/CCAPI/dc_api_device_call_info_res:
```

Table 2 describes the significant fields shown in the displays.

**Table 3** *debug voip dcapi Field Descriptions*

Field	Description
<i>nn:nn:nn:</i>	Time stamp time in hours (military format), minutes, and seconds that indicates when the DCAPI event occurred.
<i>dc_api_message:</i>	The DCAPI event in which the SCCP SPI translation occurred.

#### Related Commands

Command	Description
<b>debug voip application stcapp events</b>	Debugs STCAPP events.
<b>debug voip application stcapp functions</b>	Debugs STCAPP functions.

# show call active voice

To display call information for voice calls in progress, use the **show call active voice** command in user EXEC or privileged EXEC mode.

```
show call active voice [brief] [called-number number | calling-number number] | compact
[duration {less seconds | more seconds}] | echo-canceller call-id | id identifier |
media-inactive [called-number number | calling-number number] | [redirect {rtpvt | tbct}]
```

Syntax Description	
<b>brief</b>	(Optional) Displays a truncated version of call information.
<b>called-number</b> <i>number</i>	(Optional) Displays a specific called number pattern.
<b>calling-number</b> <i>number</i>	(Optional) Displays a specific calling number pattern.
<b>compact</b>	(Optional) Displays a compact version of call information.
<b>duration</b>	(Optional) Displays active calls that are longer or shorter than a specified <i>seconds</i> . The argument and keywords are as follows: <ul style="list-style-type: none"> <li><b>less</b>—Displays calls shorter than the <i>seconds</i> value.</li> <li><b>more</b>—Displays calls longer than the <i>seconds</i> value.</li> <li><b>seconds</b>—Elapsed time, in seconds. Range is from 1 to 2147483647. There is no default value.</li> </ul>
<b>echo-canceller</b> <i>call-id</i>	(Optional) Displays information about the state of the extended echo canceller (EC). To query the echo state, you need to know the hex ID in advance. To find the hex ID, enter the <b>show call active voice brief</b> command or use the <b>show voice call status</b> command. Range is from 0 to FFFFFFFF.
<b>id</b> <i>identifier</i>	(Optional) Displays only the call with the specified <i>identifier</i> . Range is a hex value from 1 to FFFF.
<b>media-inactive</b>	(Optional) Displays information about inactive media that have been detected.
<b>redirect</b>	(Optional) Displays information about active calls that are being redirected using Release-to-Pivot (RTPvt) or Two B-Channel Transfer (TBCT). The keywords are: <ul style="list-style-type: none"> <li><b>rtpvt</b>—Displays information about RTPvt calls.</li> <li><b>tbct</b>—Displays information about TBCT calls.</li> </ul>

Command Modes	
	User EXEC Privileged EXEC

Command History	Release	Modification
	11.3(1)T	This command was introduced on the Cisco 2600 series and Cisco 3600 series.
	12.0(3)XG	Support for VoFR was added.
	12.0(4)XJ	This command was implemented for store-and-forward fax on the Cisco AS5300.

Release	Modification
12.0(4)T	This command was implemented on the Cisco 7200 series.
12.0(7)XK	This command was implemented on the Cisco MC3810.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(3)T	This command was implemented for modem pass-through over VoIP on the Cisco AS5300.
12.1(5)XM	This command was implemented on the Cisco AS5800.
12.1(5)XM2	The command was implemented on the Cisco AS5350 and Cisco AS5400.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T. Support is not included for the Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(11)T	Support was added for the Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	The <b>echo-canceller</b> keyword was added. The command output was modified with an extra reflector location when the extended EC is present; the largest reflector location is shown.
12.3(1)	The <b>redirect</b> keyword was added.
12.3(4)T	The <b>called-number</b> , <b>calling-number</b> , and <b>media-inactive</b> keywords were added.
12.3(14)T	New output relating to Skinny Client Control Protocol (SCCP), SCCP Telephony Control Application (STCAPP), and modem pass-through traffic was added.
12.4(2)T	The LocalHostname display field was added to the VoIP call leg record and command output was enhanced to display modem relay physical layer and error correction protocols.

### Usage Guidelines

Use this command to display the contents of the active call table. This command displays information about call times, dial peers, connections, quality of service, and other status and statistical information for voice calls currently connected through the router.

When the extended EC is present, the **show call active voice** command displays the contents of the Ditech EC\_CHAN\_CTRL structure. [Table 4](#) contains name and descriptions of the fields in the EC\_CHAN\_CTRL structure.

**Table 4** *EC\_CHAN\_CTRL Field Descriptions*

Symbol	Field	Description
BYP0	Channel bypass	1 = Transparent bypass; EC is disabled. 0 = Cancel; EC is enabled.

**Table 4** EC\_CHAN\_CTRL Field Descriptions (continued)

Symbol	Field	Description
TAIL3	Max tail	0 = 24 milliseconds. 1 = 32 milliseconds. 2 = 48 milliseconds. 3 = 64 milliseconds.  <b>Note</b> This field should be set just greater than the anticipated worst round-trip tail delay.
REC3	Residual echo control	0 = Cancel only; echo is the result of linear processing; no nonlinear processing is applied. 1 = Suppress residual; residual echo is zeroed; simple nonlinear processing is applied (you might experience “dead air” when talking). 2 = Reserved. 3 = Generate comfort noise (default).
FRZ0	h-register hold	1 = Freezes h-register; used for testing.
HZ0	h-register clear	Sending the channel command with this bit set clears the h-register.
TD3	Modem tone disable	0 = Ignore 2100 Hz modem answer tone. 1 = G.164 mode (bypass canceller if 2100 Hz tone). 2 = R. 3 = G.165 mode (bypass canceller for phase reversing tone only).
ERL0	Echo return loss	0 = 6 decibel (dB). 1 = 3 dB. 2 = 0 dB. 3 = R. Worst echo return loss (ERL) situation in which canceller still works.
HLC1	High level compensation	0 = No attenuation. 1 = 6 dB if clipped. On loud circuits, the received direction can be attenuated 6 dB if clipping is observed.
R0	Reserved	Must be set to 0 to ensure compatibility with future releases.

See [Table 1](#) for a listing of the information types associated with this command.

Use the **show call active voice redirect** command to monitor any active calls that are implementing RTPvt or TBCT.

When a call is no longer active, its record is stored. You can display the record by using the **show call history voice** command.

**Examples**

The following is sample output from the **show call active voice** command for modem relay traffic:

```
Router# show call active voice

Modem Relay Local Rx Speed=0 bps
Modem Relay Local Tx Speed=0 bps
Modem Relay Remote Rx Speed=0 bps
Modem Relay Remote Tx Speed=0 bps
Modem Relay Phy Layer Protocol=v34
Modem Relay Ec Layer Protocol=v14
SPRTInfoFramesReceived=0
SPRTInfoTframesSent=0
SPRTInfoTframesResent=0
SPRTXidFramesReceived=0
SPRTXidFramesSent=0
SPRTTotalInfoBytesReceived=0
SPRTTotalInfoBytesSent=0
SPRTPacketDrops=0
```

The following is sample output from the **show call active voice** command:

```
Router# show call active voice

Total call-legs:2

  GENERIC:
SetupTime=7587246 ms
Index=1
PeerAddress=
PeerSubAddress=
PeerIG=0
PeerIfIndex=0
LogicalIfIndex=0
ConnectTime=7587506
CallDuration=00:00:11
CallState=4
CallOrigin=2
ChargedUnits=0
InfoType=2
TransmitPackets=101
TransmitBytes=1991
ReceivePackets=550
ReceiveBytes=11000
VOIP:
ConnectionId[0x7F8D82A4 0x928E11D5 0x8094FCFB 0x1C38F0FA]
IncomingConnectionId[0x7F8D82A4 0x928E11D5 0x8094FCFB 0x1C38F0FA]
RemoteIPAddress=172.29.248.111
RemoteUDPPort=17394
RoundTripDelay=4 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=TRUE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=cisco
SessionTarget=
OnTimeRvPlayout=10300
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
```

