



Integrating Data and Voice Services for ISDN PRI Interfaces on Multiservice Access Routers

Revised: June 19, 2006, OL-10383-01
First Published: February 27, 2006

This document describes how to configure ISDN PRI interfaces to support the integration of data and voice calls on multiservice access routers. This feature enables data (dial-in, dial-on-demand routing [DDR], and DDR backup) and voice call traffic to occur simultaneously from the supported ISDN PRI interfaces. You can also enable multilevel precedence and preemption (MLPP) for DDR calls over the active voice call when no idle channel is available during the DDR call setup.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the [“Feature Information for Integrating Data and Voice Services for ISDN PRI Interfaces”](#) section on page 71.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Integrating Data and Voice Services for ISDN PRI Interfaces](#), page 2
- [Restrictions for Integrating Data and Voice Services for ISDN PRI Interfaces](#), page 3
- [Information About Integrating Data and Voice Services for ISDN PRI Interfaces](#), page 4
- [How to Configure Integrated Data and Voice Services for ISDN PRI Interfaces](#), page 6
- [Troubleshooting Tips for Integrated Data and Voice Services](#), page 22
- [Configuration Examples for Integrating Data and Voice Services for ISDN PRI Interfaces](#), page 23
- [Additional References](#), page 39



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

© 2006 Cisco Systems, Inc. All rights reserved.

- [Command Reference, page 40](#)
- [Feature Information for Integrating Data and Voice Services for ISDN PRI Interfaces, page 71](#)
- [Glossary, page 71](#)

Prerequisites for Integrating Data and Voice Services for ISDN PRI Interfaces

- Establish a working H.323 or SIP network for voice calls.
- Ensure that you have a Cisco IOS image that supports this feature. Access Cisco Feature Navigator at <http://www.cisco.com/go/cfn>.
- Perform basic ISDN PRI voice configuration, including dial-on demand routing (DDR) configuration for data calls. For more information, see [Configuring ISDN PRI Voice-Interface Support](#).
- To support PRI data calls, a VWIC-1MFT-E1 voice cards must have a packet voice data module (PVDM).

Supported Modules

- This feature supports the following modules:
 - NM-HD
 - NM-HDV2
 - Onboard DSPs
- This feature supports the following voice cards:
 - VWIC-XMFT-X interface modules
 - VWIC2-XMFT-X interface modules



Note Data calls are supported only on the NM-HDV2-2T1/E1 and NM-HD-2V-E network modules, and the VWIC-2MFT-E1, VWIC-2MFT-T1 and VWIC2-T1/E1 voice cards.

Use the **isdn switch-type ?** command in interface configuration mode or global configuration mode to view the list of supported ISDN switch types. See the following example:

```
Router(config)# isdn switch-type ?
primary-4ess    Lucent 4ESS switch type for the U.S.
primary-5ess    Lucent 5ESS switch type for the U.S.
primary-dms100  Northern Telecom DMS-100 switch type for the U.S.
primary-dpnss   DPNSS switch type for Europe
primary-net5    NET5 switch type for UK, Europe, Asia and Australia
primary-ni      National ISDN Switch type for the U.S.
primary-ntt     NTT switch type for Japan
primary-qsig    QSIG switch type
primary-ts014   TS014 switch type for Australia (obsolete)
```

Restrictions for Integrating Data and Voice Services for ISDN PRI Interfaces

- This feature is supported only on C5510 DSP-based platforms.
- ISDN backhaul is not supported.
- This feature does not support modem calls.
- For platforms that support HDLC resources on the motherboard, the available on board HDLC resources are limited to 31 if all resources are not enabled.
- The Cisco 2801 platform does not support full channelized data or full integrated data and voice over T1/E1 PRI interfaces. However, data back up through one PRI channel, or one group of PRI channels for data backup, is supported on this platform.
- Only PPP with multilink is supported for multiple channels. HDLC is not supported for multiple channels.
- You can either configure ds0-groups or pri-groups on one controller, but not both. You receive a message, as in the following example:

```
Router(config-controller)#ds0-group 19 timeslots 20 type e&m-imme$9 timeslots 20 type  
e&m-immediate-start  
%A pri-group was configured already. Please remove it to configure a ds0-group
```

- The following calls are not preempted by a DDR call:
 - Calls from a T.37 store-and-forward off-ramp gateway
 - Incoming ISDN calls
- This feature is not supported from a BRI interface.
- The following dialer commands are not supported with the integrated data and voice feature:
 - **dialer aaa**
 - **dialer callback-secure**
 - **dialer callback-server**
 - **dialer dns**
 - **dialer order**
 - **dialer persistent**
 - **dialer redial**
 - **dialer vpdn**
 - **dialer watch-disable**
 - **dialer watch-group**
 - **dialer watch-list**
 - **dialer watch-list delay**

Information About Integrating Data and Voice Services for ISDN PRI Interfaces

Before you configure integrated data and voice services on ISDN interfaces, you should understand the following concepts:

