Configuring Digital E1 Packet Voice Trunk Network Module Interfaces

This document describes how to configure digital E1 packet voice trunk network module interfaces on Cisco 2600 and 3600 series routers and includes the following sections:

Feature Overview

Digital E1 packet voice trunk network modules for Cisco 2600 and 3600 series routers allow enterprises or service providers, using the equipped routers as customer premises equipment, to deploy digital voice and fax relay. These modules receive constant bit-rate telephony information over E1 interfaces and can convert that information to a compressed format, so that it can be transmitted as Voice over IP (VoIP), Voice over Frame Relay (VoFR), and Voice over ATM (VoATM).

Cisco IOS software configuration allows you to set up a variety of applications. Here are a few examples:

- Compressed voice over WANs
- Routing of dialed variable-length digits collected from the public switched telephone network or PBX for VoIP, VoFR, and VoATM.
- Support for FRF.12 fragmentation and queuing in a VoIP over Frame-Relay network
- Drop and Insert of E1 channels on a E1 trunk to allow some PBX channels to be directed to the PSTN while others are used for compressed VoIP

For more information about these applications, see “Configuration Example” on page 40.

- Dynamic bandwidth allocation using voice activity detection (VAD)

- Drop-and-Insert capability, allowing the interchange of time-division multiplexing (TDM) slots between the ports on a two-port E1 multiflex trunk voice/WAN interface card installed in a digital E1 packet voice trunk network module
- Support for a wide range of International Telecommunication Union (ITU-T) G-series compression specifications, including:
  - G.711 A Law at 64,000 bps
  - G.711 u Law at 64,000 bps
  - G.723.1 Annex A at 5,300 bps
  - G.723.1 Annex A at 6,300 bps
  - G.723.1 at 5,300 bps
  - G.723.1 at 6,300 bps
  - G.726 at 16,000 bps
  - G.726 at 24,000 bps
  - G.726 at 32,000 bps
  - G.728 at 16,000 bps
  - G.729 at 8,000 bps
  - G.729 Annex A at 8,000 bps
  - G.729 Annex B at 8,000 bps
  - G.729 Annex B with Annex A at 8,000 bps
- Depending on codec complexity, either 30 or 60 channels of compressed voice
- High-quality voice endpoint-standard features, such as high-quality echo cancellation, silence suppression, comfort noise generation, and DTMF relay
- Group 3 fax relay

Benefits

Digital E1 packet voice trunk network modules allow Cisco 2600 and 3600 series routers to provide E1 connectivity to private branch exchanges (PBXs) or to a central office (CO). With digital E1 connectivity, Cisco 2600 and 3600 series routers can provide greater voice density for enterprise and service provider VoIP networks than they could before. A digital E1 packet voice trunk network module is a complete solution, made up of a network module with installed packet voice data modules (PVDMs), and one E1 multiflex trunk voice/WAN interface card with either one or two E1 ports.

E1 Timing, Signaling, Framing, and Line Encoding

With the introduction of the digital E1 packet voice trunk network modules for the Cisco 2600 and 3600 series routers, you must set timing, signaling, framing, and line encoding. Digital E1 packet voice trunk network modules can connect to either a PBX (or similar telephony device) or to a Central Office (CO) in order provide PSTN connectivity.

The differences that set E1 digital configuration apart from analog configuration are as follows:

- **Timing.** Analog interfaces do not require specific timing configuration. Digital E1 interfaces require not only that you set timing but that you consider the source of the timers.
- **Framing.** Analog interfaces do not require specific framing configuration. Digital E1 interfaces require that you configure for cyclic redundancy checking 4 (CRC-4) framing. Set the framing format to match that of the PBX or CO that connects to the digital E1 packet voice trunk network module.
- **Line Encoding.** Analog interfaces do not require specific line encoding configuration. Digital E1 interfaces require that you configure for High Density bipolar 3 (HDB3) encoding (similar to alternative mark inversion, or AMI). Set the line encoding to match that of the PBX or CO that connects to the digital E1 packet voice trunk network module.

Timing

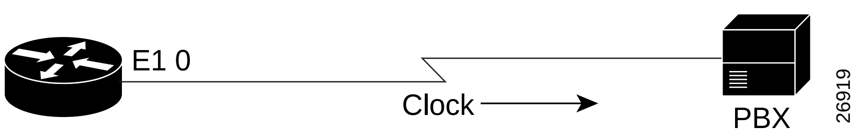
This section describes the five basic timing scenarios that can occur when a digital E1 packet voice trunk network module is connected to a PBX, CO, or both. In all of the examples below, the PSTN (or Central Office) and the PBX are interchangeable for the purposes of providing or receiving clocking.

The digital E1 module has an on-board PLL (Phase-Lock Loop) chip that can either provide a clock source to both E1s or receive clocking that can drive the second E1 in the same digital E1 packet voice trunk network module. All timing commands are E1 controller configuration commands.

Single E1 Port Provides Clocking

In this scenario, the digital E1 module is the clock source for the connected device. The PLL generates the clock internally and drives the clocking on the E1 line.

Figure 1 Single E1 Port Providing Clock



The following configuration sets up this clocking method:

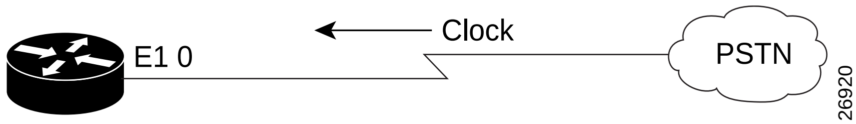
```
controller E1 1/0
framing crc4
linecoding hdb3
clock source internal
pri-group timeslots 1-31
```

Note Generally this method is useful only when connecting to a PBX, key system or channel bank. A Cisco VoIP Gateway rarely provides clocking to the CO, because CO clocking provides a higher Stratum level.

Single E1 Port Receiving Clock from the Line

In this scenario, the digital E1 module receives clocking from the connected device (CO or PBX). The PLL clocking is driven by the clock reference on the receive (*Rx*) side of the E1 connection.

Figure 2 Single E1 Receiving Clock from Line



The following configuration sets up this clocking method:

```
controller E1 1/0
framing crc4
linecoding b8zs
clock source line
pri-group timeslots 1-31
```

Dual E1s, Both Receive Clocking from the Line

In this scenario, the digital E1 has two reference clocks, one from the PBX and another from the CO. Since the PLL can only derive clocking from one source, this case is more complex than the two preceding examples.

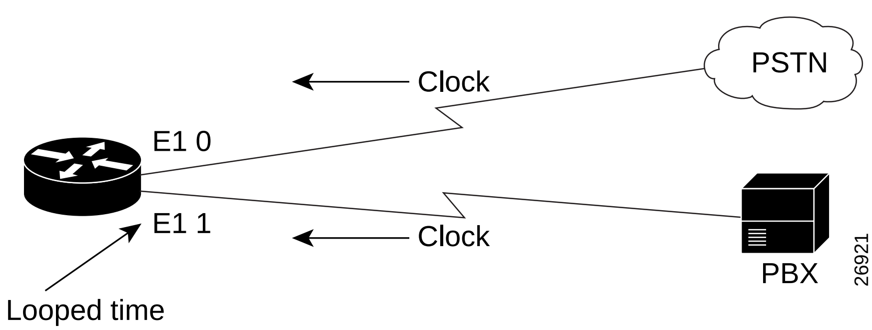
Before looking at the details, consider two important concepts that underlay the clocking method:

- **Looped-Time Clocking.** The E1 port takes the clock received on its *R*x (receive) pair and regenerates it on its *Tx* (transmit) pair. While the port receives clocking, the port is not driving the PLL on the card but is "spoofing" the E1 so that the connected device has a viable clock and does not see slips. PBXs are not designed to accept slips on a E1 line and such slips cause a PBX to drop the link into failure mode. While in looped-time mode, the router often sees slips, but because these are controlled slips, they usually do not force failures of the router's E1 port.
- **Slips.** These messages indicate that the E1 port is receiving clock information that is out of phase, that is, out of synch. Because the router has only a single PLL, it can experience controlled slips while it receives clocking from two different time sources.

  The router can usually handle controlled slips because its single PLL architecture anticipates them.

Note Physical layer issues, such as bad cabling or faulty clocking references, can also cause slips. Eliminate these slips by addressing the physical layer or clock reference problems.

Figure 3 Dual E1s Receiving Line Clocking



In this scenario, the PLL derives clocking from the CO and puts the E1 port connected to the PBX into looped-time mode. This is usually the best method because the CO provides an excellent clock source (and usually requires that it provide that source) and a PBX usually must receive clocking from the other E1.

The following configuration sets up this clocking method:

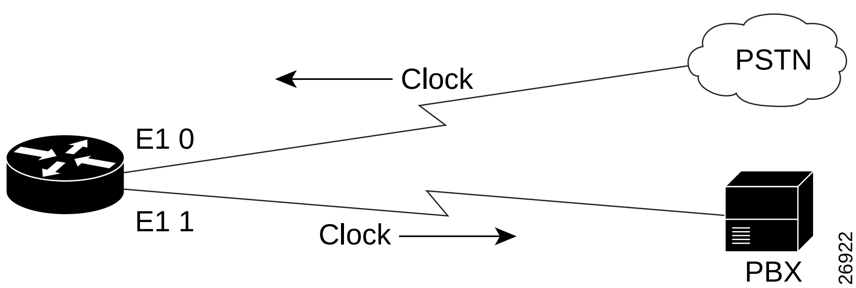
```
controller E1 1/0 << description - connected to the CO
framing crc4
linecoding hdb3
clock source line primary
pri-group timeslots 1-31
!
controller E1 1/1 << description - connected to the PBX
framing crc4
linecoding hdb3
clock source line
pri-group timeslots 1-31
```

The **clock source line primary** command tells the router to use this E1 port to drive the PLL. All other E1 ports configured as **clock source line** are then put into an implicit loop-timed mode. If the primary E1 port fails or goes down, the other E1 instead receives the clock that drives the PLL. In this configuration, E1 1/1 may see controlled slips, but these should not force it down. This method prevents the PBX from seeing slips.

Dual E1s, One Receives Clocking and One Provides Clocking

In this scenario, the digital E1 module receives clocking for the PLL from E1 0 and uses this clock as a reference to clock E1 1. If E1 0 fails, the PLL internally generates the clock reference to drive E1 1.

Figure 4 Dual E1s, One Receiving and One Providing Clocking



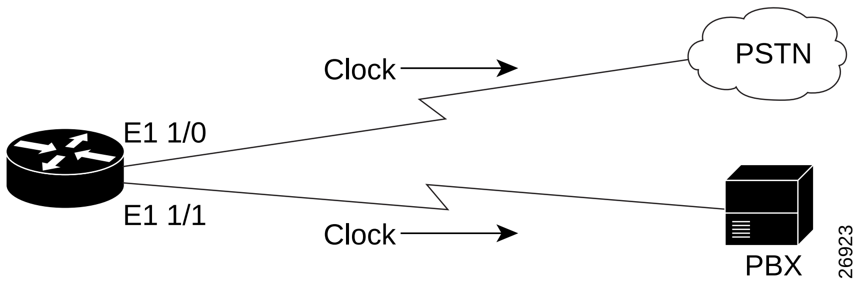
The following configuration sets up this clocking method:

```
controller E1 1/0
framing crc4
linecoding hdb3
clock source line
pri-group timeslots 1-31
!
controller E1 1/1
framing crc4
linecoding hdb3
clock source internal
pri-group timeslots 1-31
```

Dual E1s, Both Clocks from Router

In this scenario, the router is "Master of the Timing Universe," generating the clock for the PLL and therefore for both E1s.