```

GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=70 ms
LoWaterPlayoutDelay=69 ms
ReceiveDelay=69 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
VAD = enabled
CoderTypeRate=g729r8
CodecBytes=20
SignalingType=ext-signal
CallerName=
CallerIDBlocked=False
  GENERIC:
SetupTime=7587246 ms
Index=2
PeerAddress=133001
PeerSubAddress=
PeerId=133001
PeerIfIndex=8
LogicalIfIndex=7
ConnectTime=7587505
CallDuration=00:00:56
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=2
TransmitPackets=2801
TransmitBytes=56020
ReceivePackets=162
ReceiveBytes=3192
TELE:
ConnectionId=[0x7F8D82A4 0x928E11D5 0x8094FCFB 0x1C38F0FA]
IncomingConnectionId=[0x7F8D82A4 0x928E11D5 0x8094FCFB 0x1C38F0FA]
TxDuration=56030 ms
VoiceTxDuration=3210 ms
FaxTxDuration=0 ms
CoderTypeRate=g729r8
NoiseLevel=-44
ACOMLevel=-13
OutSignalLevel=-45
InSignalLevel=-45
InfoActivity=2
ERLLevel=7
EchoCancellerMaxReflector=64
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False

```

**Note**

[Table 1 on page 3](#) describes the significant fields shown in the display.

The following is sample output from the **show call active voice** command for voice traffic over call-agent controlled call legs. Note that call legs for SCCP telephony endpoints, that is, phones controlled by STCAPP, are displayed under the "Call agent controlled call-legs." ("SCCP call-legs" displays call legs for devices that are not telephony endpoints, for example, transcoding and conferencing).

Router# **show call active voice**

Telephony call-legs: 2  
 SIP call-legs: 0  
 H323 call-legs: 0  
 Call agent controlled call-legs: 2  
 SCCP call-legs: 0  
 Multicast call-legs: 0  
 Total call-legs: 4

GENERIC:

SetupTime=1557650 ms  
 Index=1  
 PeerAddress=  
 PeerSubAddress=  
 PeerId=999100  
 PeerIfIndex=14  
 LogicalIfIndex=10  
 ConnectTime=1562040 ms  
 CallDuration=00:01:01 sec  
 CallState=4  
 CallOrigin=2  
 ChargedUnits=0  
 InfoType=speech  
 TransmitPackets=3101  
 TransmitBytes=519564  
 ReceivePackets=3094  
 ReceiveBytes=494572  
 TELE:  
 ConnectionId=[0x11B1860C 0x22D711D7 0x8014E4D4 0x8FD15327]  
 IncomingConnectionId=[0x11B1860C 0x22D711D7 0x8014E4D4 0x8FD15327]  
 CallID=25  
 TxDuration=59670 ms  
 VoiceTxDuration=59670 ms  
 FaxTxDuration=0 ms  
 CoderTypeRate=g711ulaw  
 NoiseLevel=-12  
 ACOMLevel=22  
 OutSignalLevel=-12  
 InSignalLevel=-11  
 InfoActivity=1  
 ERLLevel=22  
 EchoCancellerMaxReflector=2  
 SessionTarget=  
 ImgPages=0  
 CallerName=  
 CallerIDBlocked=False  
 OriginalCallingNumber=  
 OriginalCallingOctet=0x0  
 OriginalCalledNumber=  
 OriginalCalledOctet=0x80  
 OriginalRedirectCalledNumber=  
 OriginalRedirectCalledOctet=0x0  
 TranslatedCallingNumber=  
 TranslatedCallingOctet=0x0  
 TranslatedCalledNumber=  
 TranslatedCalledOctet=0x80  
 TranslatedRedirectCalledNumber=  
 TranslatedRedirectCalledOctet=0x0  
 DSPIdentifier=1/1:1

GENERIC:

SetupTime=1559430 ms  
 Index=1

```

PeerAddress=7702
PeerSubAddress=
PeerId=999100
PeerIfIndex=14
LogicalIfIndex=11
ConnectTime=1562020 ms
CallDuration=00:01:03 sec
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=3151
TransmitBytes=528900
ReceivePackets=3158
ReceiveBytes=503876
TELE:
ConnectionId=[0x0 0x0 0x0 0x0]
IncomingConnectionId=[0x0 0x0 0x0 0x0]
CallID=26
TxDuration=60815 ms
VoiceTxDuration=60815 ms
FaxTxDuration=0 ms
CoderTypeRate=g711ulaw
NoiseLevel=-12
ACOMLevel=28
OutSignalLevel=-12
InSignalLevel=-11
InfoActivity=1
ERLLevel=28
EchoCancellerMaxReflector=2
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
AlertTimepoint=1559430 ms
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=7701
TranslatedCallingOctet=0x0
TranslatedCalledNumber=7702
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwOutpulsedCalledNumber=7702
GwOutpulsedCalledOctet3=0x0
GwOutpulsedCallingNumber=7701
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x0
DSPIdentifier=1/1:2

GENERIC:
SetupTime=1562040 ms
Index=1
PeerAddress=
PeerSubAddress=
PeerId=0
PeerIfIndex=0
LogicalIfIndex=0
ConnectTime=0 ms
CallDuration=00:00:00 sec

```

## show call active voice

```

CallState=2
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=3215
TransmitBytes=512996
ReceivePackets=3208
ReceiveBytes=512812
VOIP:
ConnectionId[0x0 0x0 0x0 0x0]
IncomingConnectionId[0x0 0x0 0x0 0x0]
CallID=27
RemoteIPAddress=0.0.0.0
RemoteUDPPort=17718
RemoteSignallingIPAddress=0.0.0.0
RemoteSignallingPort=0
RemoteMediaIPAddress=10.2.6.10
RemoteMediaPort=17718
RoundTripDelay=0 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=FALSE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=other
ProtocolCallId=
SessionTarget=
OnTimeRvPlayout=60640
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=105 ms
LoWaterPlayoutDelay=105 ms
TxPakNumber=3040
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=60815
TxVoiceDuration=60815
RxPakNumber=3035
RxSignalPak=0
RxDuration=0
TxVoiceDuration=60690
VoiceRxDuration=60640
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=105
PlayDelayMin=105
PlayDelayMax=105
PlayDelayClockOffset=-1662143961
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0
PlayErrBufferOverflow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=-12

```

```
InSignalLevel=-11
LevelTxPowerMean=0
LevelRxPowerMean=-115
LevelBgNoise=0
ERLLevel=28
ACOMLevel=28
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
PlayoutMode = undefined
PlayoutInitialDelay=0 ms
ReceiveDelay=105 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = disabled
CoderTypeRate=g711ulaw
CodecBytes=160
Media Setting=flow-around

Modem passthrough signaling method is nse:
Buffer Fill Events = 0
Buffer Drain Events = 0
Percent Packet Loss = 0
Consecutive-packets-lost Events = 0
Corrected packet-loss Events = 0
Last Buffer Drain/Fill Event = 0sec
Time between Buffer Drain/Fills = Min 0sec Max 0sec

CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=

GENERIC:
SetupTime=1562040 ms
Index=2
PeerAddress=
PeerSubAddress=
PeerId=0
PeerIfIndex=0
LogicalIfIndex=0
ConnectTime=0 ms
CallDuration=00:00:00 sec
CallState=2
CallOrigin=1
ChargedUnits=0
InfoType=speech
```

## show call active voice

```

TransmitPackets=3380
TransmitBytes=540332
ReceivePackets=3386
ReceiveBytes=540356
VOIP:
ConnectionId[0x0 0x0 0x0 0x0]
IncomingConnectionId[0x0 0x0 0x0 0x0]
CallID=28
RemoteIPAddress=0.0.0.0
RemoteUDPPort=18630
RemoteSignallingIPAddress=0.0.0.0
RemoteSignallingPort=0
RemoteMediaIPAddress=10.2.6.10
RemoteMediaPort=18630
RoundTripDelay=0 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=FALSE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=FALSE

SessionProtocol=other
ProtocolCallId=
SessionTarget=
OnTimeRvPlayout=63120
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=105 ms
LoWaterPlayoutDelay=105 ms
TxPakNumber=3158
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=63165
TxVoiceDuration=63165
RxPakNumber=3164
RxSignalPak=0
RxDuration=0
TxVoiceDuration=63165
VoiceRxDuration=63120
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=105
PlayDelayMin=105
PlayDelayMax=105
PlayDelayClockOffset=957554296
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0
PlayErrBufferOverflow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=-12
InSignalLevel=-11
LevelTxPowerMean=0
LevelRxPowerMean=-114
LevelBgNoise=0

```

```

ERLLevel=22
ACOMLevel=22
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
PlayoutMode = undefined
PlayoutInitialDelay=0 ms
ReceiveDelay=105 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = disabled
CoderTypeRate=g711ulaw
CodecBytes=160
Media Setting=flow-around

Modem passthrough signaling method is nse:
Buffer Fill Events = 0
Buffer Drain Events = 0
Percent Packet Loss = 0
Consecutive-packets-lost Events = 0
Corrected packet-loss Events = 0
Last Buffer Drain/Fill Event = 0sec
Time between Buffer Drain/Fills = Min 0sec Max 0sec

CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x0
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x0
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=
Telephony call-legs: 2
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 2
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 4

```

The following is sample output from the **show call active voice** command for fax relay traffic:

```

Router# show call active voice

Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 1
MGCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 1

```

```
show call active voice
```

```

GENERIC:
SetupTime=1049400 ms
Index=2
PeerAddress=52930
PeerSubAddress=
PeerId=82
PeerIfIndex=222
LogicalIfIndex=0
ConnectTime=105105
CallDuration=00:00:59
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=10
TransmitPackets=1837
TransmitBytes=29764
ReceivePackets=261
ReceiveBytes=4079
VOIP:
ConnectionId[0xEB630F4B 0x9F5E11D7 0x8008CF18 0xB9C3632]
IncomingConnectionId[0xEB630F4B 0x9F5E11D7 0x8008CF18 0xB9C3632]
RemoteIPAddress=10.7.95.3
RemoteUDPPort=16610
RemoteSignallingIPAddress=10.7.95.3
RemoteSignallingPort=1720
RemoteMediaIPAddress=10.7.95.3
RemoteMediaPort=16610
RoundTripDelay=13 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=TRUE