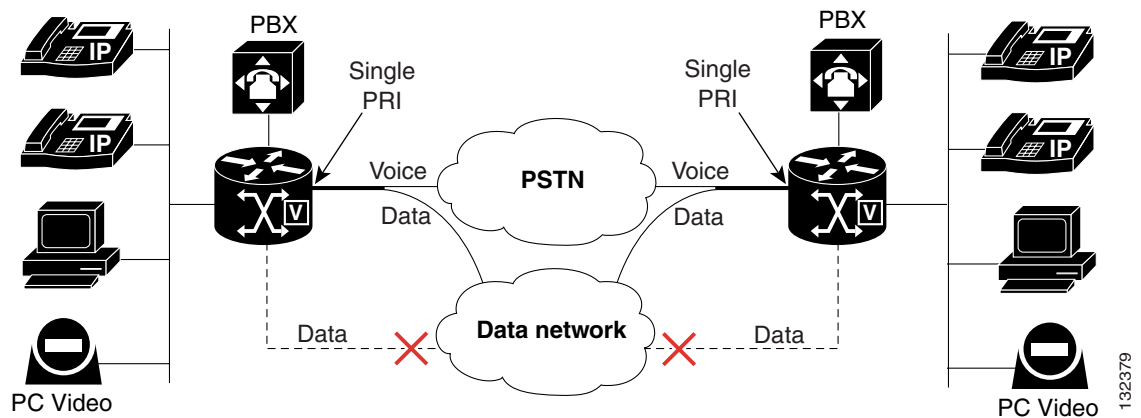
- [Integrated Services for Multiple Call Types](#), page 4
- [Resource Allocation for Voice and Data Calls](#), page 4
- [MLPP Call Preemption over Voice Calls](#), page 5

Integrated Services for Multiple Call Types

ISDN interfaces can support both data calls and voice calls. Typically, this is done using one interface for data and another for voice. This feature enables data (dial-in, dial-on-demand routing [DDR], and DDR backup) and voice call traffic to occur simultaneously from the supported ISDN PRI interfaces. To enable integrated services, the interface used for incoming voice calls is configured to accept multiple voice call types.

Figure 1 shows an ISDN network configured for integrated data and voice services.

Figure 1 *Integrated Voice with DDR Interface for WAN Failure Backup*



Resource Allocation for Voice and Data Calls

Voice calls use DSP resources and data calls use HDLC resources for transmission. When an interface is configured for integrated services, the gateway allocates the HDLC resources dynamically during call setup and frees them back to the HDLC resource pools when the call terminates. This allows spare HDLC resources to support ISDN PRI data calls and DSP resources to support voice calls.

MLPP Call Preemption over Voice Calls

Multilevel precedence and preemption (MLPP) is the placement of priority calls through the network. Precedence designates the priority level that is associated with a call. Preemption designates the process of terminating lower-priority calls so that a call of higher precedence can be extended.

Preemption levels are assigned to outgoing voice calls and DDR backup calls. DDR backup is used to provide backup to a WAN link.

From the gateway, voice and DDR backup calls are controlled by different entities:

- The preemption level of an outgoing voice call is determined using the selected outbound POTS dial peer.
- The preemption level of a DDR backup call is determined using the dialer map class.

A trunk group is used as the common channel resource pool for outgoing voice call and DDR backup calls. Calls with a higher precedence preempt an active outgoing voice call, of a lower precedence, if an idle B channel is not available. An ISDN interface that is configured for integrated mode is assigned to this trunk group to allow dialer resources and voice resources to request an idle B channel from the same resource pool.

Preemption of Outgoing Voice Calls

The trunk group and preemption level are configured as part of a map class, which can be attached to a dialer map. The dialer map class supplies configuration parameters to dialer interfaces and can be referenced from multiple dialer interfaces.

During dial-on-demand routing (DDR) backup call setup, an idle B channel is selected from the trunk group. When no idle channel is found, the trunk group resource manager (TGRM) selects a B channel on the basis of the following:

- The B channel currently active with a connected outgoing voice call
- The preemption level of the connected voice call being lower than the preemption level of a DDR call

A guard timer, configured for the trunk group, is used to delay the idle channel notification and defer the DDR setup to allow the remote channel time to become ready and accept the incoming call with the higher precedence.

By default, the preemption level of dialer calls is set to the lowest level (routine) to disable the MLPP service for a DDR call.

The preemption level of an outgoing voice call is defined from the selected outbound POTS dial peer. During the voice call setup, the trunk group resource manager (TGRM) selects an idle B channel from a trunk group on the basis of the following:

- The call ID of an outgoing voice call
- The preemption level of an outgoing call as defined by the POTS dial peer
- The voice interface B channel information of an outgoing voice call

When the preemption call notification is received, the TGRM saves the outgoing voice call to the preemption level link list based on FIFO.

Preemption Tones

When an outgoing voice call is preempted by a DDR backup call, the preemption call treatment starts by providing a preemption tone and starting the tone timer.

An MLPP preemption tone is a special tone played to the voice call announcing that the line is about to be seized by a call with a higher precedence. A steady tone, 1060 ms in duration, is played on all legs of the call until the user hangs up or the preemption tone times out.

- For the telephony leg of the call, the preemption tone is played using the DSP.
- For the IP leg (across the VoIP network) of the call, the preemption tone is played as media.
- For the ephone leg on Cisco CME, a reorder tone is played for the local user and a preemption tone is played for the remote user.