Figure 5 Dual E1s, Both Clocks from Router



The following configuration sets up this clocking method:

```
controller E1 1/0
framing crc4
linecoding hdb3
clock source internal
pri-group timeslots 1-31
!
controller E1 1/1
framing esf
linecoding b8zs
clock source internal
pri-group timeslots 1-31
```

Verifying Configuration

Use the **show controller** privileged EXEC command to verify the proper digital E1 configuration:

```
router# show controller E1 1/0
E1 1/0 is up.
  Applique type is Channelized E1
  Cablelength is short 133
  Description: Digital E1 WIC
  No alarms detected.
  Framing is CRC4, Line Code is HDB3, Clock Source is Line Primary.
  Data in current interval (2 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Restrictions

The following restrictions apply to digital E1 packet voice trunk network module configuration:

- Group 4 fax is not supported.
- The high-density voice network module has one slot for a voice/WAN interface card (VWIC); VWICs supply one or two ports. Only the dual-mode (voice/WAN) multiflex trunk cards are supported in the digital E1 packet voice trunk network module, not older VICs. For more information, see the "Prerequisites" section on page 10.
- Drop-and-Insert capability is supported only between two ports on the same multiflex card.

- Common-channel signaling (CCS) and Primary Rate Interface (PRI) are not supported.
- R2 signaling is not supported.
- Voice over ATM—including AAL5 encapsulation, circuit emulation service (CES), and AAL2—is not supported for VoATM on the Cisco 2600 series router.
- Digital E1 voice is manageable through Simple Network Management Protocol (SNMP) using release 2.0 of Cisco Voice Manager.

Related Documents

The following online documents can help you understand how to install Cisco 2600 and 3600 series routers:

- Cisco 2600 Series Hardware Installation Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/2600hig/index.htm
- Quick Start Guide Cisco 2600 Series Cabling and Setup:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/2600ja/index.htm
- Software Configuration Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/software/index.htm
- Cisco 3660 Router Cabling and Setup Quick Start Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/3660quik.htm
- Cisco 3600 Series Hardware Installation Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/3600ig/index.htm
- Cisco Network Modules Hardware Installation Guide For Cisco 3600 Series and Cisco 2600 Series Routers:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/net_mod2
  /index.htm

The following Cisco IOS Release 12.0 documents are also helpful:

- Dial Solutions Configuration Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/dial_c/index.htm
- Dial Solutions Command Reference:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/dial_r/index.htm
- Voice, Video, and Home Applications Configuration Guide:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/voice_c/index.htm
- Voice, Video, and Home Applications Command Reference:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/voice_r/index.htm

The following documents can help you troubleshoot ISDN, PRI, and BRI connections:

- Internetwork Troubleshooting Guide:
  http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/tr1917.htm
- Debug Command Reference
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12supdoc
  /debug_r/index.htm

For more information about supported hardware on a Cisco 2600 or 3600 series router, go to:

- http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/index.htm
- http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/index.htm

- For the Voice over IP Quick Start Guides, go to: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/voice/4936vqsg.htm

Related Features and Technologies

VoIP Quality of Service

This section explains the quality issues that you should consider when building Voice over IP (VoIP) networks and offers a few tips about configuring VoIP with the appropriate Quality of Service (QoS):

- **Delay.** Delay is the time it takes for VoIP packets to travel between two endpoints and you should design networks to minimize this delay. However, because of the speed of network links and the processing power of intermediate devices, some delay is expected. The human ear normally accepts up to about 150 milliseconds (ms) of delay without noticing problems (the ITU's G.114 standard recommends no more than 150 ms of one-way delay). Once delay exceeds 150 ms, a conversation becomes more and more like a walkie-talkie interchange, where one person must wait for the other to stop speaking before beginning to talk. This type of delay is often evident on international long-distance calls. You can measure delay fairly easily by using ping tests at various times of the day with different network traffic loads. If network delay is excessive, reduce it before deploying VoIP networks.
- **Jitter.** While delay can cause unnatural starting and stopping of conversations, variable-length delays (also known as *jitter*) can cause a conversation to break and become unintelligible. Jitter is not usually a problem with public switched telephone network (PSTN) calls, because the bandwidth of calls is fixed. However, in VoIP networks where existing data traffic might be bursty, jitter can become an issue. Cisco voice gateways have built-in de-jitter buffering to compensate for a certain amount of jitter, but if jitter is constant on a network, identify the source and control it before deploying a VoIP network.
- **Serialization.** Serialization is a term that describes what happens when a router attempts to send both voice and data packets out of an interface. In general, voice packets are very small (80 to 256 bytes), while data packets can be very large (1,500 to 18,000 bytes). On relatively slow links, such as WAN connections, large data packets can take a long time to transmit onto the wire. When these large packets are mixed with smaller voice packets, the excessive transmission time can lead to both delay and jitter. You can use fragmentation to reduce the size of the data packets so that the voice delay and jitter also decrease.
- **Bandwidth Consumption.** Traditional voice conversations consume 64 Kb of network bandwidth. When this voice traffic is run though a VoIP network, it can be compressed and digitized by digital signal processors (DSPs) built into the routers. This compression can reduce the calls to sizes as small as 5.3 Kb for voice samples. Once the packets go onto the IP network, the appropriate IP/UDP/RTP headers must be added, and this can add a significant amount of bandwidth to each call (about 40 bytes per packet). Technologies such as Compressed Real-Time Protocol (CRTP), however, can reduce the IP header overhead to about 4 bytes. In addition, VAD (voice activity detection) does not send any packets unless there is active speech.

Supported Platforms

This feature is supported on the following platforms:

- Cisco 2610
- Cisco 2611
- Cisco 2612
- Cisco 2613
- Cisco 2620
- Cisco 2621
- Cisco 3620
- Cisco 3640
- Cisco 3662
- Cisco 3661

Supported Standards, MIBs, and RFCs

RFCs

- RFC 1890
- RFC 1889

MIBs

- CISCO-ENTITY-VENDORTYPE-OID-MIB
- OLD-CISCO-CHASSIS-MIB
- CAS_INTF_MIB

International Telecommunication Union (ITU-T) G-Series Codec Compression Specifications

- G.711 A Law at 64,000 bps
- G.711 u Law at 64,000 bps
- G.723.1 Annex A at 5,300 bps
- G.723.1 Annex A at 6,300 bps
- G.723.1 at 5,300 bps
- G.723.1 at 6,300 bps
- G.726 at 16,000 bps
- G.726 at 24,000 bps
- G.726 at 32,000 bps
- G.728 at 16,000 bps
- G.729 at 8,000 bps
- G.729 Annex A at 8,000 bps

- G.729 Annex B at 8,000 bps
- G.729 Annex B with Annex A at 8,000 bps

Prerequisites

Digital E1 packet voice requires specific service, software, and hardware:

- Obtain E1 service from your service provider or PBX.
- Install Cisco IOS Software Release 12.0(7)XK or a later release. The *minimum* DRAM memory requirements to support digital E1 packet voice trunk network modules are as follows:
  - 48 Mb with one or two E1s
  - 64 Mb with three to eight E1s
  - 128 Mb with 9 to 12 E1s

  The memory required may be greater than listed above for high-volume applications.

  Support for digital E1 packet voice trunk network modules is included in Plus feature sets. The IP Plus feature set requires 16 Mb of flash memory.
- Install one of the following high-density E1 network modules in the router chassis:
  - Single-Port 30 Channel E1 High-Density Voice Network Module (NM-HDV-1E1-30)
  - Single-Port Enhanced 30 Channel E1 High-Density Voice Network Module (NM-HDV-1E130E)
  - Dual-Port 60 Channel High-Density Voice Network Module (NM-HDV-2E1-60)

  Note You can install one module in a Cisco 2600 series router or a Cisco 3620 router. A Cisco 3640 router can support three modules, and you can install as many as six modules in a Cisco 3660 router.

- Install at least one packet voice data module (PVDM-12) in the high-density digital E1 network module if it is not already equipped. The digital E1 packet voice trunk network module contains five 72-pin SIMM sockets or banks, numbered 0 through 4, for PVDMs. Each socket can be filled with a single 72-pin PVDM. A digital E1 packet voice trunk network module can support the following numbers of channels:
  - When the digital E1 packet voice trunk network module is configured for high-complexity codec mode, up to six voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729, , G729 Annex A, G.729 Annex B, G.723.1, G.728, and fax relay.
  - When the digital E1 packet voice trunk network module is configured for medium-complexity codec mode, up to twelve voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729 Annex A, G.729 Annex B with Annex A, and fax relay.

  Note Each PVDM holds three digital signal processors (DSPs). With five PVDM slots populated, a total of 15 DSPs are provided. High-complexity codecs support two simultaneous calls on each DSP, while medium-complexity codecs support four calls on each DSP.

- Install at least one dual-mode voice/WAN interface card (VWIC) for a voice connection if a VWIC was not included with the network module. You can install one VWIC (providing one or two line interfaces) in the digital E1 packet voice trunk network module. Only the one- and two-port E1 multiflex trunk interface cards (VWIC-1MFT-E1, VWIC-2MFT-E1, VWIC-2MFT-E1-DI) are supported.

  For Drop-and-Insert capability, you must install a two-port Drop-and-Insert E1 multiflex trunk voice/WAN interface card (VWIC-2MFT-E1-DI). To install a VWIC in a network module, see *Cisco WAN Interface Cards Hardware Installation Guide*.
- Install at least one other network module or WAN interface card to provide the connection to the IP LAN or WAN.
- Establish a working IP, frame relay, or ATM network. For more information about configuring IP, see “IP Overview,” “Configuring IP Addressing,” and “Configuring IP Services” chapters in the Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1*.
- Complete your company’s dial plan.
- Establish a working telephony network based on your company's dial plan.

Voice, Video, and Home Applications Configuration Guide and *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0 provide information about setting up voice networks.

Configuration Tasks

Perform the following tasks to configure a digital E1 packet voice trunk network module:

- Set up voice cards and E1 controllers.
- Configure serial and LAN interfaces.
- Set up voice ports.
- Configure voice dial peers.

Configuring Voice Card and E1 Controller Settings

The following steps specify codec settings for voice cards and set up E1 controllers for clocking and other E1 parameters, as well as for DS0 groups that define the channels for compressed voice and TDM groups for Drop-and-Insert capability.

Step	Command	Purpose
1	`Router#` **`configure terminal`**	Enter global configuration mode.
2	`Router(config)#` **`voice-card`** *`slot`*	Enter voice card interface configuration mode and specify the slot location by using a value from 0 to 5, depending upon your router.
3	`Router(config-voice-ca)`**`# codec complexity`** {**`high`** | **`medium`**}	Specify the codec complexity based on the codec standard you are using. High-complexity codecs support lower call density than do medium-complexity codecs. The number of channels supported is based on the number of PVDMs installed and the codec complexity. Here is a guideline: • When the digital E1 packet voice trunk network module is configured for high-complexity codec mode, up to six voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729, G.729 Annex B, G.723.1, G.723.1 Annex A, G.728, and fax relay. • When the digital E1 packet voice trunk network module is configured for medium-complexity codec mode, up to twelve voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729 Annex A, G.729 Annex B with Annex A, and fax relay All voice cards in a router must use the same codec complexity setting. The keyword that you specify for **codec complexity** affects the choice of codecs available using the **codec** dial-peer configuration command. See Step 7 in "Configuring Voice Dial Peers" on page 19. **Note** You cannot change codec complexity while DS0 groups are defined. If they are already set up, use the **no ds0-group** command before resetting the codec complexity. For more information about the **pri-group** command, see Step 9.
4	`Router(config)#` **`controller E1`** *`slot/port`*	Enter controller configuration mode for the E1 controller at the specified *slot/port* location. Valid values for *slot* and *port* are 0 and 1.

Step	Command	Purpose
5	`Router(config-controller)#` **`clock source`** `{`**`line`** `[`**`primary`**`]` `|` **`internal`**`}`	Configure controller E1 1/0 to specify the clock source. The **line** keyword specifies that the clock source is derived from the active line—rather than from the free-running internal clock. This is the default setting and is generally more reliable. These rules apply to clock sourcing on the E1 controller ports: • When both ports are set to line clocking with no primary specification, port 0 is the default primary clock source and port 1 is the default secondary clock source. • When both ports are set to line and one port is set as the primary clock source, the other port is by default the backup or secondary source and is loop-timed. • If one port is set to **clock source line** or **clock source line primary** and the other is set to **clock source internal**, the internal port recovers clock from the clock source line port if the clock source line port is up. If it is down, then the internal port generates its own clock. • If both ports are set to **clock source internal**, there is only one clock source—internal. See E1 Timing, Signaling, Framing, and Line Encoding, page 2 for more information about configurations for clocking.
6	`Router(config-controller)#` **`framing crc4`**	Set the framing according to your service provider's instructions. Choose cyclic reduncancy check 4 (CRC4) format.
7	`Router(config-controller)#` **`linecode hdb3`**	Set the line encoding according to your service provider's instructions. E1 uses High Density bipolar 3 (HDB3) encoding (similar to alternative mark inversion, or AMI).

Step	Command	Purpose
8	`Router(config-controller)#` **`cablelength long`** {**`gain26`** | **`gain36`**} {**`-15db`** | **`-22.5db`** | **`-7.5db`** | **`0db`**} or **`cablelength short`** {**`133`** | **`266`** | **`399`** | **`533`** | **`655`**}	(E1 interfaces only) The cable length setting must conform to the actual cable length you are using. For example, if you attempt to enter the **cablelength short** command on a long-haul E1 link, the command is rejected. To set a cable length longer than 655 feet for a E1 link, use the **cablelength long** command. The keywords are as follows: • **gain26** specifies the decibel pulse gain at 26. This is the default pulse gain. • **gain36** specifies the decibel pulse gain at 36. • **-15db** specifies the decibel pulse rate at -15 decibels. • **-22.5db** specifies the decibel pulse rate at -22.5 decibels. • **-7.5db** specifies the decibel pulse rate at -7.5 decibels. • **0db** specifies the decibel pulse rate at 0 decibels. This is the default pulse rate. To set a cable length 655 feet or less for a E1 link, use the **cablelength short** command. There is no default for **cablelength short**. The keywords are as follows: • **133** specifies a cable length from 0-133 feet. • **266** specifies a cable length from 134-266 feet. • **399** specifies a cable length from 267-399 feet. • **533** specifies a cable length from 400-533 feet. • **655** specifies a cable length from 534-655 feet. If you do not set the cable length, the system defaults to a setting of **cablelength long gain26 0db**.
9	`Router(config-controller)#` **`pri-group timeslots`** *`timeslot-list`*	Enter a single timeslot number, a single range of values. For E1, the allowable values are from 1 to 31.
10	`Router(config-controller)#` **`no shutdown`**	Activate the controller.
11	`Router(config-controller)#` **`exit`**	Exit controller configuration mode. Skip the next step if you are not setting up Drop and Insert.

Repeat Steps 2 and 3 for each voice card.

Repeat Steps 4 through 11 for each controller.

Verifying Voice Card and Controller Settings

To verify the configuration of voice card and controller settings, follow these steps:

Step 1 Enter the **show running-config** command to display the current voice-card setting. If no codec complexity is shown, the default of medium complexity is set. The following example shows an excerpt from the command output:

```
Router# show running-config
.
.
.
hostname router-alpha

voice-card 1
 codec complexity high
.
.
.
```

Step 2 The privileged EXEC **show controllers E1** command displays the status of E1 controllers and displays information about clock sources and other settings for the E1 ports:

```
Router# show controller E1 1/0

E1 1/0 is up.
  Applique type is Channelized E1
  Cablelength is short 133
  Description: E1 WIC card Alpha
  No alarms detected.
  Framing is CRC4, Line Code is HDB3, Clock Source is Line Primary.
  Data in current interval (1 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Configuring Serial Interfaces

The way you set up serial and LAN interfaces depends on your application. To configure VoIP, you must at least set up IP addresses for serial interfaces. When a user dials enough digits to match a configured destination pattern, the telephone number is mapped to an IP host through the dial plan mapper. The IP host has a direct connection to either the destination telephone number or a PBX that completes the call to the configured destination pattern.

This document does not explain all possible serial interface configuration options, nor does it show LAN interface configuration. For complete information, see the Cisco IOS Release 12.0 *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

The "Configuration Example" section on page 40 shows a sample configuration that sets up VoIP over Frame Relay. For more information about setting up voice networks, see *Voice, Video, and Home Applications Configuration Guide* for Cisco IOS Release 12.0.

Note For information about monitoring serial interfaces in order to trigger a busyout condition on a voice port when an interface is down, see "Configuring Voice Ports" on page 17.

Step	Command	Purpose
1	`Router#` **`configure terminal`**	Enter global configuration mode.
2	`Router(config)#` **`interface serial`** *`slot/port:channel-group`*	Enter interface configuration mode for a serial interface that you specify by slot and port. The *channel-group* portion of the command is only required for channelized E1 interfaces. (For setting up channelized E1 interfaces, see *Dial Solutions Configuration Guide* for Cisco IOS Release 12.0.)
3	`Router(config-if)#` **`ip address`** *`ip-address mask`*	Assign the IP address and subnet mask to the interface.
4	`Router(config-if)#` **`isdn switch-type`** *`primary-qsig`*	Assign a switch type PRI or BRI interface, using primary-qsig for E1.
5	`Router(config-if)#` **`isdn protocol-emulate`** `[`*`user`*` | `*`network`*`]`	Configure the router's PRI interface so serve as either the primary QSIG slave or as the QSIG master.
6	`Router(config-if)#` **`isdn incoming-voice`** `[`*`data`*`[`*`56`*`|`*`64`*`]]|`*`modem`*`[`*`56`*`|`*`64`*`]]`	Route incoming calls to the modem and treat them as analog data, bypass the modem, or treat them as data.
7	`Router(config-if)#` **`fair-queue`** `[`*`congestive-discard-threshold`*` [`*`dynamic-queues`*` [`*`reservable-queues`*`]]]`	Initiate a fair-queue for congestion control.
8	`Router(config-if)#` **`exit`**	Exit the interface

Verifying Serial Interface Configuration

To verify serial interface configuration, enter the privileged EXEC command **show interfaces serial**, which displays the status of all serial interfaces or of a specific serial interface, as shown in the following example. You can use this command to check the encapsulation, IP addressing, and other settings:

```
Router #show interface serial0/0:0
Serial0/0:0 is up, line protocol is up
  Hardware is QUICC Serial
  Internet address is 1.156.1.1/24
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive not set
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     637 packets input, 64736 bytes, 0 no buffer
     Received 181 broadcasts, 0 runts, 5 giants, 0 throttles
     3617 input errors, 1506 CRC, 1646 frame, 0 overrun, 0 ignored, 0 abort
     682 packets output, 67213 bytes, 0 underruns
     0 output errors, 0 collisions, 1070 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

```
13 carrier transitions
Timeslot(s) Used:1-24, Transmitter delay is 0 flags
```

Configuring Voice Ports

Follow these steps to set up voice ports to support the local and remote stations. Not all possible commands are shown here. To learn more, see *Voice, Video, and Home Applications Configuration Guide* and *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0.

Step	Command	Purpose
1	`Router#` **`configure terminal`**	Enter global configuration mode.
2	`Router(config)#` **`voice-port`** *`slot/port:pri-group-no`*	Enter voice-port configuration mode. *slot* is the router location where the voice module is installed. Valid entries are from 0 to 3. *port* indicates the voice interface card location. Valid entries are 0 or 1. There is only one voice port per controller for QSIG. **Note** This **voice-port** command syntax does not apply to analog voice network modules and voice interface cards. The latter are specified using *slot*/*subunit*/*port*, designating the router slot for the voice network module, the location of the voice interface card in the network module, and the port on the voice interface card.
3	`Router(config-voice-port)#` **`busyout monitor interface`** *`interface number`*	(Optional) This command allows you to specify a LAN or WAN interface that will be monitored, and, when it is down, trigger a busyout (offhook) state on the voice port. This allows rerouting of calls. Busyout state for QSIG voice port implies that both the voice port and the signaling line is down. You can issue the command repeatedly to specify as many interfaces, virtual interfaces, and subinterfaces as are required for a voice port. For example, if you issue the command three times so that three interfaces are monitored, the voice port only goes into busyout state when all three interfaces are down. When any one of the interfaces is operational, the busyout state is removed.
4	`Router(config-voice-port)#` **`comfort-noise`**	(Optional) This parameter is enabled by default. It creates subtle background noise to fill silent gaps during calls when VAD is enabled on voice dial peers. If comfort noise is not generated, the silence can be unsettling to callers.
5	`Router(config-voice-port)#` **`echo-cancel enable`**	(Optional) This setting is enabled by default. Echo cancellation adds to the quality of voice transmissions by adjusting the echo that occurs on the interface due to impedance mismatches. Some echo is reassuring; echo over 25 milliseconds can cause problems.
6	`Router(config-voice-port)#` **`echo-cancel coverage {16 | 24 |32 | 8}`**	(Optional) This command adjusts the echo canceller by the specified number of milliseconds; the default is 16.

Step	Command	Purpose
7	`Router(config-voice-port)#` **`exit`**	Exit voice-port configuration mode. Repeat Steps 2 through 7 for each DS0 group you create.
8	`Router #` **`compand type`** `[a-law | u-law ]`	This command converts between analog and digital signals in PCM format. Specifying u-law is the North American mu-law ITU-T PCM encoding standard. Specifying a-law is the European a-law ITU-T PCM encoding standard.
9	`Router #` **`cp-tone`**	This command specifies a regional analog voice interface-related tone, ring, and cadence setting. the locale keyword specifies one of the following countries: argentina, australia, austria, belgium, brazil, china, colombia, czechrepublic, denmark, finland, france, germany, greece, hongkong, iceland, israel, italy, japan, korea, luxembourg, malaysia, netherlands, newzealand, northamerica, norway, peru, philippines, poland, portugal, russia, singapore, slovakia, southafrica, spain, sweden, switzerland, taiwan, thailand, turkey, unitedkingdom, and venezuela.

Verifying Voice Ports

Follow the procedure below to verify voice-port configuration. To learn more about these commands, see *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0.

Important command output is shown in bold.

To verify the voice-port configuration, enter the privileged EXEC **show voice port** *slot*/*port*:*ds0-group* command. The following sample output from the command shows explanatory information after the “<<“ characters:

```
cisco-router# show voice port 1/0:1

receEive and transMit Slot is 1, Sub-unit is 0, Port is 1  << voice-port 1/0:1
 Type of VoicePort is E&M
 Operation State is DORMANT
 Administrative State is UP
 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to -38 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 8 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
 Region Tone is set for US
```

Configuring Voice Dial Peers

Follow these steps to set up voice dial peers to support the local and remote stations. Not all possible commands are shown here. To learn more, see *Voice, Video, and Home Applications Configuration Guide* and *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0.

Step	Command	Purpose
1	`Router#` **`configure terminal`**	Enter global configuration mode.
2	`Router(config)#` **`dial-peer voice`** *`number`* **`pots`**	Enter dial-peer configuration mode and define a local dial peer that will connect to the plain old telephone service (POTS) network. *number* is one or more digits identifying the dial peer. Valid entries are from 1 to 2147483647. **pots** indicates a peer using basic telephone service.
3	`Router(config-dialpeer)#` **`destination-pattern`** *`string`* [**`T`**]	Configure the dial peer's destination pattern so that the system can reconcile dialed digits with a telephone number. *string* is a series of digits that specify the E.164 or private dialing plan phone number. Valid entries are the digits 0 through 9 and the letters A through D. The plus symbol (+) is not valid. The following special characters can be entered: • The star character (*) that appears on standard touch-tone dial pads can be in any dial string but not as a leading character (for example, *650). • The period (.) acts as a wildcard character. • The comma (,) can be used only in prefixes and inserts a one-second pause. When the timer (T) character is included at the end of the destination pattern, the system collects dialed digits as they are entered—until the interdigit timer expires (10 seconds, by default)—or the user dials the termination of end-of-dialing key (default is #). **Note** The timer character must be a capital T.
4	`Router(config-dialpeer)#` **`prefix`** *`string`*	(Optional) Include a dial-out prefix that the system enters automatically instead of people dialing it. *string* is a value from 0 to 9, and you can use a comma (,) to indicate a pause. **Note** There are other digit manipulation commands available to handle such situations as prefixes for special services, ignoring some digits, and dialing into remote PBXs as though they are local.
5	`Router(config-dialpeer)#` **`port`** *`slot/port:ds0-group-no`*	This command associates the dial peer with a specific logical interface. *slot* is the router location where the voice module is installed. Valid entries are from 0 to 3. *port* indicates the voice interface card location. Valid entries are 0 or 1. Each defined DS0 group number is represented on a separate voice port. This allows you to define individual DS0s on the digital E1 card.

Step	Command	Purpose
6	`Router(config)#` **`dial-peer voice`** *`number`* **`voip`**	Enter dial-peer configuration mode and define a remote VoIP dial peer. *number* is one or more digits identifying the dial peer. Valid entries are from 1 to 2147483647. **voip** indicates a VoIP peer using voice encapsulation on the IP network.
7	`Router(config-dialpeer)#` **`codec`** `{`**`g711alaw`** `|` **`g711ulaw`** `|` **`g723ar53`** `|` **`g723ar63`** `|` **`g723r53`** `|` **`g723r63`** `|` **`g726r16`** `|` **`g726r24`** `|` **`g726r32`** `|` **`g728`** `|` **`g729r8`** `[`**`pre-ietf`**`]` `|` **`g729br8`** `}` `[`*`bytes`*`]`	The voice-card configuration **codec complexity** command sets the codec options that are available when you execute this command. See Step 3 of the "Configuring Voice Card and E1 Controller Settings" section on page 12. If you do not set codec complexity, **g729r8** with IETF bit-ordering is used. If you set codec complexity to **high**, the following options are available: • **g711alaw**—G.711 A Law 64,000 bps • **g711ulaw**—G.711 u Law 64,000 bps • **g723ar53**—G.723.1 Annex A 5,300 bps • **g723ar63**—G.723.1 Annex A 6,300 bps • **g723r53**—G.723.1 5,300 bps • **g723r63**—G.723.1 6,300 bps • **g726r16**—G.726 16,000 bps • **g726r24**—G.726 24,000 bps • **g726r32**—G.726 32,000 bps • **g728**—G.728 16,000 bps • **g729r8**---G.729 8,000 bps (default) • **g729br8**—G.729 Annex B 8,000 bps If you set codec complexity to **medium**, the following options are valid: • **g711alaw**—G.711 A Law 64,000 bps • **g711ulaw**—G.711 u Law 64,000 bps • **g726r16**—G.726 16,000 bps • **g726r24**—G.726 24,000 bps • **g726r32**—G.726 32,000 bps • **g729r8**—G.729 Annex A 8,000 bps • **g729br8**—G.729 Annex B with Annex A 8,000 bps The optional *bytes* parameter sets the number of voice data bytes per frame. Acceptable values are from 10 to 240 in increments of 10 (for example, 10, 20, 30, and so on). Any other value is rounded down (for example, from 236 to 230). If you specify **g729r8**, then the IETF (Internet Engineering Task Force) bit-ordering is used. For interoperability with a Cisco 2600, 3600, or AS5300 router running a Cisco IOS release prior to Release 12.0(5)T or12.0(4)XH, you *must* specify the additional key word **pre-ietf** after **g729r8**.

Step	Command	Purpose
8	`Router(config-dialpeer)#` **vad**	(Optional) This setting is enabled by default. It activates voice activity detection (VAD). VAD allows the system to reduce unnecessary voice transmissions caused by unfiltered background noise.
9	`Router(config-dialpeer)#` **dtmf-relay** [**cisco-rtp**] [**h245-signal**] [**h245-alphanumeric**]	(Optional) Dual-tone multifrequency (DTMF) describes the tone that sounds in response to a keypress on a touch-tone phone. DTMF tones are compressed at one end of a call and decompressed at the other end. If a low-bandwidth codec, such as a G.729 or G.723, is used, the tones can sound distorted. The **dtmf-relay** command transports DTMF tones generated after call establishment out-of-band by using a method that transmits with greater fidelity than is possible in-band for most low-bandwidth codecs. Without DTMF relay, calls established with low-bandwidth codecs may have trouble accessing automated phone menu systems, such as voicemail and interactive voice response (IVR) systems. A signaling method is supplied only if the remote end supports it, and the options are: Cisco proprietary (**cisco-rtp**), standard H.323 (**h245-alphanumeric**), and H.323 standard with signal duration (**h245-signal**).
10	`Router(config-dialpeer)#` **fax-rate** {**2400** | **4800** | **7200** | **9600** | **12000** | **14400** | **disable** | **voice**}	(Optional) Specify the transmission speed of a fax to be sent to this dial peer. **disable** turns off fax transmission capability, and **voice** specifies the highest possible fax speed supported by the voice rate.
11	`Router(config-dialpeer)#` **destination-pattern** *string* [**T**]	See Step 3 in this procedure.
12	`Router(config-dialpeer)#` **session target** {**ipv4:***destination-address* | **dns:**[**s.** | **d.** | **e.** | **u.**] *host-name*}	Configure the IP session target for the dial peer. **ipv4:***destination-address* indicates IP address of the dial peer. **dns:***host-name* indicates that the domain name server will resolve the name of the IP address. Valid entries for this parameter are characters representing the name of the host device. There are also wildcards available for defining domain names with the keyword by using source, destination, and dialed information in the host name. For complete command syntax information, see *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0.
13	`Router(config-dialpeer)#` **forward-digit** [*all*] *default* | *extra* | ..]	Configure the interface to forward digits for voice calls.
14	`Router(config-dialpeer)#` **huntstop**	Disable hunting by the interface for dial peers.
15	`Router(config-dialpeer)#` **exit**	Exit interface configuration.

Verifying Voice Dial Peers

Follow the procedure below to verify dial-peer configuration. To learn more about these commands, see *Voice, Video, and Home Applications Command Reference* for Cisco IOS Release 12.0.

Important command output is shown in bold.

Enter the privileged EXEC **show dial-peer voice** command. The following text is sample output from the command for a POTS dial peer:

```
cisco-router# show dial-peer voice 1
VoiceEncapPeer1
        tag = 1, dest-pat = \Q+14085551000',
        answer-address = \Q',
        group = 0, Admin state is up, Operation state is down
        Permission is Both,
        type = pots, prefix = \Q',
        session-target = \Q', voice-port =
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is "10"
        Last Disconnect Text is ""
        Last Setup Time = 0
```

The following text is sample output from the **show dial-peer voice** command for a VoIP dial peer:

```
cisco-router# show dial-peer voice 10
VoiceOverIpPeer10
        tag = 10, dest-pat = \Q',
        incall-number = \Q+14087',
        group = 0, Admin state is up, Operation state is down
        Permission is Answer,
        type = voip, session-target = \Q',
        sess-proto = cisco, req-qos = bestEffort,
        acc-qos = bestEffort,
        fax-rate = voice, codec = g729r8,
        Expect factor = 10,Icpif = 30, VAD = disabled, Poor QOV Trap = disabled,
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is "10"
        Last Disconnect Text is ""
        Last Setup Time = 0
```

Monitoring and Maintaining E1 Digital Packet Voice Configuration

This section presents some useful show and debugging commands for understanding, maintaining, and troubleshooting your configuration.

Table 1 Debug and Show Commands for Maintaining and Troubleshooting Your Configuration

Command	Purpose
`Router#` **show dialplan number** *number*	Shows which dial-peer is matched by a called number.
`Router#` **show call active voice**	Shows statistics for currently active voice calls.
`Router#` **show call active fax**	Shows statistics for currently active fax calls.
`Router#` **show call history voice**	Shows statistics on previous voice calls.

Command	Purpose
Router# **show call history fax**	Shows statistics on previous fax calls.
Router# **show voice port**	Shows the status of voice ports. See "Verifying Voice Ports" on page 18.
Router# **show controller E1** *slot/port*	Shows the status of the E1 controller. See "Verifying Voice Card and Controller Settings" on page 15.
Router# **show isdn status**	Shows the status of an individual ISDN line.
Router# **debug ccapi inout**	Debugs the E1
Router# **debug isdn q931**	Debugs calls as they are set up and torn down on ISDN network connections (Layer 3) between the local router (user side) and the network.
Router# **debug vpm all**	Debugs the E1 signaling.
Router# **debug vtsp all**	Debugs the digits received and sent.
Router# **debug voip ccapi inout**	Debugs the call setup process.

The balance of this section shows the output of the commands listed in Table 1.

Show Commands

This section illustrates some of the privileged EXEC show commands that are useful for analyzing your system. Note that important information appears in bold, and bold text preceded by the "<<" characters explains the process.

The **show dialplan number** command provides information about the dial peer associated with a specified dial-plan number. Notice that the dial peer is operational and that IP Precedence has been configured to the preferred setting of 5.

Note To pair different voice ports and telephone numbers together for troubleshooting, enter the **show dialplan incall** *number* privileged EXEC command.

```
cisco-router# show dialplan number 75435
Macro Exp.: ##75435
VoiceOverIpPeer70000
        information type = voice,
        tag = 70000, destination-pattern = `##7....',
        answer-address = `', preference=0,
        group = 70000, Admin state is up, Operation state is up,
        incoming called-number = `', connections/maximum = 0/unlimited,
        DTMF Relay = disabled,
        application associated:
        type = voip, session-target = `ipv4:171.68.253.18',
        technology prefix:
        settlement: disabled
        ip precedence = 5, UDP checksum = disabled,
        session-protocol = cisco, req-qos = best-effort,
        acc-qos = best-effort,
        fax-rate = 14400,   payload size =  20 bytes
        codec = g729r8,   payload size =  20 bytes,
        Expect factor = 10, Icpif = 30,signaling-type = cas,
        VAD = disabled, Poor QOV Trap = disabled,
        Connect Time = 0, Charged Units = 0,
        Successful Calls = 3, Failed Calls = 0,
        Accepted Calls = 3, Refused Calls = 0,
```

```
        Last Disconnect Cause is "10  ",
        Last Disconnect Text is "normal call clearing.",
        Last Setup Time = 344813.
Matched: ##75435   Digits: 3
Target: ipv4:171.68.253.18
```

The **show call active voice** command displays information about a current call:

```
cisco-router# show call active voice

GENERIC:
SetupTime=94523746 ms
Index=448
PeerAddress=##73072
PeerSubAddress=
PeerId=70000
PeerIfIndex=37
LogicalIfIndex=0
ConnectTime=94524043
DisconectTime=94546241
CallOrigin=1
ChargedUnits=0
InfoType=2
TransmitPackets=6251
TransmitBytes=125020
ReceivePackets=3300
ReceiveBytes=66000
VOIP:
ConnectionId[0x142E62FB 0x5C6705AF 0x0 0x385722B0]
RemoteIPAddress=171.68.235.18
RemoteUDPPort=16580
RoundTripDelay=29 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
SessionProtocol=cisco
SessionTarget=ipv4:171.68.235.18
OnTimeRvPlayout=63690
GapFillWithSilence=0 ms
GapFillWithPrediction=180 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=70 ms
LoWaterPlayoutDelay=30 ms
ReceiveDelay=40 ms
LostPackets=0 ms
EarlyPackets=1 ms
LatePackets=18 ms
VAD = disabled
CoderTypeRate=g729r8
CodecBytes=20
cvVoIPCallHistoryIcpif=0
SignalingType=cas
```

The **show call history voice** command shows statistics about previous calls:

```
cisco-router# show call history voice

GENERIC:
SetupTime=94893250 ms
Index=450
PeerAddress=##52258
PeerSubAddress=
PeerId=50000
PeerIfIndex=35
LogicalIfIndex=0
```

```
DisconnectCause=10
DisconnectText=normal call clearing.
ConnectTime=94893780
DisconectTime=95015500
CallOrigin=1
ChargedUnits=0
InfoType=2
TransmitPackets=32258
TransmitBytes=645160
ReceivePackets=20061
ReceiveBytes=401220
VOIP:
ConnectionId[0x142E62FB 0x5C6705B3 0x0 0x388F851C]
RemoteIPAddress=171.68.235.18
RemoteUDPPort=16552
RoundTripDelay=23 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
SessionProtocol=cisco
SessionTarget=ipv4:171.68.235.18
OnTimeRvPlayout=398000
GapFillWithSilence=0 ms
GapFillWithPrediction=1440 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=97 ms
LoWaterPlayoutDelay=30 ms
ReceiveDelay=49 ms
LostPackets=1 ms
EarlyPackets=1 ms
LatePackets=132 ms
VAD = disabled
CoderTypeRate=g729r8
CodecBytes=20
cvVoIPCallHistoryIcpif=0
SignalingType=cas
```

The **show isdn status** command shows the status of ISDN calls:

```
cisco-router# show isdn status

Global ISDN Switchtype = primary-qsig
ISDN Serial1/015 interface
        ******* Network side configuration *******
        dsl 0, interface ISDN Switchtype = primary-qsig
         **** Master side configuration ****
    Layer 1 Status
        ACTIVE
    Layer 2 Status
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status
        24 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 24
        CCBcallid=E3C, sapi=0, ces=0, B-chan=1, calltype=VOICE
        CCBcallid=E3D, sapi=0, ces=0, B-chan=2, calltype=VOICE
        CCBcallid=E3E, sapi=0, ces=0, B-chan=3, calltype=VOICE
        CCBcallid=E3F, sapi=0, ces=0, B-chan=4, calltype=VOICE
        CCBcallid=E40, sapi=0, ces=0, B-chan=5, calltype=VOICE
        CCBcallid=E47, sapi=0, ces=0, B-chan=6, calltype=VOICE
        CCBcallid=E48, sapi=0, ces=0, B-chan=7, calltype=VOICE
        CCBcallid=E49, sapi=0, ces=0, B-chan=8, calltype=VOICE
        CCBcallid=E50, sapi=0, ces=0, B-chan=9, calltype=VOICE
        CCBcallid=E51, sapi=0, ces=0, B-chan=10, calltype=VOICE
        CCBcallid=E52, sapi=0, ces=0, B-chan=11, calltype=VOICE
        CCBcallid=E53, sapi=0, ces=0, B-chan=12, calltype=VOICE
```

```
        CCBcallid=E54, sapi=0, ces=0, B-chan=13, calltype=VOICE
        CCBcallid=E5B, sapi=0, ces=0, B-chan=14, calltype=VOICE
        CCBcallid=E5C, sapi=0, ces=0, B-chan=15, calltype=VOICE
        CCBcallid=E5D, sapi=0, ces=0, B-chan=17, calltype=VOICE
        CCBcallid=E5E, sapi=0, ces=0, B-chan=18, calltype=VOICE
        CCBcallid=E5F, sapi=0, ces=0, B-chan=19, calltype=VOICE
        CCBcallid=E60, sapi=0, ces=0, B-chan=20, calltype=VOICE
        CCBcallid=E61, sapi=0, ces=0, B-chan=21, calltype=VOICE
        CCBcallid=E62, sapi=0, ces=0, B-chan=22, calltype=VOICE
        CCBcallid=E63, sapi=0, ces=0, B-chan=23, calltype=VOICE
        CCBcallid=E64, sapi=0, ces=0, B-chan=24, calltype=VOICE
        CCBcallid=E6B, sapi=0, ces=0, B-chan=25, calltype=VOICE
    The Free Channel Mask  0xFE000000
    Total Allocated ISDN CCBs = 24
```

The **show dial-peer voice summary** command displays information about dial-peers that are active:

```
cisco-router# show dial-peer voice summary

dial-peer hunt 0
  TAG TYPE   ADMIN OPER PREFIX   DEST-PATTERN    PREF SESS-TARGET    PORT
    1 pots   up    up            3                0                  1/015
  100 voip   down  down          1                0   ipv41.2.79.7
  200 voip   down  down          1                0   ipv41.2.79.31
  300 vofr   up    up            1                0   Serial0/0 990
  400 voip   down  down          1                0   ipv45.5.5.2
```

The **show voice call summary** command displays a summary of all dial-peers that are active:

```
cisco-router# show voice call summary

PORT      CODEC    VAD VTSP STATE            VPM STATE
========= ======== === ==================== ========================
1/015.1  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.2  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.3  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.4  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.5  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.6  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.7  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.8  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.9  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.10 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.11 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.12 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.13 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.14 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.15 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.17 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.18 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.19 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.20 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.21 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.22 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.23 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.24 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.25 g729r8    y  S_CONNECT            S_TSP_CONNECT
```

The **show voice dsp** command displays current status of all DSP voice channels:

```
cisco-router# show voice dsp

                                 BOOT                         PAK
TYPE DSP CH CODEC    VERS STATE STATE   RST AI PORT     TS ABORT   TX/RX-PAK-CNT
==== === == ======== ==== ===== ======= === == ======= == ===== ===============
C549 010 00 g729r8    3.3 busy  idle      0  0 1/015     1     0     67400/85384
         01 g729r8     .8 busy  idle      0  0 1/015     7     0     67566/83623
         02 g729r8        busy  idle      0  0 1/015    13     0     65675/81851
         03 g729r8        busy  idle      0  0 1/015    20     0     65530/83610
C549 011 00 g729r8    3.3 busy  idle      0  0 1/015     2     0     66820/84799
         01 g729r8     .8 busy  idle      0  0 1/015     8     0     59028/66946
         02 g729r8        busy  idle      0  0 1/015    14     0     65591/81084
         03 g729r8        busy  idle      0  0 1/015    21     0     66336/82739
C549 012 00 g729r8    3.3 busy  idle      0  0 1/015     3     0     59036/65245
         01 g729r8     .8 busy  idle      0  0 1/015     9     0     65826/81950
         02 g729r8        busy  idle      0  0 1/015    15     0     65606/80733
         03 g729r8        busy  idle      0  0 1/015    22     0     65577/83532
C549 013 00 g729r8    3.3 busy  idle      0  0 1/015     4     0     67655/82974
         01 g729r8     .8 busy  idle      0  0 1/015    10     0     65647/82088
         02 g729r8        busy  idle      0  0 1/015    17     0     66366/80894
         03 g729r8        busy  idle      0  0 1/015    23     0     66339/82628
C549 014 00 g729r8    3.3 busy  idle      0  0 1/015     5     0     68439/84677
         01 g729r8     .8 busy  idle      0  0 1/015    11     0     65664/81737
         02 g729r8        busy  idle      0  0 1/015    18     0     65607/81820
         03 g729r8        busy  idle      0  0 1/015    24     0     65589/83889
C549 015 00 g729r8    3.3 busy  idle      0  0 1/015     6     0     66889/83331
         01 g729r8     .8 busy  idle      0  0 1/015    12     0     65690/81700
         02 g729r8        busy  idle      0  0 1/015    19     0     66422/82099
         03 g729r8        busy  idle      0  0 1/015    25     0     65566/83852
```

The **show voice trace** command displays a trace of all active voice transitions:

```
cisco-router# show voice trace

1/015 1  State Transitions (state, event) -> (state, event) ...
(S_NULL, E_TSP_INFO_IND) -> (S_SETUP_INDICATED, E_TSP_INFO_IND) ->
(S_SETUP_INDICATED, E_TSP_INFO_IND) -> (S_SETUP_INDICATED, E_CC_PROCEEDING) ->
(S_SETUP_INDICATED, E_CC_ALERT) -> (S_ALERTING, E_CC_BRIDGE) ->
(S_ALERTING, E_CC_CONNECT) -> (S_CONNECT, E_CC_CAPS_IND) ->
(S_CONNECT, E_CC_CAPS_ACK) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN) ->
(S_CONNECT, E_DSP_DTMF_DIGIT) -> (S_CONNECT, E_TIMER) ->
```

The **show adapi** command displays information about the call distribution application programming interface (CDAPI):

```
cisco-router# show cdapi

Registered CDAPI Applications/Stacks
====================================

Application TSP CDAPI Application Voice
        Application Type(s)  Voice Facility Signaling
        Application Level    Tunnel
        Application Mode     Enbloc

Signaling Stack ISDN
        Interface Se1/015

CDAPI Message Buffers
=====================

Used Msg Buffers 0, Free Msg Buffers 6400
Used Raw Buffers 0, Free Raw Buffers 3200
Used Large-Raw Buffers 0, Free Large-Raw Buffers 320
2600-1#
2600-1#
2600-1#s vo call 1/015.1
1/015 1  vtsp level 0 state = S_CONNECT

callid 0x0EDE B01 state S_TSP_CONNECT clld 1 cllg 3456546347
2600-1# ***DSP VOICE VP_DELAY STATISTICS***
Clk Offset(ms) -383401219, Rx Delay Est(ms) 61
Rx Delay Lo Water Mark(ms) 61, Rx Delay Hi Water Mark(ms) 90
        ***DSP VOICE VP_ERROR STATISTICS***
Predict Conceal(ms) 0, Interpolate Conceal(ms) 0
Silence Conceal(ms) 0, Retroact Mem Update(ms) 0
Buf Overflow Discard(ms) 20, Talkspurt Endpoint Detect Err 0
        ***DSP VOICE RX STATISTICS***
Rx Vox/Fax Pkts 286, Rx Signal Pkts 0, Rx Comfort Pkts 0
Rx Dur(ms) 24870, Rx Vox Dur(ms) 8510, Rx Fax Dur(ms) 0
Rx Non-seq Pkts 0, Rx Bad Hdr Pkts 0
Rx Early Pkts 0, Rx Late Pkts 0
        ***DSP VOICE TX STATISTICS***
Tx Vox/Fax Pkts 826, Tx Sig Pkts 0, Tx Comfort Pkts 0
Tx Dur(ms) 24870, Tx Vox Dur(ms) 24790, Tx Fax Dur(ms) 0
        ***DSP VOICE ERROR STATISTICS***
Rx Pkt Drops(Invalid Header) 0, Tx Pkt Drops(HPI SAM Overflow) 0
        ***DSP LEVELS***
TDM Bus Levels(dBm0) Rx -12.5 from PBX/Phone, Tx -13.2 to PBX/Phone
TDM ACOM Levels(dBm0) +0.0, TDM ERL Level(dBm0) +23.5
TDM Bgd Levels(dBm0) -12.1, with activity being voice
```

Debug Commands

This section illustrates some of the EXEC mode debug commands that are useful when analyzing and troubleshooting your system. Note that important information appears in bold, and bold text preceded by the “<<“ characters explains the process.

The **debug isdn q931** command displays information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network.

The **debug voip ccapi inout** EXEC command traces the execution path through the call control API, which serves as the interface between the call-session application and the underlying network-specific software.

During the capabilities exchange shown in the command output, both sides agree on what compression to use, and the **debug voip ccapi inout** output helps you determine what each side is negotiating.

You can use the output from these command to understand how calls are being handled by the router. This command shows how a call flows through the system. By using this debug level, you can see the call setup and teardown operations performed on both the telephony and network call legs:

```
cisco-router# debug isdn q931
cisco-router# debug voip ccapi inout

001041 ISDN Se1/015 RX <-  SETUP pd = 8  callref = 0x1EC5 << the originating call
001041          Sending Complete
001041          Bearer Capability i = 0x8090A3
001041          Channel ID i = 0xA98381
001041          Calling Party Number i = 0x91, '0987654321'
001041          Calling Party SubAddr i = 0x80, 'P123'
001041          Called Party Number i = 0x91, '2312'
001041          Called Party SubAddr i = 0x80, 'P321'
001041          High Layer Compat i = 0x9181
001041          Locking Shift to Codeset 5
001041          Codeset 5 IE 0x31  i = 0x80
001041          Codeset 5 IE 0x32  i = 0x80
0010180388626431 vtsp_tsp_call_setup_ind (sdb=0x81A57008, tdm_info=0x0,
tsp_info=0x81A8687C, calling_number=0987654321 called_number=2312
redirect_number=
oct3a=0x0) peer_tag=1
001041 vtsp_do_call_setup_ind
001041 vtsp_do_call_setup_ind Call ID=65557, guid=813EC4AC
001041 vtsp_do_call_setup_ind type=0, under_spec=0, name=, id0=0, id1=0,
id2=0,
calling=0987654321, called=2312
001041 vtsp_do_nomal_call_setup_ind
001041 cc_api_call_setup_ind (vdbPtr=0x81B4FEEC, callInfo={called=2312,
calling=0987654321, fdest=1 peer_tag=1},
callID=0x813EC41C)vtsp_open_voice_and_set_params
001041 dsp_close_voice_channel [1/01511] packet_len=8 channel_id=1
packet_id=75
001041 dsp_open_voice_channel_20 [1/01511] packet_len=16 channel_id=1
packet_id=74
alaw_ulaw_select=1 associated_signaling_channel=128 time_slot=0 serial_port=0
001041 dsp_encap_config [1/01511] packet_len=24 channel_id=1 packet_id=92
TransportProtocol 2 t_ssrc=0x0 r_ssrc=0x0 t_vpxcc=0x0 r_vpxcc=0x0
001041 dsp_set_playout_delay [1/01511] packet_len=18 channel_id=1
packet_id=76
mode=1 initial=60 min=4 max=200 fax_nom=300
001041 dsp_echo_canceller_control [1/01511] packet_len=10 channel_id=1
packet_id=66
flags=0x0
001041 dsp_set_gains [1/01511] packet_len=12 channel_id=1 packet_id=91
in_gain=0
out_gain=0
001041 dsp_vad_enable [1/01511] packet_len=10 channel_id=1 packet_id=78
thresh=-38
001041 cc_process_call_setup_ind (event=0x81C83D98) handed call to app
"SESSION"
001041 sess_appl ev(SSA_EV_CALL_SETUP_IND), cid(11), disp(0)
001041 ccCallSetContext (callID=0xB, context=0x81A4659C)
001041 ssaCallSetupInd finalDest cllng(0987654321), clled(2312)
001041 ssaSetupPeer cid(11) peer list  tag(200)
001041 ssaSetupPeer cid(11), destPat(2312), matched(1), prefix(),
peer(81BF501C)
001041 ccCallProceeding (callID=0xB, prog_ind=0x0)
001041 ccCallSetupRequest (peer=0x81BF501C, dest=, params=0x81A465B0 mode=0,
```