AnnexE=FALSE

Separate H245 Connection=FALSE

H245 Tunneling=TRUE

SessionProtocol=cisco
ProtocolCallId=
SessionTarget=ipv4:10.7.95.3
OnTimeRvPlayout=1000
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=110 ms
LoWaterPlayoutDelay=70 ms
ReceiveDelay=70 ms
LostPackets=0
EarlyPackets=1
LatePackets=0
VAD = enabled
CoderTypeRate=t38
CodecBytes=40
Media Setting=flow-through
AlertTimepoint=104972
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=4085550130
OriginalCallingOctet=0x0
OriginalCalledNumber=52930
OriginalCalledOctet=0xE9
OriginalRedirectCalledNumber=

```

```

OriginalRedirectCalledOctet=0x7F
TranslatedCallingNumber=4085550130
TranslatedCallingOctet=0x0
TranslatedCalledNumber=52930
TranslatedCalledOctet=0xE9
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0xFF
GwReceivedCalledNumber=52930
GwReceivedCalledOctet3=0xE9
GwOutpulsedCalledNumber=52930
GwOutpulsedCalledOctet3=0xE9
GwReceivedCallingNumber=4085452930
GwReceivedCallingOctet3=0x0
GwReceivedCallingOctet3a=0x80
GwOutpulsedCallingNumber=4085550130
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x80
Username=
FaxRelayMaxJitterBufDepth = 0 ms
FaxRelayJitterBufOverflow = 0
FaxRelayHSmodulation = 0
FaxRelayNumberOfPages = 0
Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 1
MGCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 1

```

**Note**

Table 1 on page 3 and Table 49 on page 1404 describe fields in the display.

The following is sample output from the **show call active voice brief** command:

```
Router# show call active voice brief
```

```

<ID>: <CallID> <start>hs.<index> +<connect> pid:<peer_id> <dir> <addr> <state>
dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes>
IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
delay:<last>/<min>/<max>ms <codec>
media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>
MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
last <buf event time>s dur:<Min>/<Max>s
FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
<codec> (payload size)
ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
<codec> (payload size)
Tele <int> (callID) [channel_id] tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l>
i/o:<l>/<l> dBm
MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops>
speeds (bps): local <rx>/<tx> remote <rx>/<tx>
Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
bw: <req>/<act> codec: <audio>/<video>
tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>

```

```

Total call-legs:2
1269 :7587246hs.1 +260 pid:0 Answer active
dur 00:07:14 tx:590/11550 rx:21721/434420
IP 172.29.248.111:17394 rtt:3ms pl:431850/0ms lost:0/0/0 dela
y:69/69/70ms g729r8

```

```
1269 :7587246hs.2 +259 pid:133001 Originate 133001 active
```

■ **show call active voice**

```
dur 00:07:14 tx:21717/434340 rx:590/11550
Tele 1/0:1 (2):tx:434350/11640/0ms g729r8 noise:-44 acom:-19
i/0:-45/-45 dBm
```

The following is an example of the **show call active voice** command using the **echo-canceller** keyword. The number 9 represents the hexadecimal ID of an active voice call.

```
Router# show call active voice echo-canceller 9
```

```
ACOM=-65 ERL=45
Echo canceller control words=6C 0
Bypass=OFF Tail=64 Residual ecan=Comfort noise
Freeze=OFF Modem tone disable=Ignore 2100Hz tone
Worst ERL=6 High level compensation=OFF
Max amplitude reflector (in msec)=5
Ecan version = 8180
```

The following is sample output from the **show call active voice echo-canceller** command for a call with a hexadecimal ID of 10.

```
Router# show call active voice echo-canceller 10
```

```
ACOM=-15 ERL=7
Echo canceller control words=6C 0
Bypass=OFF Tail=64 Residual ecan=Comfort noise
Freeze=OFF Modem tone disable=Ignore 2100Hz tone
Worst ERL=6 High level compensation=OFF
Max amplitude reflector (in msec)=64
```

The call ID number (which is 10 in the previous example) changes with every new active call. When an active call is up, you must enter the **show call active voice brief** command to obtain the call ID number. The call ID must be converted to hex if you want to use the **show call active voice echo-canceller x** command ( $x$  = call ID converted to hex).

The following are call ID examples converted to hex (generally incremented by 2):

Decimal	Hex
2	2
4	4
6	6
8	8
10	A
12	C

Alternatively, you can use the **show voice call status** command to obtain the call ID. The call ID output is already in hex form when you use this command:

```
Router# show voice call status
```

```
CallID      CID  ccVdb      Port      DSP/Ch  Called #  Codec   Dial-peers
0x1         11CE 0x02407B20 1:0.1     1/1     1000     g711ulaw 2000/1000
```

The following is sample output from the **show call active voice redirect** command using the **tbct** keyword:

```
Router# show call active voice redirect tbct
```

```
TBCT:
```

```
Maximum no. of TBCT calls allowed:No limit
```

```
Maximum TBCT call duration:No limit
```

```
Total number TBCT calls currently being monitored = 1
```

```
ctrl name=T1-2/0, tag=13, call-ids=(7, 8), start_time=*00:12:25.985 UTC Mon Mar 1 1993
```

[Table 5](#) describes significant fields shown in the **show call active voice redirect** display.

**Table 5** *show call active voice redirect Field Descriptions*

Field	Description
Maximum no. of TBCT calls allowed	Maximum number of calls that can use TBCT as defined by the <b>tbct max calls</b> command.
Maximum TBCT call duration	Maximum length allowed for a TBCT call as defined by the <b>tbct max call-duration</b> command.
Total number TBCT calls currently being monitored	Total number of active TBCT calls.
ctrl name	Name of the T1 controller where the call originated.
tag	Call tag number that identifies the call.
call-ids	Numbers that uniquely identify the call legs.
start_time	Time, in hours, minutes, and seconds, when the redirected call began.

#### Related Commands

Command	Description
<b>show call active fax</b>	Displays call information for fax transmissions that are in progress.
<b>show call history</b>	Displays the call history table.
<b>show call-router routes</b>	Displays the dynamic routes in the cache of the BE.
<b>show call-router status</b>	Displays the Annex G BE status.
<b>show dial-peer voice</b>	Displays configuration information for dial peers.
<b>show num-exp</b>	Displays how the number expansions are configured in VoIP.
<b>show voice call status</b>	Displays the call status for voice ports on the Cisco router or concentrator.
<b>show voice port</b>	Displays configuration information about a specific voice port.

# show call application voice

To display information about voice applications, use the **show call application voice** command in EXEC mode.

**show call application voice** [*name* | **summary**]

Syntax Description	
<i>name</i>	(Optional) Name of the desired voice application. Output displays information about that application.
<b>summary</b>	(Optional) Output displays a one-line summary of each voice application.

**Defaults** If both the *name* argument and **summary** keyword are omitted, command output displays detailed information about all IVR applications.

**Command Modes** EXEC

Command History	Release	Modification
	11.3(6)NA2	This command was introduced.
	12.0(3)T	This command was integrated into Cisco IOS Release 12.0(3)T.
	12.1(5)T	This command was implemented on the Cisco AS5800.
	12.1(5)XM2	This command was implemented on the Cisco AS5350 and Cisco AS5400.
	12.2(2)XB	This command was modified to support VoiceXML applications.
	12.2(2)XB1	This command was implemented on the Cisco AS5850.
	12.2(4)XM	This command was implemented on the Cisco 1750 and Cisco 1751. This command was not supported on any other platforms in this release.
	12.2(8)T	This command was implemented on the Cisco 1751, Cisco 2600 series, Cisco 3600 series, Cisco 3725, Cisco 3745, and Cisco 7200.
	12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T for VoiceXML applications. This command is supported on the Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco AS5850 in this release.
	12.3(14)T	New output relating to the SCCP Telephony Control Application (STCAPP) was added.

**Usage Guidelines** The **show call application voice** command displays a detailed description of each configured application.

If the name of a specific application is entered, the command displays detailed information about only that application.

If the **summary** keyword is entered, the command displays a one-line summary about each application.

If STCAPP is enabled, the **summary** command displays STCAPP as an available call application.

If an asterisk is displayed next to the application name when the **summary** keyword is used, the application is configured, but not running. Normally this is because the application was not successfully loaded, for example:

```
name          description
*vapptest2    flash:helloworld.vxml
```

TCL scripts and VoiceXML documents can be stored in any of the following locations: TFTP, FTP, or HTTP servers, Flash memory of the gateway, or the removable disks of the Cisco 3600 series. The audio files that they use can be stored in any of these locations and on RTSP servers.

## Examples

The following example shows the output for the session TCL script:

```
Router# show call application voice session

Application session
  The script is compiled into the image
  It has 0 calls active.
  Interpreted by infrastructure version 2.0

The TCL Script is:
-----
# app_session.tcl
#-----
# August 1999, Saravanan Shanmugham
#
# Copyright (c) 1998, 1999, 2000, 2001 by cisco Systems, Inc.
# All rights reserved.
#-----
#
# This tcl script mimics the default SESSION app
#
# If DID is configured, just place the call to the dnis
# Otherwise, output dial-tone and collect digits from the
# caller against the dial-plan.
#
# Then place the call. If successful, connect it up, otherwise
# the caller should hear a busy or congested signal.