Preemption Cause Codes

When the preemption tone timer is expired and the call is still in a connected state, both call legs are disconnected by the gateway with the following cause code:

Preemption - Circuit Reserved 0x8

If you release the call before the preemption tone timer expires, the following cause code is used:

Normal Call Clear 0x10

In both cases, the following internal cause code is used for the release calls:

Preemption Circuit Reserved 0x8

How to Configure Integrated Data and Voice Services for ISDN PRI Interfaces

This section describes the tasks required to configure integrated services for ISDN interfaces:

- [Configuring the ISDN PRI Interface for Multiple Call Types, page 6](#) (Required)
- [Configuring MLPP Call Preemption over Outgoing Voice Calls, page 14](#) (Optional)

Configuring the ISDN PRI Interface for Multiple Call Types

An ISDN serial interface configured for integrated mode supports data and voice calls using incoming call type checking to accept incoming voice and data calls when an inbound voice dial peer is matched.

Perform the following tasks to configure integrated services:

- [Prerequisites, page 7](#)
- [Configuring the POTS Dial-Peer Incoming Called Number, page 8](#)
- [Configuring the Data Dial Peer Lookup Preference, page 9](#)
- [Enabling Integrated Services, page 10](#)
- [Creating a Trunkgroup and Configuring Maximum Calls Based on Call Type, page 11](#)
- [Disabling Integrated Services, page 13](#)

Prerequisites

Unlike voice calls, which use DSP resources, data calls use HDLC resources for transmission. To use the integrated services feature, the gateway must allocate HDLC resources dynamically during call setup and free them back to the HDLC resource pools when the call terminates.

Use the following **show** commands to view the availability of HDLC resources:

- **show tdm connections**

The following example shows HDLC resources on the TDM side.

```
Router# show tdm connections slot 0

Active TDM connections for slot 0
=====
(Key: GT=FLEX TDM, V0=VWIC0, V1=VWIC1, V2=VWIC2, V3=VWIC3
    IC=EXPANSION, P0=PVDM0, P1=PVDM1, P2=PVDM2, P3=PVDM3
    HD=HDLC, BP=Backplane (AIM/NM) )

V0:04/04-->HD:31/18, V0:04/06-->HD:31/06, V0:04/08-->HD:31/12
V0:04/10-->HD:31/36, V0:04/12-->HD:31/16, V0:04/14-->HD:31/10
V0:04/16-->HD:31/04, V0:04/18-->HD:31/14, V0:04/20-->HD:31/22
V0:04/22-->HD:31/20, V0:04/24-->HD:31/24, V0:04/26-->HD:31/30
V0:04/28-->HD:31/26, V0:04/30-->HD:31/32, V0:04/32-->HD:31/08
V0:04/34-->HD:31/34, V0:04/36-->HD:31/28, V0:04/38-->HD:31/38
V0:04/64-->HD:31/00, V0:04/66-->HD:31/02, HD:31/00-->V0:04/64
HD:31/02-->V0:04/66, HD:31/04-->V0:04/16, HD:31/06-->V0:04/06
HD:31/08-->V0:04/32, HD:31/10-->V0:04/14, HD:31/12-->V0:04/08
HD:31/14-->V0:04/18, HD:31/16-->V0:04/12, HD:31/18-->V0:04/04
HD:31/20-->V0:04/22, HD:31/22-->V0:04/20, HD:31/24-->V0:04/24
HD:31/26-->V0:04/28, HD:31/28-->V0:04/36, HD:31/30-->V0:04/26
HD:31/32-->V0:04/30, HD:31/34-->V0:04/34, HD:31/36-->V0:04/10
HD:31/38-->V0:04/38,
```

- **show controllers serial [slot/port]**

In the following example, the -1 listings under the `hdlc_chan` column show the free HDLC channels.

```
Router# show controllers Serial 1/1:0

Interface Serial1/1:0
Hardware is HDLC32
HDLC32 resource allocated to this interface:
Slot 1, Vic_slot 1, Port 1
CRC on 1, idle flags 1, frame inverted 0, clocking 0
Channel-group number 0, hdlc32 channel number 2
Channel-group bitfield 0x80000000, hdlc32 quad used 0x4
Channel HW state: 2
TX Ring:
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x0, descriptor: 0x102
data_ptr: 0x2DD1918C, descriptor: 0xB8830102
```

Configuring the POTS Dial-Peer Incoming Called Number

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer data** *tag pots*
4. **incoming called number** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	dial-peer data tag pots Example: Router(config)# dial-peer data 100 pots	Creates a data dial peer and enters data dial-peer configuration mode.
Step 4	incoming called number string Example: Router(config-dial-peer)# incoming called number 4085550110	For data dial-peer matching, only the called number of an incoming call is used to match the incoming called number of POTS dial peers. Wild cards are accepted. Note The <i>string</i> must match the dialer string on the remote gateway.