```
*callID=0x81C2FBA8)
001041 callingNumber=0987654321, calledNumber=2312, redirectNumber=
001041 accountNumber=, finalDestFlag=1,
guid=fe47.5e74.92c9.0017.0000.0000.0009.caf4
001041 peer_tag=200
001041 ccIFCallSetupRequest (vdbPtr=0x81AF0B9C, dest=,
callParams={called=2312,
calling=0987654321, fdest=1, voice_peer_tag=200}, mode=0x0)
001041 ccSaveDialpeerTag (callID=0xC8, dialpeer_ tag=
001041 vtsp_save_dialpeer_tag tag=
001041 ccCallSetContext (callID=0xC, context=0x81DC2EB4)
001041 vtsp[1/01511, 0.S_SETUP_INDICATED, E_CC_PROCEEDING]
act_proceeding
0010176093659136 ISDN Se1/015 TX ->  CALL_PROC pd = 8  callref = 0x9EC5
0010178259955276         Channel ID i = 0xA98381
001041 cc_api_call_proceeding(vdbPtr=0x81AF0B9C, callID=0xC,
      prog_ind=0x8)
001041 cid(12)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_PROCEEDING)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(0)fDest(0)
001041 -cid2(11)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_MAPPING)
001041 ssaIgnore cid(12), st(SSA_CS_CALL_SETTING),oldst(1), ev(20)
001050 cc_api_call_alert(vdbPtr=0x81AF0B9C, callID=0xC, prog_ind=0x8,
sig_ind=0x1)
001050 cid(12)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_ALERT)
oldst(SSA_CS_CALL_SETTING)cfid(-1)csize(0)in(0)fDest(0)
001050 -cid2(11)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_MAPPING)
001050 ccCallAlert (callID=0xB, prog_ind=0x8, sig_ind=0x1)
001050 ccConferenceCreate (confID=0x81C2FC08, callID1=0xB, callID2=0xC,
tag=0x0)
001050 cc_api_bridge_done (confID=0x3, srcIF=0x81AF0B9C, srcCallID=0xC,
dstCallID=0xB,
disposition=0, tag=0x0)
001050 vtsp[1/01511, 0.S_SETUP_INDICATED, E_CC_ALERT]
act_alert
001050 vtsp[1/01511, 0.S_ALERTING, E_CC_BRIDGE]
act_bridge
001050 cc_api_bridge_done (confID=0x3, srcIF=0x81B4FEEC, srcCallID=0xB,
dstCallID=0xC,
disposition=0, tag=0x0)
001050 cc_api_caps_ind (dstVdbPtr=0x81AF0B9C, dstCallId=0xC, srcCallId=0xB,
     caps={codec=0x887F, fax_rate=0x7F, vad=0x3, modem=0x81CC9F20
           codec_bytes=0, signal_type=3})
001050 cc_api_caps_ind (dstVdbPtr=0x81B4FEEC, dstCallId=0xB, srcCallId=0xC,
     caps={codec=0x4, fax_rate=0x2, vad=0x2, modem=0x1
           codec_bytes=30, signal_type=2})
001050 cc_api_caps_ack (dstVdbPtr=0x81B4FEEC, dstCallId=0xB, srcCallId=0xC,
     caps={codec=0x4, fax_rate=0x2, vad=0x2, modem=0x1
           codec_bytes=30, signal_type=2})
001050 vtsp[1/01511, 0.S_ALERTING, E_CC_CAPS_IND]
act_caps_ind
001050 act_caps_ind Encap 2, Vad 2, Codec 0x4, CodecBytes 30,
              FaxRate 2, FaxBytes 30,
              Sub-channel 10, Bitmask 0x0 SignalType 2
001050 cc_api_caps_ack (dstVdbPtr=0x81AF0B9C, dstCallId=0xC, srcCallId=0xB,
     caps={codec=0x4, fax_rate=0x2, vad=0x2, modem=0x1
           codec_bytes=30, signal_type=2})
001050 vtsp[1/01511, 0.S_ALERTING, E_CC_CAPS_ACK]
act_caps_ack
001050 dsp_idle_mode [1/01511] packet_len=8 channel_id=1 packet_id=68
001050 act_caps_ack codec = 15, ret = 1

001050 dsp_cp_tone_off [1/01511] packet_len=8 channel_id=1 packet_id=71
001050 dsp_idle_mode [1/01511] packet_len=8 channel_id=1 packet_id=68
001050 dsp_encap_config [1/01511] packet_len=24 channel_id=1 packet_id=92
TransportProtocol 3 SID_support=0 sequence_number=0 rotate_flag=0 header_bytes
```

```
0xA0
001050 dsp_voice_mode [1/01511] packet_len=22 channel_id=1 packet_id=73
coding_type=19 voice_field_size=30 VAD_flag=1 echo_length=64 comfort_noise=1
inband_detect=1 digit_relay=2
001050 cid(11)st(SSA_CS_CONFERENCING_ALERT)ev(SSA_EV_CONF_CREATE_DONE)
oldst(SSA_CS_MAPPING)cfid(3)csize(0)in(1)fDest(1)
001050 -cid2(12)st2(SSA_CS_CONFERENCING_ALERT)oldst2(SSA_CS_CALL_SETTING)
0010214748364800 ISDN Se1/015 TX ->  ALERTING pd = 8  callref = 0x9EC5
0010216914660940         Progress Ind i = 0x8181 - Call not end-to-end ISDN,
may have
in-band info
0010214748364800         Locking Shift to Codeset 5
0010216914660548         Codeset 5 IE 0x32  i = 0x80
001057 vtsp_process_dsp_message MSG_TX_DTMF_DIGIT_BEGIN digit=4
001057 vtsp[1/01511, 0.S_ALERTING, E_DSP_DTMF_DIGIT_BEGIN]
act_report_digit_begin
001057 cc_api_call_digit_begin (vdbPtr=0x81B4FEEC, callID=0xB, digit=4,
flags=0x1,
timestamp=0x0, expiration=0x0)
001057 cid(11)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_DIGIT_BEGIN)
oldst(SSA_CS_CONFERENCING_ALERT)cfid(3)csize(0)in(1)fDest(1)
001057 -cid2(12)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING)
001057 ccCallDigitBegin (callID=0xC, db=0x81C2FC2C)
001100 vtsp_process_dsp_message MSG_TX_DTMF_DIGIT_OFF digit=4,
duration=2510
001100 vtsp[1/01511, 0.S_ALERTING, E_DSP_DTMF_DIGIT]
act_report_digit_end
001100 vtsp_timer_stop 66005
001100 cc_api_call_digit (vdbPtr=0x81B4FEEC, callID=0xB, digit=4,
duration=2510)
001100 vtsp_timer_start 66006
001100 cid(11)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_CALL_DIGIT)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csize(0)in(1)fDest(1)
001100 -cid2(12)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING)
001100 ccCallDigitEnd (callID=0xC, de=0x81C2FC2C)
001100 cc_api_call_connected(vdbPtr=0x81AF0B9C, callID=0xC)
001100 cid(12)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_CALL_CONNECTED)
oldst(SSA_CS_CALL_SETTING)cfid(3)csize(0)in(0)fDest(0)
001100 -cid2(11)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CONFERENCED_ALERT)
001100 ccCallConnect (callID=0xB)
001100 ssaFlushPeerTagQueue cid(11) peer list (empty)
001100 vtsp[1/01511, 0.S_ALERTING, E_CC_CONNECT]
act_alert_connect
001100 vtsp_ring_noan_timer_stop 66035
001100 dsp_cp_tone_off [1/01511] packet_len=8 channel_id=1 packet_id=71
001164 ISDN Se1/015 TX ->  CONNECT pd = 8  callref = 0x9EC5
00112166296140         Progress Ind i = 0x8181 - Call not end-to-end ISDN,
may have
in-band info
001100         Connected Number i = 0x8933343536
001100         Connected SubAddr i = 0xA8333333B3
001100         Locking Shift to Codeset 5
00112166295748         Codeset 5 IE 0x32  i = 0x80
001100 ISDN Se1/015 RX <-  CONNECT_ACK pd = 8  callref = 0x1EC5
001110 vtsp_main timer 67006
001110 vtsp[1/01511, 0.S_CONNECT, E_TIMER]
act_dcollect_timer
001110 cc_api_call_digit (vdbPtr=0x81B4FEEC, callID=0xB, digit=T, duration=0)
001110 cid(11)st(SSA_CS_ACTIVE)ev(SSA_EV_CALL_DIGIT)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csize(0)in(1)fDest(1)
001110 -cid2(12)st2(SSA_CS_ACTIVE)oldst2(SSA_CS_CONFERENCED_ALERT)
001112 cc_api_call_disconnected(vdbPtr=0x81AF0B9C, callID=0xC, cause=0x1F)
001112 cid(12)st(SSA_CS_ACTIVE)ev(SSA_EV_CALL_DISCONNECTED)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csize(0)in(0)fDest(0)
001112 -cid2(11)st2(SSA_CS_ACTIVE)oldst2(SSA_CS_ACTIVE)
```

```
001112 ssa Disconnected cid(12) state(5) cause(0x1F)
001112 ccConferenceDestroy (confID=0x3, tag=0x0)
001112 cc_api_bridge_done (confID=0x3, srcIF=0x81AF0B9C, srcCallID=0xC,
dstCallID=0xB,
disposition=0 tag=0x0)
001112 vtsp[1/01511, 0.S_CONNECT, E_CC_BRIDGE_DROP]
act_bdrop
001112 dsp_cp_tone_off [1/01511] packet_len=8 channel_id=1 packet_id=71
001112 cc_api_bridge_done (confID=0x3, srcIF=0x81B4FEEC, srcCallID=0xB,
dstCallID=0xC,
disposition=0 tag=0x0)
001112 cid(11)st(SSA_CS_CONF_DESTROYING)ev(SSA_EV_CONF_DESTROY_DONE)
oldst(SSA_CS_ACTIVE)cfid(3)csize(0)in(1)fDest(1)
001112 -cid2(12)st2(SSA_CS_CONF_DESTROYING)oldst2(SSA_CS_ACTIVE)
001112 ccCallDisconnect (callID=0xB, cause=0x1F tag=0x0)
001112 ccCallDisconnect (callID=0xC, cause=0x1F tag=0x0)
001112 vtsp[1/01511, 0.S_CONNECT, E_CC_DISCONNECT]
act_disconnect
001112 vtsp_ring_noan_timer_stop 67247
001112 vtsp_cot_timer_stop 67247
001112 vtsp_timer_stop 67247
001112 dsp_get_error_stat [1/01511] packet_len=10 channel_id=1 packet_id=6
reset_flag=1
001112 vtsp_timer_start 67247
001112 cc_api_call_disconnect_done(vdbPtr=0x81AF0B9C, callID=0xC, disp=0,
tag=0x0)
001112 cid(12)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_DISCONNECT_DONE)
oldst(SSA_CS_ACTIVE)cfid(-1)csize(0)in(0)fDest(0)
001112 -cid2(11)st2(SSA_CS_DISCONNECTING)oldst2(SSA_CS_CONF_DESTROYING)
001112 vtsp[1/01511, 0.S_WAIT_STATS, E_DSP_GET_ERROR]
act_get_error
001112 1/01511 rx_dropped=0 tx_dropped=0 rx_control=34 tx_control=5
tx_control_dropped=0 dsp_mode_channel_1=2 dsp_mode_channel_2=0 c[0]=0 c[1]=0
c[2]=75
c[3]=75 c[4]=74 c[5]=92 c[6]=76 c[7]=66 c[8]=91 c[9]=78 c[10]=68 c[11]=71
c[12]=68
c[13]=92 c[14]=73 c[15]=71
001112 dsp_get_levels [1/01511] packet_len=8 channel_id=1 packet_id=89
001112 vtsp[1/01511, 0.S_WAIT_STATS, E_DSP_GET_LEVELS]
act_get_levels
001112 dsp_get_tx_stats [1/01511] packet_len=10 channel_id=1 packet_id=86
reset_flag=1
001112 vtsp[1/01511, 0.S_WAIT_STATS, E_DSP_GET_TX]
act_stats_complete
001112 vtsp_timer_stop 67249
001112 vtsp_ring_noan_timer_stop 67249
001112 dsp_idle_mode [1/01511] packet_len=8 channel_id=1 packet_id=68
001112 vtsp_timer_start 67249
001151539607616 ISDN Se1/015 TX ->  DISCONNECT pd = 8  callref = 0x9EC5
001153705903692         Cause i = 0x8086 - Channel unacceptable
001112 vtsp[1/01511, 0.S_WAIT_RELEASE, E_TSP_DISCONNECT_CONF]
act_wrelease_release
001112 vtsp_timer_stop 67250
001112 dsp_cp_tone_off [1/01511] packet_len=8 channel_id=1 packet_id=71
001112 dsp_idle_mode [1/01511] packet_len=8 channel_id=1 packet_id=68
001112 dsp_close_voice_channel [1/01511] packet_len=8 channel_id=1
packet_id=75
001112 vtsp[1/01511, 0.S_CLOSE_DSPRM, E_DSPRM_CLOSE_COMPLETE]
act_terminate
001112 cc_api_call_disconnect_done(vdbPtr=0x81B4FEEC, callID=0xB, disp=0,
tag=0x0)
001112 vtsp_free_cdb,cdb 0x81AB1244
001112 cid(11)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_DISCONNECT_DONE)
oldst(SSA_CS_CONF_DESTROYING)cfid(-1)csize(1)in(1)fDest(1)
001112 ISDN Se1/015 RX <-  RELEASE pd = 8  callref = 0x1EC5
```

```
001112         Cause i = 0x8086 - Channel unacceptable
001151539607552 ISDN Se1/015 TX ->  RELEASE_COMP pd = 8  callref = 0x9EC5
```