# The main routine just establishes the statemachine and then exits.
# From then on the system drives the statemachine depending on the
# events it receives and calls the appropriate tcl procedure

#-----
# Example Script
#-----

proc init { } {
    global param

    set param(interruptPrompt) true
    set param(abortKey) *
    set param(terminationKey) #
}

proc act_Setup { } {
    global dest
    global beep
```

```

set beep 0

if { [infotag get leg_isdid] } {
    set dest [infotag get leg_dnis]
    leg proceeding leg_incoming
    leg setup $dest callInfo leg_incoming
    fsm setstate PLACECALL
} else {
    leg setupack leg_incoming
    playtone leg_incoming tn_dial

    set param(dialPlan) true
    leg collectdigits leg_incoming param
}

}

proc act_GotDest { } {
    global dest

    set status [infotag get evt_status]

    if { $status == "cd_004" } {
        set dest [infotag get evt_dcdigits]
        leg proceeding leg_incoming
        leg setup $dest callInfo leg_incoming
    } else {
        puts "\nCall [infotag get con_all] got event $status collecting destina"
        call close
    }
}

proc act_CallSetupDone { } {
    global beep

    set status [infotag get evt_status]

    if { $status == "ls_000" } {

        set creditTimeLeft [infotag get leg_settlement_time leg_all]

        if { ($creditTimeLeft == "unlimited") ||
            ($creditTimeLeft == "uninitialized") } {
            puts "\n Unlimited Time"
        } else {
            # start the timer for ...
            if { $creditTimeLeft < 10 } {
                set beep 1
                set delay $creditTimeLeft
            } else {
                set delay [expr $creditTimeLeft - 10]
            }
            timer start leg_timer $delay leg_incoming
        }
    } else {
        puts "Call [infotag get con_all] got event $status collecting destinati"
        call close
    }
}

```

```

proc act_Timer { } {
    global beep
    global incoming
    global outgoing

    set incoming [infotag get leg_incoming]
    set outgoing [infotag get leg_outgoing]

    if { $beep == 0 } {
        #insert a beep ...to the caller
        connection destroy con_all
        set beep 1
    } else {
        connection destroy con_all
        fsm setstate LASTWARN
    }
}

proc act_LastWarn { } {
    media play leg_incoming flash:out_of_time.au
}

proc act_Destroy { } {
    media play leg_incoming flash:beep.au
}

proc act_Beeped { } {
    global incoming
    global outgoing

    connection create $incoming $outgoing
}

proc act_ConnectedAgain { } {
    timer start leg_timer 10 leg_incoming
}

proc act_Ignore { } {
    # Dummy
    puts "Event Capture"
}

proc act_Cleanup { } {
    call close
}

init

#-----
#   State Machine
#-----
set fsm(any_state,ev_disconnected)  "act_Cleanup          same_state"

set fsm(CALL_INIT,ev_setup_indication) "act_Setup          GETDEST"

set fsm(GETDEST,ev_collectdigits_done)  "act_GotDest        PLACECALL"

set fsm(PLACECALL,ev_setup_done)        "act_CallSetupDone  CALLACTIVE"

set fsm(CALLACTIVE,ev_leg_timer)        "act_Timer          INSERTBEEP"
set fsm(INSERTBEEP,ev_destroy_done)     "act_Destroy        same_state"
set fsm(INSERTBEEP,ev_media_done)       "act_Beeped         same_state"
set fsm(INSERTBEEP,ev_create_done)      "act_ConnectedAgain CALLACTIVE"

```

## show call application voice

```

set fsm(LASTWARN, ev_destroy_done)      "act_LastWarn      CALLDISCONNECT"

set fsm(CALLACTIVE, ev_disconnected)   "act_Cleanup       CALLDISCONNECT"
set fsm(CALLDISCONNECT, ev_disconnected) "act_Cleanup       same_state"
set fsm(CALLDISCONNECT, ev_media_done)  "act_Cleanup       same_state"
set fsm(CALLDISCONNECT, ev_disconnect_done) "act_Cleanup       same_state"
set fsm(CALLDISCONNECT, ev_leg_timer)   "act_Cleanup       same_state"

fsm define fsm CALL_INIT

```

The following is sample output for the **summary** keyword:

```
Router# show call application voice summary
```

name	description
session	Basic app to do DID, or supply dialtone.
fax_hop_on	Script to talk to a fax redialer
clid_authen	Authenticate with (ani, dnis)
clid_authen_collect	Authenticate with (ani, dnis), collect if that fails
clid_authen_npw	Authenticate with (ani, NULL)
clid_authen_col_npw	Authenticate with (ani, NULL), collect if that fails
clid_col_npw_3	Authenticate with (ani, NULL), and 3 tries collecting
clid_col_npw_npw	Authenticate with (ani, NULL) and 3 tries without pw
DEFAULT	Default system session application
lib_off_app	Libretto Offramp

TCL Script Version 2.0 supported.

TCL Script Version 1.1 supported.

Voice Browser Version 2.0 for VoiceXML 1.0 & 2.0 supported.

The following is sample output for the **summary** keyword when the SCCP Telephony Control Application (STCAPP) is enabled:

```
Router# show call application voice summary
```

```
SERVICES (standalone applications):
```

name	type	description
ipsla-responder	Tcl Script	builtin:app_test_rcvr_script.tcl
clid_authen	Tcl Script	builtin:app_clid_authen_script.tcl
clid_col_npw_npw	Tcl Script	builtin:app_clid_col_npw_npw_script.tcl
DEFAULT	C Script	builtin:Session_Service.C
CTAPP	C Script	builtin:CallTreatment_Service.C
clid_authen_col_npw	Tcl Script	builtin:app_clid_authen_col_npw_script.tcl
fax_hop_on	Tcl Script	builtin:app_fax_hop_on_script.tcl
ipsla-testcall	Tcl Script	builtin:app_test_place_script.tcl
clid_authen_npw	Tcl Script	builtin:app_clid_authen_npw_script.tcl
session	Tcl Script	builtin:app_session_script.tcl
clid_authen_collect	Tcl Script	builtin:app_clid_authen_collect_script.tcl
clid_col_npw_3	Tcl Script	builtin:app_clid_col_npw_3_script.tcl
lib_off_app	CCAPI	Libretto Offramp
DEFAULT.C.OLD	CCAPI	Obsolete system session application
stcapp	CCAPI	SCCP Call Control Application
MGCPAPP	CCAPI	MGCP Application

The following is sample output for the *stcapp* argument when the SCCP Telephony Control Application (STCAPP) is enabled:

```
Router# show call application voice stcapp
```

```
App Status:      Active
CCM Status:      UP
```

```

CCM Group:                2
Registration Mode:        CCM
Total Devices:            5
Total Calls in Progress: 0
Total Call Legs in Use:  0

```

The following is sample output from the **show call application voice** command for a VoiceXML application named “vapptest1”:

```

Router> show call application voice vapptest1

VXML Application vapptest1
  URL=flash:demo0.vxml
  Security not trusted
  No languages configured
  It has: 0 calls active.
    0 incoming calls
    0 calls handed off to it
    0 call transfers initiated
    0 pages loaded, 0 successful
    0 prompts played
    0 recorded messages
  Interpreted by Voice Browser Version 2.0 for VoiceXML 1.0 & 2.0.

The VXML Script is:
-----
<?xml version="1.0"?>
<vxml version="1.0">

  <form>
    <block>
      <audio src="flash:demo0.au"/>
    </block>
  </form>
</vxml>

```

Table 6 describes the fields shown in the **show call application voice** display:

**Table 6** show call application voice Field Descriptions

Field	Description
URL	Location of the document used by the application.
It has: <i>n</i> calls active.	Number of calls that are currently using this application.
incoming calls	Number of incoming PSTN or IP calls that invoked this application.
calls handed off to it	Number of calls that were handed off to this application by another TCL or VoiceXML application.
call transfers initiated	Number of call transfers that were initiated by this application.
pages loaded	Number of VoiceXML pages that were loaded by the application.
successful	Number of VoiceXML pages that were successfully completed.
prompts played	Number of audio prompts that were played by the application.
recorded messages	Number of audio recordings made by the VoiceXML application.
Interpreted by	Programming language used by the application.
The TCL or VoiceXML Script is	Content of the VoiceXML document or TCL script.

## ■ show call application voice

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 the application.
	<b>call application voice load</b>	Reloads the designated TCL script or VoiceXML document.

## show ccm-manager

To display a list of Cisco CallManager servers and their current status and availability, use the **show ccm-manager** command in privileged EXEC mode.