Configuring the Data Dial Peer Lookup Preference

To optimize data or voice dial-peer searches for incoming ISDN calls, configure the preference of dial-peer lookup during the call type checking. Use the following procedure to configure a search for dial peers by type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer search type {data | none | voice} {data | voice}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	dial-peer search type {data none voice} {data voice} Example: Router(config)# dial-peer search type data voice	Configures the preference of voice or data dial-peer lookup during the calltype checking for incoming ISDN calls. • data—Search dial peers with type data first. • none—Search dial peers with any type at the same preference. • voice—Search dial peers with type voice first. By default, the data dial peer is searched first before voice dial peers.

Enabling Integrated Services

Enabling integrated services allows data and voice call traffic to occur from ISDN PRI interfaces simultaneously.

When an interface is in integrated service mode:

- ISDN performs calltype checking for the incoming call. The call is rejected by ISDN if no voice or data dial peer is matched for an incoming call.
- The **voice** option for the **isdn incoming-voice** command, which treats incoming calls as voice calls, is not available.

By default, the integrated service option is disabled from the supported interfaces. Use the following procedure to enable integrated mode on a serial interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot/port:timeslot*
4. **shutdown**
5. **isdn integrate calltype all**
6. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface serial slot/port:timeslot Example: Router(config)# interface serial 0/1:15	Specifies a serial interface for ISDN PRI channel-associated signaling and enters interface configuration mode.
Step 4	shutdown Example: Router(config-if)# shutdown	Shuts down the interface.
Step 5	isdn integrate calltype all Example: Router(config-if)# isdn integrate calltype all	Enables the serial interface for integrated mode, which allows data and voice call traffic to occur simultaneously. Note This configuration disables the <i>voice</i> option for the isdn incoming-voice command on the interface.
Step 6	no shutdown Example: Router(config-if)# no shutdown	Returns the interface to the active state.

Creating a Trunkgroup and Configuring Maximum Calls Based on Call Type

After an ISDN interface is assigned to a trunk group, you can configure maximum incoming and outgoing calls based on the call type (voice or data) or direction (inbound or outbound) through the trunk group.

**Note**

If trunk groups are not configured, data and voice calls are treated as first-come first-served.

Use the following procedure to create a trunk group and configure maximum calls based on call type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **trunk group** *name*
4. **max-calls** { *any* | *data* | *voice* } *number* [*direction* [*in* | *out*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	trunk group <i>name</i> Example: Router(config)# trunk group 20	Defines a trunk group and enters trunk group configuration mode. <i>name</i>—Name of the trunk group. Valid names contain a maximum of 63 alphanumeric characters.
Step 4	max-calls { any data voice } <i>number</i> [direction [in out]] Example: Router(config-trunk-group)# max-calls data 100 direction out	Defines the maximum number of dial-in or DDR data calls, or voice calls (incoming or outgoing) that can be accepted. any—Assigns the maximum number of calls that the trunk group can handle, regardless of the call type. data—Assigns the maximum number of data calls to the trunk group. voice—Assigns the maximum number of voice calls to the trunk group. <i>number</i>—Specifies number of allowed calls. Range is from 0 to 1000. direction—(Optional) Specifies direction of calls. in—(Optional) Allows only incoming calls. out—(Optional) Allows only outgoing calls.

Examples

See the following sample configurations for the **max-calls** command:

- This example configuration for trunk group 1 accepts up to a maximum of 7 dial-in data or DDR calls and places no restriction on voice calls:

```
trunk group 1
max-calls data 7
```

- This sample configuration for trunk group 2 accepts up to a maximum of 2 data dial-in, 3 DDR calls, and 16 voice calls in any direction:

```
trunk group 2
max-calls data 2 direction in
max-calls data 3 direction out
max-calls voice 16
```

- This sample configuration for trunk group 3 accepts up to a maximum of 10 incoming voice and dial-in data calls.

```
trunk group 3
max-calls any 10 direction in
```

Disabling Integrated Services

When the **isdn integrate calltype all** command is removed from the interface, the **isdn incoming-voice voice** setting is restored and the interface returns to voice mode. Use the following procedure to remove the integrated services option from the interface.

1. **enable**
2. **configure terminal**
3. **interface serial** *slot/port:timeslot*
4. **shutdown**
5. **no isdn integrate calltype all**
6. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface serial <i>slot/port:timeslot</i> Example: Router(config)# interface serial 0/1:15	Specifies a serial interface for ISDN PRI channel-associated signalling and enters interface configuration mode.
Step 4	shutdown Example: Router(config-if)# shutdown	Shuts down the interface.

	Command or Action	Purpose
Step 5	no isdn integrate calltype all	Disables the serial interface from being in integrated mode. You are prompted to confirm this command.
	Example: Router(config-if)# no isdn integrate calltype all	Note This configuration restores the <i>voice</i> option for the isdn incoming-voice command on the interface.
Step 6	no shutdown	Returns the interface to the active state.
	Example: Router(config-if)# no shutdown	

Configuring MLPP Call Preemption over Outgoing Voice Calls

This feature adds support for multilevel precedence and preemption (MLPP) for dial-on-demand routing (DDR) backup calls over outgoing voice calls.

Precedence designates the priority level that is associated with a call. Preemption designates the process of terminating lower-precedence calls so that a call of higher precedence can be extended. DDR backup is used to provide backup to a WAN link using any DDR or a dial-capable interface, like ISDN PRI interfaces.