```
0029107374182399 ISDN BR1/0 TX ->  SETUP pd = 8  callref = 0x0001 << terminating call
0029105245511244         Bearer Capability i = 0x8090A3
0029103079215104         Channel ID i = 0xA98381
0029103079215104         Calling Party Number i = 0x91, '0987654321'
0029103079215104         Calling Party SubAddr i = 0x80, 'P123'
0029103079215104         Called Party Number i = 0x91, '312'
0029103079215104         Called Party SubAddr i = 0x80, 'P321'
0029103079215104         Sending Complete
0029103079215104         High Layer Compat i = 0x9181
0029103079215104         Locking Shift to Codeset 5
0029105245510852         Codeset 5 IE 0x31  i = 0x80
0029103079215104         Codeset 5 IE 0x32  i = 0x80
002925 ISDN BR1/0 RX <-  RELEASE_COMP pd = 8  callref = 0x8001
002925         Cause i = 0x8096 - Number changed
002925         Facility i = 0x91A4053132333435
002925         User-User i = 0x08, 'USER', 0x20, 'INFORMATION'
0030128849018944 ISDN BR1/0 TX ->  SETUP pd = 8  callref = 0x0002
0030131015315020         Bearer Capability i = 0x8090A3
0030128849018880         Channel ID i = 0xA98381
0030128849018880         Calling Party Number i = 0x91, '0987654321'
0030128849018880         Calling Party SubAddr i = 0x80, 'P123'
0030128849018880         Called Party Number i = 0x91, '312'
0030128849018880         Called Party SubAddr i = 0x80, 'P321'
0030128849018880         Sending Complete
0030128849018880         High Layer Compat i = 0x9181
0030128849018880         Locking Shift to Codeset 5
0030131015314628         Codeset 5 IE 0x31  i = 0x80
0030128849018880         Codeset 5 IE 0x32  i = 0x80
0030154618822720 ISDN BR1/0 TX ->  SETUP pd = 8  callref = 0x0002
0030156785118796         Bearer Capability i = 0x8090A3
0030154618822656         Channel ID i = 0xA98381
0030154618822656         Calling Party Number i = 0x91, '0987654321'
0030154618822656         Calling Party SubAddr i = 0x80, 'P123'
0030154618822656         Called Party Number i = 0x91, '312'
0030154618822656         Called Party SubAddr i = 0x80, 'P321'
0030154618822656         Sending Complete
0030154618822656         High Layer Compat i = 0x9181
0030154618822656         Locking Shift to Codeset 5
0030156785118404         Codeset 5 IE 0x31  i = 0x80
0030154618822656         Codeset 5 IE 0x32  i = 0x80
003037 ISDN BR1/0 RX <-  CALL_PROC pd = 8  callref = 0x8002
003037         Channel ID i = 0xA98381
003050 ISDN BR1/0 RX <-  PROGRESS pd = 8  callref = 0x8002
003050         Progress Ind i = 0x8181 - Call not end-to-end ISDN, may have
in-band
info
003050         Locking Shift to Codeset 5
003050         Codeset 5 IE 0x31  i = 0x80
003050         Codeset 5 IE 0x32  i = 0x80
003059 ISDN BR1/0 RX <-  ALERTING pd = 8  callref = 0x8002
003059         Progress Ind i = 0x8181 - Call not end-to-end ISDN, may have
in-band
info
003059         Locking Shift to Codeset 5
003059         Codeset 5 IE 0x31  i = 0x80
003059         Codeset 5 IE 0x32  i = 0x80
003103 ISDN BR1/0 RX <-  CONNECT pd = 8  callref = 0x8002
```

```
003103          Progress Ind i = 0x8181 - Call not end-to-end ISDN, may have
in-band
info
003103          Connected Number i = 0x8933343536
003103          Connected SubAddr i = 0xA8333333B3
003103          Locking Shift to Codeset 5
003103          Codeset 5 IE 0x31  i = 0x80
003103          Codeset 5 IE 0x32  i = 0x80
003112884901952 ISDN BR1/0 TX ->  CONNECT_ACK pd = 8  callref = 0x0002
003109 ISDN BR1/0 RX <-  DISCONNECT pd = 8  callref = 0x8002
003109          Cause i = 0x8186 - Channel unacceptable
003138654705664 ISDN BR1/0 TX ->  RELEASE pd = 8  callref = 0x0002
003140821001804         Cause i = 0x8086 - Channel unacceptable
003115 ISDN BR1/0 RX <-  RELEASE_COMP pd = 8  callref = 0x8002
003115          Cause i = 0x8096 - Number changed
003115          Facility i = 0x91A4053132333435
003115          User-User i = 0x08, 'USER', 0x20, 'INFORMATION'
003234359738368 ISDN BR1/0 TX ->  SETUP pd = 8  callref = 0x0003
003236526034508         Bearer Capability i = 0x8090A3
003234359738368         Channel ID i = 0xA98381
003234359738368         Calling Party Number i = 0x91, '0987654321'
003234359738368         Calling Party SubAddr i = 0x80, 'P123'
003234359738368         Called Party Number i = 0x91, '312'
003234359738368         Called Party SubAddr i = 0x80, 'P321'
003234359738368         Sending Complete
003234359738368         High Layer Compat i = 0x9181
003234359738368         Locking Shift to Codeset 5
003236526034116         Codeset 5 IE 0x31  i = 0x80
003234359738368         Codeset 5 IE 0x32  i = 0x80
003209 ISDN BR1/0 RX <-  CALL_PROC pd = 8  callref = 0x8003
003209          Channel ID i = 0xA98381
003224 ISDN BR1/0 RX <-  PROGRESS pd = 8  callref = 0x8003
003224          Progress Ind i = 0x8181 - Call not end-to-end ISDN, may have
in-band
info
003224          Locking Shift to Codeset 5
003224          Codeset 5 IE 0x31  i = 0x80
003224          Codeset 5 IE 0x32  i = 0x80
003234 ISDN BR1/0 RX <-  CONNECT pd = 8  callref = 0x8003
003234          Progress Ind i = 0x8181 - Call not end-to-end ISDN, may have
in-band
info
003234          Connected Number i = 0x8933343536
003234          Connected SubAddr i = 0xA8333333B3
003234          Locking Shift to Codeset 5
003234          Codeset 5 IE 0x31  i = 0x80
003234          Codeset 5 IE 0x32  i = 0x80
0032146028888128 ISDN BR1/0 TX ->  CONNECT_ACK pd = 8  callref = 0x0003
003251 ISDN BR1/0 RX <-  DISCONNECT pd = 8  callref = 0x8003
003251          Cause i = 0x8186 - Channel unacceptable
0032219043332096 ISDN BR1/0 TX ->  RELEASE pd = 8  callref = 0x0003
0032221209628236         Cause i = 0x8086 - Channel unacceptable
003255 ISDN BR1/0 RX <-  RELEASE_COMP pd = 8  callref = 0x8003
003255          Cause i = 0x8096 - Number changed
003255          Facility i = 0x91A4053132333435
003255          User-User i = 0x08, 'USER', 0x20, 'INFORMATION'
```

Table 2 explains the codec negotiation values that appear—in hexadecimal format— during the capabilities exchange portion of the command output.

Table 2 Codec Negotiation Values in debug voip ccapi inout

Negotiation Value in Decimal	Meaning
1	U-law PCM (g711ulaw)
2	A-law PCM (g711alaw)
3	32k ADPCM (g726r32)
4	24k ADPCM (g726r24)
5	16k ADPCM (g726r16)
6	CS-ACELP - pre-IETF (g729r8 pre-ietf)
7	medium complexity CS-ACELP - pre-IETF (g729ar8 pre-ietf)
8	CS-ACELP with VAD (g729br8)
9	medium complexity CS-ACELP with VAD (G.729abr8)
10	16K LD-CELP (g728)
11	G.723.1 High Rate - 6300 bps (g723r63)
12	G.723.1 High Rate with VAD - 6300 bps (g723ar63)
13	G.723.1 Low Rate - 5300 bps (g723r53)
14	G.723.1 Low Rate with VAD - 5300 bps (g723ar53)
19	CS-ACELP - IETF standard (g729r8)
20	medium complexity CS-ACELP - IETF standard (g729ar8)

Reference Information

The information in this section helps you interpret the output from **debug** and **show** commands.

Table 3 shows Q.931 call disconnection causes. In the examples that follow, the disconnects are caused by normal call clearing.

Table 3 Q.931 Call Disconnection Causes

Call Disconnection Cause Value	Meaning and Number
CC_CAUSE_UANUM = 0x1	/* unassigned number. (1) */
CC_CAUSE_NO_ROUTE = 0x3	/* no route to destination. (3) */
CC_CAUSE_NORM = 0x10	/* normal call clearing. (16) */
CC_CAUSE_BUSY = 0x11	/* user busy. (17) */
CC_CAUSE_NORS = 0x12	/* no user response. (18) */
CC_CAUSE_NOAN = 0x13	/* no user answer. (19) */
CC_CAUSE_REJECT = 0x15	/* call rejected. (21) */
CC_CAUSE_INVALID_NUMBER = 0x1C	/* invalid number. (28) */
CC_CAUSE_UNSP = 0x1F	/* normal, unspecified. (31) */

Call Disconnection Cause Value	Meaning and Number
CC_CAUSE_NO_CIRCUIT = 0x22	/* no circuit. (34) */
CC_CAUSE_NO_REQ_CIRCUIT = 0x2C	/* no requested circuit. (44) */
CC_CAUSE_NO_RESOURCE = 0x2F	/* no resource. (47) */
CC_CAUSE_NOSV = 0x3F	/* service or option not available, Unspecified. (63) */
CC_CAUSE_UNINITIALIZED = 0	/* un-initialized (0) */
CC_CAUSE_UANUM = 1	/* unassigned num */
CC_CAUSE_NO_ROUTE_TO_TRANSIT_NETWORK = 2	
CC_CAUSE_NO_ROUTE = 3	/* no rt to dest */
CC_CAUSE_SEND_INFO_TONE = 4	
CC_CAUSE_MISDIALLED_TRUNK_PREFIX = 5	
CC_CAUSE_CHANNEL_UNACCEPTABLE = 6	
CC_CAUSE_CALL_AWARDED = 7	
CC_CAUSE_PREEMPTION = 8	
CC_CAUSE_PREEMPTION_RESERVED = 9	
CC_CAUSE_NORM = 16	
CC_CAUSE_BUSY = 17	/* user busy */
CC_CAUSE_NORS = 18	/* no user response*/
CC_CAUSE_NOAN = 19	/* no user answer. */
CC_CAUSE_SUBSCRIBER_ABSENT = 20	
CC_CAUSE_REJECT = 21	/* call rejected. */
CC_CAUSE_NUMBER_CHANGED = 22	
CC_CAUSE_NON_SELECTED_USER_CLEARING = 26	
CC_CAUSE_DESTINATION_OUT_OF_ORDER = 27	
CC_CAUSE_INVALID_NUMBER = 28	
CC_CAUSE_FACILITY_REJECTED = 29	
CC_CAUSE_RESPONSE_TO_STATUS_ENQUIRY = 30	
CC_CAUSE_UNSP = 31	/* unspecified. */
CC_CAUSE_NO_CIRCUIT = 34	/* no circuit. */
CC_CAUSE_REQUESTED_VPCI_VCI_NOT_AVAILABLE = 35	
CC_CAUSE_VPCI_VCI_ASSIGNMENT_FAILURE = 36	
CC_CAUSE_CELL_RATE_NOT_AVAILABLE = 37	
CC_CAUSE_NETWORK_OUT_OF_ORDER = 38	
CC_CAUSE_PERM_FRAME_MODE_OUT_OF_SERVICE = 39	
CC_CAUSE_PERM_FRAME_MODE_OPERATIONAL = 40	
CC_CAUSE_TEMPORARY_FAILURE = 41	

Call Disconnection Cause Value	Meaning and Number
CC_CAUSE_SWITCH_CONGESTION = 42	
CC_CAUSE_ACCESS_INFO_DISCARDED = 43	
CC_CAUSE_NO_REQ_CIRCUIT = 44	
CC_CAUSE_NO_VPCI_VCI_AVAILABLE = 45	
CC_CAUSE_PRECEDENCE_CALL_BLOCKED = 46	
CC_CAUSE_NO_RESOURCE = 47	/* no resource. */
CC_CAUSE_QOS_UNAVAILABLE = 49	
CC_CAUSE_FACILITY_NOT_SUBCRIBED = 50	
CC_CAUSE_CUG_OUTGOING_CALLS_BARRED = 53	
CC_CAUSE_CUG_INCOMING_CALLS_BARRED = 55	
CC_CAUSE_BEARER_CAPABILITY_NOT_AUTHORIZED = 57	
CC_CAUSE_BEARER_CAPABILITY_NOT_AVAILABLE = 58	
CC_CAUSE_INCONSISTENCY_IN_INFO_AND_CLASS = 62	
CC_CAUSE_NOSV = 63	/* service or option * not available * unspecified. */
CC_CAUSE_BEARER_CAPABILITY_NOT_IMPLEMENTED = 65	
CC_CAUSE_CHAN_TYPE_NOT_IMPLEMENTED = 66	
CC_CAUSE_FACILITY_NOT_IMPLEMENTED = 69	
CC_CAUSE_RESTRICTED_DIGITAL_INFO_BC_ONLY = 70	
CC_CAUSE_SERVICE_NOT_IMPLEMENTED = 79	
CC_CAUSE_INVALID_CALL_REF_VALUE = 81	
CC_CAUSE_CHANNEL_DOES_NOT_EXIST = 82	
CC_CAUSE_CALL_EXISTS_CALL_ID_IN_USE = 83	
CC_CAUSE_CALL_ID_IN_USE = 84	
CC_CAUSE_NO_CALL_SUSPENDED = 85	
CC_CAUSE_CALL_CLEARED = 86	
CC_CAUSE_USER_NOT_IN_CUG = 87	
CC_CAUSE_INCOMPATIBLE_DESTINATION = 88	
CC_CAUSE_NON_EXISTENT_CUG = 90	
CC_CAUSE_INVALID_TRANSIT_NETWORK = 91	
CC_CAUSE_AAL_PARMS_NOT_SUPPORTED = 93	
CC_CAUSE_INVALID_MESSAGE = 95	
CC_CAUSE_MANDATORY_IE_MISSING = 96	
CC_CAUSE_MESSAGE_TYPE_NOT_IMPLEMENTED = 97	
CC_CAUSE_MESSAGE_TYPE_NOT_COMPATIBLE = 98	
CC_CAUSE_IE_NOT_IMPLEMENTED = 99	
CC_CAUSE_INVALID_IE_CONTENTS = 100	

Call Disconnection Cause Value	Meaning and Number
CC_CAUSE_MESSAGE_IN_INCOMP_CALL_STATE = 101	
CC_CAUSE_RECOVERY_ON_TIMER_EXPIRY = 102	
CC_CAUSE_NON_IMPLEMENTED_PARAM_PASSED_ON = 103	
CC_CAUSE_UNRECOGNIZED_PARAM_MSG_DISCARDED = 110	
CC_CAUSE_PROTOCOL_ERROR = 111	
CC_CAUSE_INTERWORKING = 127	

Table 4 Tone Types and Their Meanings

Tone Type	Meaning
CC_TONE_RINGBACK	0x1 - Ring Tone
CC_TONE_FAX	0x2 - Fax Tone
CC_TONE_BUSY	0x4 - Busy Tone
CC_TONE_DIALTONE	0x8 - Dial Tone
CC_TONE_OOS	0x10 - Out of Service Tone
CC_TONE_ADDR_ACK	0x20 - Address Acknowledgement Tone
CC_TONE_DISCONNECT	0x40 - Disconnect Tone
CC_TONE_OFF_HOOK_NOTICE	0x80 - Tone indicating the phone was left off hook
CC_TONE_OFF_HOOK_ALERT	0x100 /* A more urgent version of CC_TONE_OFF_HOOK_NOTICE*/
CC_TONE_CUSTOM	0x200 - Custom Tone - used when specifying a custom tone
CC_TONE_NULL	0x0 - Null Tone

These are codec capabilities bits that can appear in command output:

- CC_CAP_CODEC_G711U 0x1
- CC_CAP_CODEC_G711A 0x2
- CC_CAP_CODEC_G723ar63 0x2000
- CC_CAP_CODEC_G723ar53 0x4000
- CC_CAP_CODEC_G723r63 0x100
- CC_CAP_CODEC_G723r53 0x200
- CC_CAP_CODEC_G726r16 0x10
- CC_CAP_CODEC_G729 0x4
- CC_CAP_CODEC_G729 0x8000
- CC_CAP_CODEC_G729a 0x8
- CC_CAP_CODEC_G729b 0x800
- CC_CAP_CODEC_G729ab 0x1000

These are fax capabilities bits that can appear in command output. The numbers following “FAX_” refer to the fax speed (for example, “144” means 14,400 bps):

- CC_CAP_FAX_NONE 0x1
- CC_CAP_FAX_VOICE 0x2
- CC_CAP_FAX_144 0x4
- CC_CAP_FAX_96 0x8
- CC_CAP_FAX_72 0x10
- CC_CAP_FAX_48 0x20
- CC_CAP_FAX_24 0x40
- CC_CAP_FAX_120 0x80

These are the VAD on and off capability bits:

- CC_CAP_VAD_OFF 0x1
- CC_CAP_VAD_ON 0x2

Configuration Example

This section includes the following configuration example:

```
cisco-router#show running-config


Building configuration...

Current configuration
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 2600-1
!
enable secret 5 $1$5O8W$pzps91xiu3/avMQNyyZQb.
enable password ard
!
!
!
!
!
memory-size iomem 10
voice-card 1
!
ip subnet-zero
no ip domain-lookup
!
frame-relay switching
isdn switch-type primary-qsig
isdn voice-call-failure 0
voice hunt user-busy
!
!
!
!
controller E1 1/0
 pri-group timeslots 1-31
!
controller E1 1/1
 shutdown
!
!
!
interface Ethernet0/0
 ip address 1.2.79.1 255.255.0.0
 no ip directed-broadcast
 no cdp enable
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 load-interval 30
 clockrate 800000
 frame-relay traffic-shaping
 frame-relay class voice-vc
 frame-relay interface-dlci 990
  vofr data 4 call-control 5
 frame-relay intf-type dce
```

```
!
interface Ethernet0/1
 no ip address
 no ip directed-broadcast
 shutdown
 no cdp enable
!
interface Serial0/1
 ip address 5.5.5.1 255.0.0.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 clockrate 800000
 frame-relay traffic-shaping
 frame-relay class voice-data
 frame-relay interface-dlci 991
 frame-relay ip rtp header-compression
 frame-relay intf-type dce
!
interface Serial1/015
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn overlap-receiving
 isdn protocol-emulate network
 isdn incoming-voice voice
 no isdn T309-enable
 isdn bchan-number-order ascending
 fair-queue 64 256 0
 no cdp enable
!
router rip
 network 172.28.0.0
!
 router igrp 1
 redistribute connected
 network 1.0.0.0
!
ip default-gateway 1.2.0.1
ip classless
ip route 223.255.254.254 255.255.255.255 1.2.0.1
no ip http server
!
!
map-class frame-relay voice-vc
 no frame-relay adaptive-shaping
 frame-relay cir 512000
 frame-relay bc 512000
 frame-relay fair-queue
 frame-relay voice bandwidth 512000
 frame-relay fragment 100
!
map-class frame-relay voice-data
 no frame-relay adaptive-shaping
 frame-relay cir 512000
 frame-relay bc 1000
 frame-relay fair-queue
 frame-relay fragment 200
 frame-relay ip rtp priority 2000 16383 500
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
no cdp run
!
```

```
voice-port 1/015
 compand-type a-law
!
dial-peer voice 1 pots
 destination-pattern 3
 direct-inward-dial
 port 1/015
 forward-digits all
!
dial-peer voice 100 voip
 shutdown
 destination-pattern 1
 session target ipv41.2.79.7
!
dial-peer voice 200 voip
 shutdown
 destination-pattern 1
 session target ipv41.2.79.31
!
dial-peer voice 300 vofr
 destination-pattern 1
 session target Serial0/0 990
!
dial-peer voice 400 voip
 shutdown
 destination-pattern 1
 session target ipv45.5.5.2
!
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password ard
 login
!
end
```

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command references.

- **pri-group**

pri-group

To specify a ISDN Primary Rate interface (PRI) on a channelized T1 or E1 controller, enter the **pri-group** controller configuration command. Enter the **no** form of this command removes the remove the ISDN-PRI configuration.

pri-group timeslots *timeslot-range*

no pri-group

Syntax Description

timeslot-range	*timeslot-list* is a single timeslot number, a single range of values. For T1, the allowable range is from 1 to 23. For E1, the allowable values are from 1 to 15.

Default

There is no ISDN-PRI group configured.

Command Mode

Controller configuration

Command History

Release	Modification
12.0(2)T	The command was introduced for the Cisco MC3810 multiservice access concentrator.
12.0(7)XK	The command was introduced for the Cisco 2600 and 3600 series with a different name and some keyword modifications.

Usage Guidelines

The **pri-group** command applies to the configuration of Voice over Frame Relay, Voice over ATM, and Voice over HDLC on the Cisco MC3810 multiservice concentrator and the Cisco 2600 and 3600 series routers.

Before you enter the **pri-group** command, you must specify an ISDN-PRI switch type and an E1 or T1 controller. Only one pri group can be configured on a controller.

Example

The following example configures ISDN-PRI on all timeslots of controller E1 1 on a Cisco 2600 series router:

```
cisco-router# pri-group timeslots 1-7, 16

controller E1 4/0
!
controller E1 4/1
 pri-group timeslots 1-7,16
!
```

Related Command

Command	Description
isdn switch-type	To configure the Cisco 2600 series router PRI interface to support QSIG signalling, enter this command.

Glossary

AAL—ATM Adaptation Layer. Service-dependent sublayer of the data link layer. The AAL accepts data from different applications and presents it to the ATM layer in the form of 48-byte ATM payload segments. AALs consist of two sublayers: convergence sublayer (CS) and segmentation and reassembly (SAR). AALs differ on the basis of the source-destination timing used, whether they use constant bit rate (CBR) or variable bit rate (VBR), and whether they are used for connection-oriented or connectionless mode data transfer. At present, the four types of AAL recommended by the ITU-T are AAL1, AAL2, AAL3/4, and AAL5.

AAL1—ATM adaptation layer 1. One of four AALs recommended by the ITU-T. AAL1 is used for connection-oriented, delay-sensitive services requiring constant bit rates, such as uncompressed video and other isochronous traffic.

AMI—alternate mark inversion. Line-code type used on T1 and E1 circuits. In AMI, zeros are represented by 01 during each bit cell, and ones are represented by 11 or 00, alternately, during each bit cell. AMI requires that the sending device maintain ones density. Ones density is not maintained independent of the data stream. Sometimes called *binary coded alternate mark inversion*.

ATM—Asynchronous Transfer Mode. International standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

B8ZS—binary 8-zero substitution. Line-code type, used on T1 and E1 circuits, in which a special code is substituted whenever 8 consecutive zeros are sent over the link. This code is then interpreted at the remote end of the connection. This technique guarantees ones density independent of the data stream.

CAS—channel-associated signaling. Trunk signaling (for example, in a T1 line) in which control signals, such as those for synchronizing and bounding frames, are carried in the same channel along with voice and data signals.

CBR—constant bit rate. QoS class defined by the ATM Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery.

CCS—common channel signaling. Trunk signaling (for example, using Primary Rate Interface) in which a control channel carries signaling for separate voice and data channels.

CES—circuit emulation service. Enables users to multiplex or concentrate multiple circuit emulation streams for voice and video with packet data on a single high-speed ATM link without a separate ATM access multiplexer.

CO—central office. Local telephone company office to which all local loops in a given area connect and in which circuit switching of subscriber lines occurs.

codec—Coder-decoder. Device that typically uses pulse code modulation to transform analog signals into a digital bit stream and digital signals back into analog.

DTMF—Dual-tone multifrequency. Use of two simultaneous voice-band tones for dialing (such as touch tone).

Drop and Insert—(also called TDM Cross-Connect) Allows DSO channels from one T1 or E1 facility to be digitally cross-connected to DS0 channels on another T1 or E1. Using this method, channel traffic is sent between a PBX and CO PSTN switch or other telephony device, so that some PBX channels are directed for long-distance service through the PSTN while the router compresses others for interoffice VoIP calls. In addition, Drop and Insert can cross-connect a telephony switch (from the CO or PSTN) to a channel bank for external analog connectivity.

DSP—digital signal processor, same as PVDM

E1—European digital carrier facility used for transmitting data through the telephone hierarchy. The transmission rate for E1 is 2.048 megabits per second (Mbps).

E&M—rEceive and transMit, or Ear and Mouth. Type of signaling originally developed for analog two-state voltage telephony using the ear and mouth leads; in digital telephony, uses two bits.

ESF—Extended Superframe. Framing type used on T1 circuits that consists of 24 frames of 192 bits each, with the 193rd bit providing timing and other functions. ESF is an enhanced version of SF format.

FXO—Foreign Exchange Office. A voice interface emulating a PBX trunk line to a switch or telephone equipment to a PBX extension interface.

FXS—Foreign Exchange Station. A voice interface for connecting telephone equipment, emulates the extension interface of a PBX or the subscriber interface for a switch.

IETF—Internet Engineering Task Force

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

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

packet—Logical grouping of information that includes a header containing control information and (usually) user data. Packets are most often used to refer to network layer units of data.

POTS—plain old telephone service

PDVM—packet data voice module

PSTN—Public Switched Telephone Network. General term referring to the variety of telephone networks and services in place worldwide.

QoS—quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

SF—Super Frame. Common framing type used on T1 circuits. SF consists of 12 frames of 192 bits each, with the 193rd bit providing error checking and other functions. SF is superseded by ESF, but is still widely used. Also called D4 framing.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

T1—Digital WAN carrier facility. T1 transmits DS 1-formatted data at 1.544 Mbps through the telephone switching network, using alternate mark inversion or B8ZS coding.

T1 trunk—Digital WAN carrier facility. See T1.

TDM—time-division multiplexing

Trunk—Physical and logical connection between two switches across which network traffic travels. A backbone is composed of a number of trunks.

UNI—User-Network Interface. ATM Forum specification that defines an interoperability standard for the interface between ATM-based products (a router or an ATM switch) located in a private network and the ATM switches located within the public carrier networks. Also used to describe similar connections in Frame Relay networks.

VAD—voice activity detection