```
show ccm-manager [backhaul | config-download | fallback-mgcp | hosts | music-on-hold |
redundancy | download-tones [c1 | c2]]
```

Syntax	Description
<b>backhaul</b>	(Optional) Information about the backhaul link.
<b>config-download</b>	(Optional) Displays information about the status of Media Gateway Control Protocol (MGCP) and Skinny Client Control Protocol (SCCP) configuration download.
<b>fallback-mgcp</b>	(Optional) Displays the status of the MGCP gateway fallback feature.
<b>hosts</b>	(Optional) Displays a list of each configured Cisco CallManager server in the network, together with its operational status and host IP address.
<b>music-on-hold</b>	(Optional) Displays information about all the multicast music-on-hold (MOH) sessions in the gateway at any given point in time.
<b>redundancy</b>	(Optional) Displays failover mode and status information for hosts, including the redundant link port, failover interval, keepalive interval, MGCP traffic time, switchover time, and switchback mode.
<b>download-tones [c1   c2]</b>	(Optional) Displays custom tones downloaded to the gateway. The custom tone value of c1 or c2 specifies which tone information to display.

**Defaults** If none of the optional keywords is specified, information related to all keywords is displayed.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(3)T	This feature was introduced on the Cisco CallManager Version 3.0 and Cisco VG200.
	12.2(2)XA	This command was implemented on the Cisco 2600 series and Cisco 3600 series.
	12.2(2)XN	This command was modified to provide enhanced MGCP voice gateway interoperability to Cisco CallManager Version 3.1 for the Cisco 2600 series, Cisco 3600 series, and Cisco VG200.
	12.2(11)T	This command was integrated into Cisco IOS Release 12.2(2)11 and the Cisco CallManager Version 3.2. It was implemented on the Cisco IAD2420 series.
	12.2(15)ZJ	The <b>download-tones [c1   c2]</b> keyword was added for the following platforms: Cisco 2610XM, Cisco 2611XM, Cisco 2620XM, Cisco 2621XM, Cisco 2650XM, Cisco 2651XM, Cisco 2691, Cisco 3640A, Cisco 3660, Cisco 3725, and Cisco 3745.

Release	Modification
12.3(4)T	The <b>download-tones [c1   c2]</b> keyword was integrated into Cisco IOS Release 12.3(4)T.
12.3(14)T	New output relating to SCCP auto-configuration was added.

### Usage Guidelines

Use the **show ccm-manager config-download** command to determine the status of Cisco CallManager servers and the automatic download information and statistics.

### Examples

The following sample output shows the configured amplitudes, frequencies, and cadences of custom tone 1, Hong Kong:

```
Router# show ccm-manager download-tones C1
!
Custom Tone 1 : Hong Kong
Pulse dial:normal, Percent make:35%, DTMF low Amp.= 65424, high Amp.= 65446, Pcm:u-Law
FXS FXO E&M FXS FXO E&M
Dual Tone DR NF FOF FOS AOF AOF AOF AOS AOS AOS ONTF OTTF ONTS OFTS ONTT OFTT ONT4 OFT4
(optional) FOF2 FOS2 FOF3 FOS3 FOF4 FOS4 FOT FO4 AOT AO4 RCT1 RCT2 RCT3 RCT4
BUSY 0 2 480 620 -120 -120 -120 -120 -120 -120 500 500 0 0 0 0 0 0
RING_BACK 0 2 440 520 -120 -120 -120 -120 -120 -120 400 200 400 3000
CONGESTION 0 2 480 620 -200 -200 -200 -240 -240 -240 250 250 0 0 0
NUMBER_UNOBTAINABLE 0 2 480 620 -120 -120 -120 -120 -120 -120 65535 0 0 0
DIAL_TONE 0 2 350 440 -150 -150 -150 -150 -150 -150 65535 0 0 0
DIAL_TONE2 0 2 350 440 -150 -150 -150 -150 -150 -150 65535 0 0 0
OUT_OF_SERVICE 0 1 950 0 -150 -150 -150 0 0 0 330 330 0 0 0
ADDR_ACK 0 1 600 0 -240 -240 -240 0 0 0 125 125 125 65535
DISCONNECT 0 1 600 0 -150 -150 -150 0 0 0 330 330 330 65535
OFF_HOOK_NOTICE 0 2 1400 2040 -240 -240 -240 -240 -240 -240 100 100 0 0 0
OFF_HOOK_ALERT 0 2 1400 2040 -240 -240 -240 -240 -240 -240 100 100 0 0 0
WAITING 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CONFIRM 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CNFWRN_J 0 1 950 0 -170 -170 -190 0 0 0 100 100 100 65535
CNFWRN_D 0 1 600 0 -170 -170 -190 0 0 0 100 100 100 65535
STUTT_DIALTONE 0 2 350 440 -150 -150 -150 -150 -150 -150 100 100 100 100
100 100 65535 0
PERM_SIG_TONE 0 1 480 0 -170 -170 -170 0 0 0 65535 0 0 0
WAITING1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
WAITING2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
WAITING3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
WAITING4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MSGWAIT_IND 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
OFF_HOOK_WARN 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Sequence Tone DR NF F1C1 F2C1 AOF AOS C1ONT C1OFT C2ONT C2OFT C3ONT C3OFT
C4ONT C4OFT F1C2 F2C2 F1C3 F2C3 F1C4 F2C4
INTERCEPT 0 0 0 0 0 0 0 0 0 0 0 0 0 0
TONE_ON_HOLD 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NO_CIRCUIT 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

#### Legend:

DR: direction NF: number of frequency FO<F,S,T,4>: frequency of<1st,2nd,3rd,4th>

AO<F,S,T,4>: amplitude of<1st,2nd,3rd,4th>

FOF<1-4>: frequency of 1st, cadence<1-4> FOS<1-4>: frequency of 2nd, cadence<1-4>

RCT<1-4>: repeat count for cadence<1-4> F(1-4>C<1-4> : frequency<1-4> of cadence<1-4>

C<1-4>ONT: cadence<1-4> on time C<1-4>OFT: cadence<1-4> off time

[Table 7](#), [Table 8](#), and [Table 9](#) give descriptions of significant fields once the tones are automatically downloaded to the gateway.

**Table 7** *show ccm-manager config-download download-tones Significant Output Fields*

Field	Description
Percent make	Pulse ratio in percent of make.
DTMF low Amp.	Low frequency level.
high Amp.	High frequency level.
Pcm	Pulse Code Modulation (u-law or a-law).

**Table 8** *show ccm-manager config-download download-tones Output Fields for Dual Tones*

Field of Dual Tone	Description
DR	Direction to PSTN (0) or Packet Network(1).
NF	Number of Frequency (from 1 to 4).
FOF	Frequency of First component (in Hz).
FXS AOF	Amplitude of First component (from 1 to 65535 = +3 dBm0) for FXS.
FXO AOF	Amplitude of First component (from 1 to 65535 = +3 dBm0) for FXO.
E&M AOF	Amplitude of First component (from 1 to 65535 = +3 dBm0) for E&M.
FXS AOS	Amplitude of Second component (from 1 to 65535 = +3 dBm0) for FXS.
FXO AOS	Amplitude of Second component (from 1 to 65535 = +3 dBm0) for FXO.
E&M AOS	Amplitude of Second component (from 1 to 65535 = +3 dBm0) for E&M.
ONTF	On time, time the tone is generated (mSec) for 1st frequency.
OFTF	Off time, silence time (mSec) for 1st frequency.
ONTS	On time, time the tone is generated (mSec) for 2nd frequency.
OFTS	Off time, silence time (mSec) for 2nd frequency.
ONTT	On time, time the tone is generated (mSec) for 3rd frequency.
OFTT	Off time, silence time (mSec) for 3rd frequency.
ONT4	On time, time the tone is generated (mSec) for 4th frequency.
OFT4	Off time, silence time (mSec) for 4th frequency.
FOF2	Frequency of First component for 2nd cadence.
FOS2	Frequency of Second component for 2nd cadence.
FOF3	Frequency of First component for 3rd cadence.
FOS3	Frequency of Second component for 3rd cadence.
FOF4	Frequency of First component for 4th cadence.
FOS4	Frequency of Second component for 4th cadence.
FOT	Frequency of Third component (in Hz).
FO4	Frequency of Fourth component (in Hz).
AOT	Amplitude of Third component (from 1 to 65535 = +3dBm0).
AO4	Amplitude of Fourth component (from 1 to 65535 = +3dBm0).
RCT1	Number of repeat for 1st cadence.

**Table 8** *show ccm-manager config-download download-tones Output Fields for Dual Tones*

RCT2	Number of repeat for 2nd cadence.
RCT3	Number of repeat for 3rd cadence.
RCT4	Number of repeat for 4th cadence.