From the gateway, voice and DDR backup calls are controlled by different entities.

- The preemption level of an outgoing voice call is determined using the selected outbound POTS dial peer.
- The preemption level of a DDR backup call is determined using the dialer map class.

A DDR backup call with higher precedence preempts the active outgoing voice call with a lower precedence if the idle B channel is not available from a trunk group during the DDR backup call setup. If MLPP is not configured, data calls wait for a free channel.

Perform the following tasks to configure call preemption:

- [Enabling Preemption on the Trunk Group, page 14](#)
- [Defining a Dialer Map Class and Setting the Preemption Level, page 16](#)
- [Associating the Class Parameter on the Dialer Interface, page 17](#)
- [Disabling TDM Hairpinning on the Voice Card, page 20](#)
- [Configuring the POTS Dial Peer for Outgoing Voice Calls, page 21](#)
- [Troubleshooting Tips for Integrated Data and Voice Services, page 22](#)

Enabling Preemption on the Trunk Group

A trunk group is used as a common channel resource pool for idle channel allocation for outgoing voice calls and DDR backup calls. Multiple ISDN PRI interfaces that have been configured for integrated services are assigned to this trunk group to build up a channel resource pool for both voice and data calls. Enabling preemption on the trunk group allows DDR call preemption over a voice call per trunk group.



Note

If the trunk group channel resource pool is not shared between voice and DDR calls, you should not enable preemption on the trunk group.

The tone timer defines the expiry timer for the preemption tone for the outgoing voice call, which is being preempted by a DDR backup call. When the tone timer expires, the call is disconnected.

Use the following procedure to create a trunk group resource pool and enable preemption on the trunk group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **trunk group** *name*
4. **preemption enable**
5. **preemption tone timer** *seconds*
6. **preemption guard timer** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	trunk group <i>name</i> Example: Router(config)# trunk group 20	Defines a trunk group and enters trunk group configuration mode. <i>name</i>—Name of the trunk group. Valid names contain a maximum of 63 alphanumeric characters.
Step 4	preemption enable Example: Router(config-trunk-group)# preemption enable	Enables preemption capabilities on a trunk group.

	Command or Action	Purpose
Step 5	preemption tone timer <i>seconds</i> Example: Router(config-trunk-group)# preemption tone timer 20	Defines the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call. <i>seconds</i>—Expiry time, in seconds. The range is 4 to 30. The default value is 10. Note Use the default preemption tone timer command to change back to the default value and no preemption tone timer to disable the tone timer.
Step 6	preemption guard timer <i>value</i> Example: Router(config-trunk-group)# preemption guard timer 60	Defines the guard timer for the DDR call to allow time to clear the last call from the channel. <i>value</i>—Guard timer, in milliseconds. The range is 60 to 500. When preemption is enabled on the trunk group, the default value is 60.

Defining a Dialer Map Class and Setting the Preemption Level

During dial-on-demand routing (DDR) call setup, an idle B channel is selected from the trunk group. The trunk group and preemption level are configured as part of a map class, which can be attached to a dialer map or dialer string. By default, the preemption level of dialer calls is set to the lowest level (routine) to disable the MLPP service for a DDR call.

Use the following procedure to define a map class for the dialer interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **map-class dialer** *class-name*
4. **dialer trunkgroup** *label*
5. **dialer preemption level** {flash-override | flash | immediate | priority | routine}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	map-class dialer <i>class-name</i> Example: Router(config)# map-class dialer dial1	Defines a class of shared configuration parameters associated with the dialer map command for outgoing calls from an ISDN interface. The class name is a unique class identifier. <i>class-name</i>—Unique class identifier.
Step 4	dialer trunkgroup <i>label</i> Example: Router(config-map-class)# dialer trunkgroup 20	Defines the dial-on-demand trunk group label. <i>label</i>—Unique name for the dialer interface trunk group. Valid names contain a maximum of 63 alphanumeric characters.
Step 5	dialer preemption level { flash-override flash immediate priority routine } Example: Router(config-map-class)# dialer preemption level flash	Defines the preemption level of the DDR call on the dialer interface. The default is routine . flash-override—Level 0 (highest) flash—Level 1 immediate—Level 2 priority—Level 3 routine—Level 4 (lowest)

Associating the Class Parameter on the Dialer Interface

The trunk group preemption level is configured as part of a map class, which can be attached to a dialer map or dialer string.

- For legacy DDR, configure the dialer interface to associate the class parameter with the **dialer in-band** and **dialer map** commands.
- For dialer profiles, configure the dialer interface to associate the class parameter with the **dialer pool** and **dialer string** commands.

Use the following procedure to associate the class parameter on the dialer interface.