**Table 9** *show ccm-manager download-tones Output Fields for Sequence Tones*

Field of Sequence Tone	Description
DR	Direction to PSTN (0) or Packet Network (1).
NF	Number of Frequency (from 1 to 4).
F1C1	Frequency 1 of Cadence 1.
F2C1	Frequency 2 of Cadence 1.
AOF	Amplitude of First component (from 1 to 65535).
AOS	Amplitude of Second component (from 1 to 65535).
C1ONT	Cadence 1 On Time.
C1OFT	Cadence 1 Off Time.
C2ONT	Cadence 2 On Time.
C2OFT	Cadence 2 Off Time.
C3ONT	Cadence 3 On Time.
C3OFT	Cadence 3 Off Time.
C4ONT	Cadence 4 On Time.
C4OFT	Cadence 4 Off Time.
F1C2	Frequency 1 of Cadence 2.
F2C2	Frequency 2 of Cadence 2.
F1C3	Frequency 1 of Cadence 3.
F2C3	Frequency 2 of Cadence 3.
F1C4	Frequency 1 of Cadence 4.
F2C4	Frequency 2 of Cadence 4.

The following is sample output from the **show ccm-manager** command for displaying the status and availability of both the primary and the backup Cisco CallManager server.

Router# **show ccm-manager**

```

MGCP Domain Name: c3660A.cisco.com
Priority      Status      Host
=====
Primary      Registered  IOS-38 (172.20.71.38)
First Backup  Backup Ready  ios-44 (172.20.71.44)
Second Backup  None

Current active Call Manager: 172.20.71.38
Backhaul/Redundant link port: 2428
Failover Interval: 30 seconds
Keepalive Interval: 15 seconds
Last keepalive sent: 03:06:24 (elapsed time: 00:00:06)

```

```

Last MGCP traffic time:          03:06:24 (elapsed time: 00:00:06)
Last failover time:             02:56:35 from (172.20.71.44)
Switchback mode:               Graceful
MGCP Fallback mode:            Enabled/OFF
Last MGCP Fallback start time:  00:00:00
Last MGCP Fallback end time:    00:00:00

PRI Backhaul Link info:
  Link Protocol:                TCP
  Remote Port Number:           2428
  Remote IP Address:            172.20.71.38
  Current Link State:           OPEN
  Statistics:
    Packets recvd:              1
    Recv failures:              0
    Packets xmitted:            3
    Xmit failures:              0
  PRI Ports being backhauled:
    Slot 1, port 1
MGCP Download Tones:           Enabled

Configuration Auto-Download Information
=====
Current version-id: {1645327B-F59A-4417-8E01-7312C61216AE}
Last config-downloaded:00:00:49
Current state: Waiting for commands
Configuration Download statistics:
  Download Attempted            : 6
  Download Successful           : 6
  Download Failed               : 0
  Configuration Attempted      : 1
  Configuration Successful      : 1
  Configuration Failed(Parsing): 0
  Configuration Failed(config) : 0
Last config download command: New Registration
Configuration Error History:
FAX mode: cisco

```

Table 10 describes significant fields shown in this output.

**Table 10** show ccm-manager Field Descriptions

Field	Description
MGCP Domain Name ( <i>system</i> )	System used in the Internet for translating domain names of network nodes into IP addresses.
Priority	Priority of the Cisco CallManager servers present in the network. Possible priorities are primary, first backup, and second backup.
Status	Current usage of the Cisco CallManager server. Possible values are registered, idle, backup polling, and undefined.
Host	Host IP address of the Cisco CallManager server.
Current active Call Manager	Active Cisco CallManager server. This field can be any one of the following: primary, first backup, and second backup.
Backhaul/Redundant link port	Port that the Cisco CallManager server is to use.
Failover Interval	Maximum amount of time that can elapse without the gateway receiving messages from the currently active Cisco Call Manager before the gateway switches to the backup Cisco Call Manager.

**Table 10** *show ccm-manager Field Descriptions (continued)*

Field	Description
Keepalive Interval	If the gateway has not received any messages from the currently active Cisco CallManager server within the specified amount of time, the gateway sends a keepalive message to the Cisco CallManager server to determine if it is operational.
Last keepalive sent	Time at which the last keepalive message was sent.
Last MGCP traffic time	Time at which the last MGCP traffic message was sent.
Switchback mode	Displays the switchback mode configuration that determines when the primary Cisco CallManager server is used if it becomes available again while a backup Cisco CallManager server is being used.  Values that can appear in this field include graceful, immediate, schedule-time, and uptime-delay.
MGCP Fallback mode	Displays the MGCP fallback mode configuration. If “Not Selected” displays, then fallback is not configured. If “Enabled/OFF” displays, then fallback is configured but not in effect. If “Enabled/ON” displays, then fallback is configured and in effect.
Last MGCP Fallback start time	Start time stamp of the last fallback.
Lasts MGCP Fallback end time	End time stamp of the last fallback.

The following is sample output from the **show ccm-manager config-download** command showing the status of the SCCP download:

```
Router# show ccm-manager config-download

Configuration Auto-Download Information
=====
Current version-id:{4171F93A-D8FC-49D8-B1C4-CE33FA8095BF}
Last config-downloaded:00:00:47
Current state:Waiting for commands
Configuration Download statistics:
    Download Attempted           :6
    Download Successful          :6
    Download Failed              :0
    Configuration Attempted     :1
    Configuration Successful     :1
    Configuration Failed(Parsing):0
    Configuration Failed(config) :0
Last config download command:New Registration

SCCP auto-configuration status
=====
Registered with Call Manager: No
Local interface: FastEthernet0/0 (000c.8522.6910)
Current version-id: {D3A886A2-9BC9-41F8-9DB2-0E565CF51E5A}
Current config applied at: 04:44:45 EST Jan 9 2003
Gateway downloads succeeded: 1
Gateway download attempts: 1
Last gateway download attempt: 04:44:45 EST Jan 9 2003
Last successful gateway download: 04:44:45 EST Jan 9 2003
Current TFTP server: 10.2.6.101
Gateway resets: 0
Gateway restarts: 0
```

```

Managed endpoints: 6
Endpoint downloads succeeded: 6
Endpoint download attempts: 6
Last endpoint download attempt: 04:44:45 EST Jan 9 2003
Last successful endpoint download: 04:44:45 EST Jan 9 2003
Endpoint resets: 0
Endpoint restarts: 0

Configuration Error History:
sccp ccm CCM-PUB7 identifier 1
end

controller T1 2/0no shut

controller T1 2/0no shut

controller T1 2/0no shut

isdn switch-type primary-ni
end

```

Table 11 describes significant fields shown in this output.

**Table 11** *show ccm-manager config-download Field Descriptions*

Field	Description
Current state	Current configuration state.
Download Attempted	Number of times the gateway has tried to download the configuration file. The number of successes and failures is displayed.
Configuration Attempted	Number of times the gateway has tried to configure the gateway based on the configuration file. The number of successes and failures is displayed.
Configuration Error History	Displays SCCP auto-configuration errors.
Endpoint download attempts	Number of times the gateway has tried to download the configuration files for SCCP-controlled endpoints.
Endpoint downloads succeeded	Number of times the gateway has successfully downloaded the configuration files for SCCP-controlled endpoints.
Endpoint resets	Number of SCCP gateway resets.
Endpoint restarts	Number of SCCP gateway restarts.
Managed endpoints	Number of SCCP-controlled endpoints (analog and BRI phones).

The following is sample output from the **show ccm-manager fallback-mgcp** command:

```
Router# show ccm-manager fallback-mgcp

Current active Call Manager: 172.20.71.38
MGCP Fallback mode: Enabled/OFF
Last MGCP Fallback start time: 00:14:35
Last MGCP Fallback end time: 00:17:25
```

Table 12 describes significant fields shown in this output.

**Table 12** *show ccm-manager fallback-mgcp Field Descriptions*

Field	Description
MGCP Fallback mode	The following are displayed: <ul style="list-style-type: none"> <li>• Not Selected—Fallback is not configured.</li> <li>• Enabled/OFF—Fallback is configured but not in effect.</li> <li>• Enabled/ON—Fallback is configured and in effect.</li> </ul>
Last MGCP Fallback start time	Start time stamp of the last fallback.
Last MGCP Fallback end time	End time stamp of the last fallback.

The following is sample output from the **show ccm-manager music-on-hold** command:

```
Router# show ccm-manager music-on-hold

Current active multicast sessions :1
Multicast      RTP port  Packets      Call   Codec   Incoming
Address        number    in/out       id     Codec   Interface
=====
172.20.71.38   2428     5/5         99     g711
```

Table 13 describes significant fields shown in this output.

**Table 13** *show ccm-manager music-on-hold Field Descriptions*

Field	Description
Current active multicast sessions	Number of active calls on hold.
Multicast Address	Valid class D address from which the gateway is getting the RTP streams.
RTP port number	Valid RTP port number on which the gateway receives the RTP packets.
Packets in/out	Number of RTP packets received and sent to the digital signal processor (DSP).
Call id	Call ID of the call that is on hold.
Codec	Codec number.
Incoming Interface	Interface through which the gateway is receiving the RTP stream.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>ccm-manager config</b>	Supplies the local MGCP voice gateway with the IP address or logical name of the TFTP server from which to download XML configuration files and enable the download of the configuration.
<b>debug ccm-manager</b>	Displays debugging information about the Cisco CallManager.
<b>show ccm-manager</b>	Displays a list of Cisco CallManager servers, their current status, and their availability.
<b>show ccm-manager fallback-mgcp</b>	Displays the status of the MGCP gateway fallback feature.
<b>show isdn status</b>	Displays the Cisco IOS gateway ISDN interface status, shows that L3 is configured as Cisco CallManager, and shows that L2 is established.
<b>show mgcp</b>	Displays the MGCP configuration information.

# show stcapp device

To display configuration information about SCCP Telephony Control Application (STCAPP) analog voice ports, use the **show stcapp device** command in privileged EXEC mode.

**show stcapp device** {**name** *device-name* | **summary** | **voice-port** *port*}

## Syntax Description

<b>name</b> <i>device-name</i>	Displays information for the analog voice port with the specified device name. The device name is the unique device ID that is assigned to the port when it registers with the call-control system.
<b>summary</b>	Displays a summary of all voice ports.
<b>voice-port</b> <i>port</i>	Displays information for the specified analog voice port.
<b>Note</b>	The <i>port</i> value is platform-dependent; type ? to determine.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.3(14)T	This command was introduced.
12.4(2)T	Command output was enhanced to display call control block (CCB) and call-control device information.
12.4(4)T	Command output was enhanced to display supported modem transport capability.
12.4(6)XE	Command output was enhanced to display supported visual message waiting indicator (VMWI) and dial tone after remote onhook information.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

## Usage Guidelines

Use this command to display configuration and voice interface card (VIC)-specific port information. The Active Call Info field is populated only if a call is active on the voice port.

## Examples

The following is sample output from the **show stcapp device voice-port** command for an SCCP analog port with VMWI and the dial tone after remote onhook feature is activated in Cisco IOS 12.4(6)XE:

```
Router#show stcapp device voice-port 2/4

Port Identifier: 2/4
Device Type:    ALG
Device Id:      4
Device Name:    AN0C863967C9404
Modem Capability: None
Device State:   IS
Diagnostic:     None
Directory Number: 7204
Dial Peer(s):  4
Dialtone after remote onhook feature: activated
Last Event:     STCAPP_CC_EV_CALL_DISCONNECT_DONE
```

```

Line State:      IDLE
Hook State:     ONHOOK
mwi:           ENABLE
vmwi:          ON
PLAR:          DISABLE
Number of CCBs: 0

```

The following is sample output from the **show stcapp device voice-port** command in Cisco IOS 12.3(14)T for an STCAPP analog voice port on a VIC2-2FXS voice interface card :

```

Router# show stcapp device voice-port 1/0/0

Port Identifier: 1/0/0
Device Type:    ALG
Device Id:      4
Device Name:    AN0C85226910200
Device State:   IS
Call State:     ACTIVE
Diagnostic:     None
Directory Number: 7701
Dial Peer(s):  999100
Last Event:     STCAPP_CC_EV_CALL_DIGIT_END_TIMEOUT
Line State:    IDLE
Hook State:    ONHOOK
Number of CCBs: 0
Global call info:
  Total CCB count = 0
  Total call leg count = 0

Connected Call Info:
  Call Reference: 33554606
  Local IP Addr: 10.2.6.10
  Local IP Port: 17718
  Remote IP Addr: 10.2.6.10
  Remote IP Port: 18630
  Calling Number: 7701
  Called Number: 7702
  Codec:          g711ulaw

```

The following is sample output from the **show stcapp device name** command in Cisco IOS 12.3(14)T for an STCAPP analog voice port:

```

Router# show stcapp device name AN0C863972F5401

Port Identifier: 2/1
Device Type:    ALG
Device Id:      25
Device Name:    AN0C863972F5401
Device State:   IS
Diagnostic:     None
Directory Number: 9101
Dial Peer(s):  2
Last Event:     STCAPP_CC_EV_CALL_MODIFY_DONE
Line State:     ACTIVE
Hook State:     OFFHOOK
Number of CCBs: 1
Global call info:
  Total CCB count      = 3
  Total call leg count = 6

Call State for Connection 1: TsConnected
Connected Call Info:
  Call Reference: 16777509
  Local IP Addr: 10.1.0.1

```

```
show stcapp device
```

```
Local IP Port: 18768
Remote IP Addr: 10.1.0.1
Remote IP Port: 18542
Calling Number: 9101
Called Number: 9102
Codec: g711ulaw
```

The following is sample output from the **show stcapp device summary** command in Cisco IOS 12.3(14)T for STCAPP analog voice ports:

```
Router# show stcapp device summary
```

```
Total Devices:          24
Total Calls in Progress: 3
Total Call Legs in Use: 6
```

Port Identifier	Device Name	Device State	Call State	Dev Type	Directory Number	Dev Cntl
2/1	AN0C863972F5401	IS	ACTIVE	ALG	9101	CCM
2/2	AN0C863972F5402	IS	ACTIVE	ALG	9102	CCM
2/3	AN0C863972F5403	IS	ACTIVE	ALG	9103	CCM
2/0	AN0C863972F5400	IS	IDLE	ALG	9100	CCM
2/4	AN0C863972F5404	IS	IDLE	ALG	9104	CCM
2/5	AN0C863972F5405	IS	IDLE	ALG	9105	CCM
2/6	AN0C863972F5406	IS	IDLE	ALG	9106	CCM
2/7	AN0C863972F5407	IS	IDLE	ALG	9107	CCM
2/8	AN0C863972F5408	IS	IDLE	ALG	9108	CCM
2/9	AN0C863972F5409	IS	IDLE	ALG	9109	CCM
2/10	AN0C863972F540A	IS	IDLE	ALG	9110	CCM
2/11	AN0C863972F540B	IS	IDLE	ALG	9111	CCM
2/12	AN0C863972F540C	IS	IDLE	ALG	9112	CCM
2/13	AN0C863972F540D	IS	IDLE	ALG	9113	CCM
2/14	AN0C863972F540E	IS	IDLE	ALG	9114	CCM
2/15	AN0C863972F540F	IS	IDLE	ALG	9115	CCM
2/16	AN0C863972F5410	IS	IDLE	ALG	9116	CCM
2/17	AN0C863972F5411	IS	IDLE	ALG	9117	CCM
2/18	AN0C863972F5412	IS	IDLE	ALG	9118	CCM
2/19	AN0C863972F5413	IS	IDLE	ALG	9119	CCM
2/20	AN0C863972F5414	IS	IDLE	ALG	9120	CCM
2/21	AN0C863972F5415	IS	IDLE	ALG	9121	CCM
2/22	AN0C863972F5416	IS	IDLE	ALG	9122	CCM
2/23	AN0C863972F5417	IS	IDLE	ALG	9123	CCM

Table 14 describes the significant fields shown in these displays, in alphabetical order.

**Table 14** *show stcapp device Field Descriptions*

Field	Description
Active Call Info	Displays only when an active call is in progress.
Call Reference	Reference number created by Cisco CallManager to track messages associated with a specific call.
Call State	Call processing state: <ul style="list-style-type: none"> <li>ACTIVE—Established call connection</li> <li>IDLE—No call connection</li> <li>UNREGISTERED—Device is not registered with the Cisco CallManager</li> </ul>

**Table 14** *show stcapp device Field Descriptions*

Field	Description
Called Number	Device called number.
Calling Number	Device calling number.
Codec	Displays codec type.
Dev Cntl	Call-control device that is managing the analog endpoints. CCM represents Cisco CallManager. CME represents Cisco CallManager Express system.
Device Id	Identifier used between the Cisco CallManager and gateway to uniquely identify an endpoint.
Device Name	Unique device ID of the analog endpoint. The device ID is derived from an algorithm using the MAC address of the Skinny Client Control Protocol (SCCP) interface on the voice gateway and the hexadecimal translation of the port's slot number and port number.
Device State	Displays whether device is available for use: <ul style="list-style-type: none"> <li>• ACTIVE_PENDING—Call is pending certain events before going active</li> <li>• INFO_RCVD—Call information is received from the Cisco CallManager during call setup</li> <li>• INIT—Waiting to reinitialize</li> <li>• IS—In service</li> <li>• OFFHOOK—Device is off-hook</li> <li>• OFFHOOK_TIMEOUT—Digit timeout has occurred while the device is off-hook</li> <li>• ONHOOK_PENDING—Call is pending certain events before going to the on-hook state</li> <li>• OOS—Out of service</li> <li>• PROCEED—Dialed number translation is complete and call setup is in progress</li> <li>• REM_ONHOOK_PENDING—Call is pending certain events before going to the on-hook state</li> <li>• RINGING—An incoming call has invoked ringing of the receiving device</li> </ul>
Device Type	Shows phone type: <ul style="list-style-type: none"> <li>• ALG—Analog</li> <li>• BRI—ISDN BRI</li> </ul>
Diagnostic	Reason code for a device error condition.
Dial Peer(s)	Dial peer name.