SUMMARY STEPS

- enable**
- configure terminal**
- interface dialer** *dialer-rotary-group-number*
- dialer in-band** [**no-parity** | **odd-parity**]
or
dialer pool *number*
- dialer map** *protocol-keyword protocol-next-hop-address* [**name** *host-name*] [**speed 56** | **speed 64**] [**broadcast**] **class** *dialer-map-class-name* [*dial-string[:isdn-subaddress]*]
or
dialer string *dial-string* [**class** *class-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface dialer <i>dialer-rotary-group-number</i> Example: Router(config)# interface dialer 10	Defines a dialer rotary group. <i>dialer-rotary-group-number</i>—Number of the dialer rotary group. The range is 0 to 255.
Step 4	dialer in-band [no-parity odd-parity] or dialer pool <i>number</i> Example: Router(config-if)# dialer in-band or Example: Router(config-if)# dialer pool 1	Specifies that dial-on-demand routing (DDR) is to be supported on this interface. no-parity—(Optional) No parity is to be applied to the dialer string that is sent out to the modem on synchronous interfaces. odd-parity—(Optional) Dialed number has odd parity (7-bit ASCII characters with the eighth bit as the parity bit) on synchronous interfaces. or Specifies, for a dialer interface, which dialing pool to use to connect to a specific destination subnetwork. <i>number</i>—The dialing pool number. The range is 1 to 255.

Command or Action	Purpose
Step 5 <pre>dialer map protocol-keyword protocol-next-hop-address [name host-name] [speed 56 speed 64] [broadcast] class dialer-map-class-name [dial-string[:isdn-subaddress]]</pre> or <pre>dialer string dial-string [class class name]</pre> Example: <pre>Router(config-if)# dialer map ip 172.22.82.2 name gw3845 class dial1 20009</pre> or Example: <pre>Router(config-if)# dialer string 4081234 class test</pre>	Configures an ISDN interface to place a call to multiple sites and to authenticate calls from multiple sites. <i>protocol-keyword protocol-next-hop-address</i>—For ISDN services, you must use ip for the protocol-keyword. name host-name—(Optional) The remote system with which the local router or access server communicates. Used for authenticating the remote system on incoming calls. The <i>host-name</i> argument is a case-sensitive name or ID of the remote device. For routers with ISDN interfaces, if calling line identification—sometimes called CLID, but also known as caller ID and automatic number identification (ANI)—is provided, the <i>host-name</i> argument can contain the number that the calling line ID provides. speed 56 speed 64—(Optional) Keyword and value indicating the line speed in kbps to use. Used for ISDN only. The default speed is 64 kbps. broadcast—(Optional) Forwards broadcasts to the address specified with the <i>protocol-next-hop-address</i> argument. class dialer-map-class-name—Dialer map class name. <i>dial-string[:isdn-subaddress]</i>—(Optional) Dial string (telephone number) sent to the dialing device when it recognizes packets with the specified address that matches the configured access lists, and the optional subaddress number used for ISDN multipoint connections. The colon is required for separating numbers. The dial string and ISDN subaddress, when used, must be the last item in the command line. or Specifies the string (telephone number) to be used when placing a call from an interface. <i>dial-string</i>—Telephone number to be sent to a DCE device. class class name—(Optional) Dialer map class associated with this telephone number.

Examples

Legacy DDR Example

```
interface Dialer11
 ip address 172.22.82.1 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 172.22.82.2 name gw3845 class dial1 20009
 dialer load-threshold 1 outbound
 dialer-group 1
```

```

ppp callback accept
ppp authentication chap
ppp multilink

map-class dialer dial1
dialer trunkgroup 1
dialer preemption level flash-override

```

Dialer Profiles Example

```

interface Dialer10
 ip address 192.168.254.1 255.255.255.0
 dialer pool 1
 dialer remote-name is2811
 dialer string 4081234 class test
 dialer-group 1

map-class dialer test
 dialer trunkgroup 1
 dialer preemption level flash-override

```

Disabling TDM Hairpinning on the Voice Card

For TDM-only calls, or for calls that are hairpinned, the preemption tone is not heard as the DSPs are dropped. For this reason, you must disable TDM hairpinning on the voice card to use the MLPP DDR backup call preemption feature.

Use the following procedure to disable TDM hairpinning on the voice card.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice-card** *slot*
4. **no local-bypass**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	voice-card <i>slot</i> Example: Router(config)# voice-card 1	Enters voice-card configuration mode to configure a voice card. <i>slot</i>—Slot number for the card to be configured. Note Valid entries vary by router platform; enter the show voice port summary command for available values.
Step 4	no local-bypass Example: Router(config-voicecard)# no local-bypass	Disables TDM hairpinning.

Configuring the POTS Dial Peer for Outgoing Voice Calls

The preemption level of an outgoing voice call is defined from the outbound POTS dial peer. The preemption level defines the preemption priority level of an outgoing voice call. Use the following procedure to set the preemption level for outgoing voice calls on a POTS dial peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag pots*
4. **trunkgroup** *name* [*preference number*]
5. **preemption level** {*flash-override* | **flash** | **immediate** | **priority** | **routine**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	dial-peer voice <i>tag pots</i> Example: Router(config)# dial-peer voice 25 pots	Defines a particular dial peer, specifies the method of voice encapsulation, and enters dial-peer configuration mode. <i>tag</i>—Digits that define a particular dial peer. The range is from 1 to 2147483647. pots—Indicates that this is a POTS peer that uses VoIP encapsulation on the IP backbone.

	Command or Action	Purpose
Step 4	trunkgroup <i>name</i> [<i>preference-number</i>] Example: Router(config-dial-peer)# trunkgroup 1	Defines the trunk group associated with this dial peer. • <i>name</i>—Label of the trunk group to use for the call. Valid trunk group names contain a maximum of 63 alphanumeric characters. • <i>preference-number</i>—Preference or priority of the trunk group. Range is from 1 (highest priority) to 64 (lowest priority).
Step 5	preemption level { flash-override flash immediate priority routine } Example: Router(config-dial-peer)# preemption level flash	Sets the preemption level of the selected outbound dial peer. Voice calls can be preempted by a DDR call with a higher preemption level. The default is routine . • flash-override—Level 0 (highest) • flash—Level 1 • immediate—Level 2 • priority—Level 3 • routine—Level 4 (lowest) Note The preemption level flash-override setting can prevent the call to be preempted by a DDR call.

Troubleshooting Tips for Integrated Data and Voice Services

ISDN call failures are most commonly attributed to the following issues:

- Dial-on-demand routing (DDR)
- ISDN layers 1, 2 and 3
- Point-to-Point Protocol (PPP): including link control protocol (LCP), Authentication, or IP Control Protocol (IPCP) related issues.

Use the following commands to troubleshoot integrated data and voice for ISDN interfaces:

- **debug dialer events**—Used to display debugging information about the packets received on a dialer interface.
- **debug isdn q931**—Used to check outgoing dial-peer matching for an ISDN incoming call. Enable this command on both sides of the call. The output indicates whether the messages are generated by the calling party router (indicated by TX ->) or by the called party router (indicated by RX <-).
- **debug tgrm inout**—Used to check voice or DDR channel selection request and return status. From the output, you can determine what type of call enabled the preemption and which timeslot is selected from which trunkgroup.
- **debug voip ccapi individual 146**—Used to troubleshoot the call control application programming interface (CCAPI) contents. The **individual 146** command option is used to log call preemption indication information.
- **debug voip ccapi inout**—Used to show how a call flows through the system. From the output, you can see the call setup and teardown operations performed on both the telephony and network call legs.
- **show call history voice | i Cause**—Used to gather DisconnectCause information from the **show call history voice** command line display.

- **show isdn active** and **show isdn status**—Used to show the active data and voice calls.
- **show trunk group**—Used to check the preemption active or pending calls counter for MLPP preemption calls. The output shows the number of active channels from the trunkgroup and the current preemption levels. If a data call with a higher priority initiates the preemption of voice call, it is shown as pending against the higher priority preemption level.

Configuration Examples for Integrating Data and Voice Services for ISDN PRI Interfaces

This section provides the following configuration examples:

- [MLPP DDR Backup Call Preemption over Voice Call: Example, page 23](#)
- [Legacy DDR \(Dialer Map\): Example, page 29](#)
- [Dialer Profiles: Example, page 30](#)
- [Maximum Number of Data and Voice Calls on the Dial-Out Trunk Group: Example, page 32](#)
- [Dial-Peer Configuration: Example, page 35](#)

MLPP DDR Backup Call Preemption over Voice Call: Example

The following example shows that preemption is enabled on the trunk group, the trunk group is associated with a map class, and the preemption level is set on the dialer interface.

Router# **show running-config**

Building configuration...

```
Current configuration : 5984 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
card type e1 0 3
no logging buffered
!
no aaa new-model
!
resource manager
!
network-clock-participate slot 1
network-clock-participate wic 3
ip subnet-zero
!
!
ip cef
no ip dhcp use vrf connected
!
ip dhcp pool ITS
```

```

        network 10.0.0.0 255.255.0.0
        option 150 ip 10.0.0.1
        default-router 10.0.0.1
    !
    !
no ip domain lookup
ip name-server 192.168.2.87
ftp-server enable
no ftp-server write-enable
ftp-server topdir flash:/
isdn switch-type primary-ntt
!
!
trunk group 1
    max-calls data 10 direction out
    preemption enable
    preemption tone 4!

voice-card 0
    dspfarm
    no local-bypass
!
voice-card 1
    dspfarm
    no local-bypass
!
!
voice call send-alert
!
!
!
controller E1 0/3/0
    clock source internal
    pri-group timeslots 1-5,16
    trunk-group 1 timeslots 1-5
!
controller E1 0/3/1
    clock source internal
    pri-group timeslots 1-2,16
    trunk-group 1 timeslots 1-2
!
controller E1 1/0/0
    clock source internal
    pri-group timeslots 1-31
    trunk-group 1 timeslots 1-31
!
controller E1 1/0/1
    clock source internal
    pri-group timeslots 1-10,16
    trunk-group 1 timeslots 1-10
!
!
!
interface Loopback0
    ip address 10.10.1.1 255.255.255.255
!
interface GigabitEthernet0/0
    ip address 10.3.202.87 255.255.0.0
    no ip proxy-arp
    duplex auto
    speed auto
!
interface GigabitEthernet0/1
    ip address 10.0.0.2 255.255.0.0