**Table 14** *show stcapp device Field Descriptions*

<b>Field</b>	<b>Description</b>
Dialtone after remote onhook feature	Displays feature status: <ul style="list-style-type: none"> <li>• Activated</li> <li>• Not activated</li> </ul>
Directory Number	Assigned to the device by the Cisco CallManager.
Last Event	Last event processed by this port.
Local IP Addr	IP address of this gateway used to stream audio using the Real-Time Transport Protocol (RTP).
Local IP Port	IP port of this gateway used to stream audio using RTP.
Port Identifier	Identifies the physical voice port.
Remote IP Addr	IP address of the far-end gateway that streams audio using RTP.
Remote IP Port	IP port of the far-end gateway that streams audio using RTP.
VMWI	Displays LED status: <ul style="list-style-type: none"> <li>• On</li> <li>• Off</li> </ul>

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show stcapp statistics</b>	Displays call statistics for STCAPP devices.

# show stcapp statistics

To display call statistics for SCCP Telephony Control Application (STCAPP) voice ports, use the **show stcapp statistics** command in privileged EXEC mode.

**show stcapp statistics** [**all** | **voice-port** *port-number*]

## Syntax Description

**voice-port** *port-number* (Optional) Displays information for a specific voice port.

- *port-number*—Number of the port on the interface. Refer to the appropriate platform manual or online help for port numbers on your networking device.

**all** (Optional) Displays a summary of all voice ports.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.3(14)T	This command was introduced.

## Usage Guidelines

Use this command to display call statistics for STCAPP voice ports.

## Examples

The following is sample output for the **show stcapp statistics** command for STCAPP voice port 1/0/0.1:

```
Router# show stcapp statistics voice-port 1/0/0.1

STCAPP Device/Call Statistics
  OA = Origination Attempts, TA = Termination Attempts
  Err = Call Errors, PE = Call PreEmptions
Port      DevErr  CallOA  CallTA  CallErr  CallPE
-----
1/0/0.1  0        7        0        0        0
```

The following is sample output for the **show stcapp statistics** command for all STCAPP voice ports:

```
Router# show stcapp statistics all

STCAPP Device/Call Statistics
  OA = Origination Attempts, TA = Termination Attempts
  Err = Call Errors, PE = Call PreEmptions
Port      DevErr  CallOA  CallTA  CallErr  CallPE
-----
1/0/0      0        7        0        0        0
1/0/1      0        0        7        0        0
1/0/3      0        0        0        0        0
1/1/0.1    0        0        0        0        0
1/1/1.1    0        0        0        0        0
1/0/2      0        0        0        0        0
```

[Table 15](#) describes the significant fields shown in the display.

**Table 15** *show stcapp statistics Field Descriptions*

<b>Field</b>	<b>Description</b>
DevErr	Device errors.
CallOA	Call origination attempts.
CallTA	Call termination attempts.
CallErr	Call errors.
CallPE	Call preemptions.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show stcapp device</b>	Displays configuration information about STCAPP voice port.s

# stcapp

To enable the SCCP Telephony Control Application (STCAPP), use the **stcapp** command in global configuration mode. To disable the STCAPP, use the **no** form of this command.

**stcapp**

**no stcapp**

## Syntax Description

This command has no arguments or keywords.

## Defaults

The Cisco CallManager does not control Cisco IOS gateway-connected analog and BRI endpoints.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(14)T	This command was introduced.

## Usage Guidelines

Use the **stcapp** command to enable basic Skinny Client Call Control (SCCP) call control features for Basic Rate Interface (BRI) and Foreign Exchange Stations (FXS) analog ports within Cisco IOS voice gateways. The **stcapp** command enables the Cisco IOS Gateway application to support the following features:

- Line-side support for the Multilevel Precedence and Preemption (MLPP) feature
- Cisco CallManager registration of analog and Basic Rate Interface (BRI) endpoints
- Cisco CallManager endpoint auto configuration support
- Modem pass-through support
- Cisco Survivable Remote Site Telephony (SRST) support

## Examples

The following example shows that STCAPP is enabled:

```
Router(config)# stcapp
```

## Related Commands

Command	Description
<b>stcapp ccm-group</b>	Configures the Cisco CallManager group number for use by the SCCP Telephony Control Application (STCAPP).
<b>stcapp timer</b>	Enables SCCP Telephony Control Application (STCAPP) timer configuration.
<b>ccm- manager config server</b>	Specifies the TFTP server for Skinny Client Control Protocol (SCCP) gateway downloads.

<b>Command</b>	<b>Description</b>
<b>ccm-manager sccp local</b>	Specifies the Skinny Client Control Protocol (SCCP) local interface.
<b>sccp</b>	Enables the Skinny Client Control Protocol (SCCP) protocol.
<b>show stcapp device</b>	Displays configuration information about SCCP Telephony Control Application (STCAPP) voice ports.
<b>show stcapp statistics</b>	Displays call statistics for SCCP Telephony Control Application (STCAPP) voice ports.

## stcapp ccm-group

To configure the Cisco CallManager group number for use by the SCCP Telephony Control Application (STCAPP), use the **stcapp ccm-group** command in global configuration mode. To disable STCAPP Cisco CallManager group number configuration, use the **no** form of this command.

**stcapp ccm-group** *group-id*

**no stcapp ccm-group** *group-id*

### Syntax Description

<i>group-id</i>	Cisco CallManager group number.
-----------------	---------------------------------

### Defaults

No Cisco CallManager group number is configured.

### Command Modes

Global configuration.,

### Command History

Release	Modification
12.3(14)T	This command was introduced.

### Usage Guidelines

The Cisco CallManager group identifier must have been configured for the Service Provider Interface (SPI) using the command **sccp ccm-group** *group-id*.

### Examples

The following example configures the STCAPP to use Cisco CallManager group 2:

```
Router(config)# stcapp ccm-group 2
```

### Related Commands

Command	Description
<b>stcapp</b>	Enables the SCCP Telephony Control Application (STCAPP).
<b>stcapp timer</b>	Enables SCCP Telephony Control Application (STCAPP) timer configuration.
<b>show stcapp device</b>	Displays configuration information about SCCP Telephony Control Application (STCAPP) voice ports.
<b>show stcapp statistics</b>	Displays call statistics for SCCP Telephony Control Application (STCAPP) voice ports.

# stcapp timer

To enable SCCP Telephony Control Application (STCAPP) timer configuration, use the **stcapp timer** command in global configuration mode. To disable STCAPP timer configuration, use the **no** form of this command.

**stcapp timer roh** *seconds*

**no stcapp timer**

<b>Syntax Description</b>	<i>seconds</i>	Duration, in seconds, that the receiver off-key tone is played. Range is 0 to 120 seconds.
	<b>roh</b>	Receiver off hook (ROH) tone time-out.

**Defaults** 45 seconds

**Command Modes** Global configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
		12.3(14)T

**Usage Guidelines** Use this command to configure the STCAPP receiver off hook (ROH) timer for the maximum time that ROH tone is played. ROH tone signals a subscriber that the phone remains offhook when there is no active call.

**Examples** The following example configures the receiver off hook timer for 30 seconds:

```
Router(config)# stcapp timer roh 30
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>stcapp</b>	Enables the SCCP Telephony Control Application (STCAPP)
	<b>show call application voice stcapp</b>	Displays information about the SCCP Telephony Control Application (STCAPP)

# Glossary

**API**—Application program interface. The means by which an application program talks to communications software.

**B channel**—Bearer channel. DS0 time slot that carries analog voice or digital data over ISDN. In ISDN, a full-duplex, 64-kbps channel used to send user data.

**BRI**—Basic Rate Interface. ISDN interface composed of two B channels and one D channel for circuit-switched communication of voice, video, and data.

**CCAPI**—Call control API.

**CPE**—Customer premises equipment. Terminating equipment, such as terminals, telephones, and modems, supplied by the telephone company, installed at customer sites, and connected to the telephone company network. Can also refer to any telephone equipment residing on the customer site.

**DCAPI**—Device control API.

**FXS**—Foreign Exchange Station. An FXS interface connects directly to a standard telephone and supplies ring, voltage, and dial tone. Cisco's FXS interface is an RJ-11 connector that allows connections to basic telephone service equipment, keysets, and PBXs.

**H.323**—H.323 protocol 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.

**ISDN**—Integrated Services Digital Network. Communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

**MAC address**—Standardized data link layer address that is required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, MAC layer address, and physical address.

**MGCP**—Media Gateway Control Protocol. A merging of the IPDC and SGCP protocols.

**MLPP**—Multilevel Priority and Preemption.

**NSE**—Named Signaling Events

**RTP**—Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

**SCCP**—Skinny Client Control Protocol.

**SRST**—Survivable Remote Site Telephony.

**STCAPP**—SCCP Telephony Control Application.

**TFTP**—Trivial File Transfer Protocol. Simplified version of FTP that allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password).

**VoIP**—Voice over IP. The capability to carry normal telephony-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

**XML**— extensible markup language. A standard maintained by the World Wide Web Consortium (W3C). It defines a syntax that lets you create markup languages to specify information structures. Information structures define the type of information, for example, subscriber name or address, not how the information looks (bold, italic, and so on). External processes can manipulate these information structures and publish them in a variety of formats. Text markup language designed to enable the use of SGML on the World Wide Web. XML allows you to define your own customized markup language.

**Note**

---

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

---

---

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, Phone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website 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. (0711R)

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

Copyright © 2006 Cisco Systems, Inc. All rights reserved.