```



```
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1/0
 switchport access vlan 2
 no ip address
 load-interval 30
 duplex full
 speed 100
!
interface FastEthernet0/1/1
 no ip address
!
interface FastEthernet0/1/2
 no ip address
!
interface FastEthernet0/1/3
 no ip address
!
interface FastEthernet0/1/4
 no ip address
!
interface FastEthernet0/1/5
 no ip address
!
interface FastEthernet0/1/6
 no ip address
!
interface FastEthernet0/1/7
 no ip address
!
interface FastEthernet0/1/8
 no ip address
!
interface Serial0/2/0
 no ip address
 encapsulation frame-relay
 load-interval 30
 shutdown
 no keepalive
 clockrate 2000000
!
interface Serial0/2/0.1 point-to-point
 ip address 10.3.3.1 255.255.255.0
 frame-relay interface-dlci 100
!
interface Serial0/2/1
 no ip address
 shutdown
 clockrate 2000000
!
interface Serial0/3/0:15
 no ip address
 dialer pool-member 1
 isdn switch-type primary-ntt
 isdn protocol-emulate network
 isdn T310 15000
 isdn bchan-number-order descending
 isdn integrate calltype all
 no cdp enable
!
interface Serial0/3/1:15
 no ip address
```

```

dialer pool-member 1
isdn switch-type primary-ntt
isdn protocol-emulate network
isdn T310 15000
isdn bchan-number-order descending
isdn integrate calltype all
no cdp enable
!
interface Serial1/0/0:15
no ip address
dialer pool-member 1
isdn switch-type primary-dms100
isdn protocol-emulate network
isdn T310 15000
isdn bchan-number-order descending
isdn integrate calltype all
no cdp enable
!
interface Serial1/0/1:15
no ip address
encapsulation ppp
dialer pool-member 1
isdn switch-type primary-ntt
isdn protocol-emulate network
isdn T310 15000
isdn bchan-number-order descending
isdn integrate calltype all
ppp multilink
!
interface Vlan1
ip address 10.0.0.1 255.255.0.0
load-interval 30
!
interface Vlan2
ip address 10.7.7.7 255.255.0.0
!
interface Dialer0
ip address 10.5.5.5 255.0.0.0
encapsulation ppp
load-interval 30
dialer pool 1
dialer remote-name Router
dialer string 4081234 class test
dialer load-threshold 10 outbound
dialer-group 1
ppp multilink
ppp multilink load-threshold 5 outbound !
interface Dialer1
ip address 192.168.253.1 255.255.255.0
dialer pool 1
dialer string 4085678 class test
dialer-group 1
!
interface Dialer2
ip address 192.168.252.1 255.255.255.0
dialer pool 1
dialer string 4087777 class test
dialer-group 1
!
ip default-gateway 5.5.5.6
ip classless
ip route 172.16.254.254 255.255.255.255 10.3.0.1 !
ip http server
!

```

```
!  
map-class dialer test  
  dialer trunkgroup 1  
  dialer preemption level flash  
dialer-list 1 protocol ip permit  
snmp-server community public RO  
snmp-server enable traps tty  
!  
!  
!  
control-plane  
!  
!  
!  
voice-port 0/3/0:15  
  echo-cancel enable type hardware  
!  
voice-port 0/3/1:15  
  echo-cancel enable type hardware  
!  
voice-port 1/0/0:15  
  compand-type u-law  
!  
voice-port 1/0/1:15  
!  
voice-port 2/0/0  
  shutdown  
!  
voice-port 2/0/1  
!  
voice-port 2/0/2  
!  
voice-port 2/0/3  
!  
voice-port 2/0/4  
!  
voice-port 2/0/5  
!  
voice-port 2/0/6  
!  
voice-port 2/0/7  
!  
!  
!  
!  
!  
dial-peer voice 100 pots  
  destination-pattern 1...  
  port 2/0/1  
  forward-digits all  
!  
dial-peer voice 2001 pots  
  trunkgroup 1  
  destination-pattern 2...  
  forward-digits all  
!  
dial-peer voice 3001 pots  
  trunkgroup 1  
  destination-pattern 3...  
  forward-digits all  
!  
dial-peer voice 300 pots  
  destination-pattern 4...
```

```

port 2/0/2
forward-digits all
!
dial-peer voice 10 pots
incoming called-number .
direct-inward-dial
forward-digits 0
!
dial-peer voice 5001 pots
trunkgroup 1
destination-pattern 5...
forward-digits all
!
dial-peer voice 500 pots
destination-pattern 6...
port 2/0/3
forward-digits all
!
dial-peer voice 800 pots
trunkgroup 1
destination-pattern 8...
forward-digits all
!
dial-peer data 50 pots
incoming called-number 650T
!
!
!
telephony-service
load 7960-7940 P00303020214
max-ephones 5
max-dn 5
ip source-address 10.0.0.1 port 2000
create cnf-files version-stamp Jan 01 2002 00:00:00 max-conferences 8 gain -6
transfer-system full-consult transfer-pattern .T !
!
ephone-dn 1 dual-line
number 7000
!
!
ephone-dn 2
number 7002
!
!
ephone-dn 3
number 1003
!
!
ephone-dn 4
number 1004
!
!
ephone 1
mac-address 0030.94C2.6073
type 7960
button 1:1
!
!
!
ephone 2
mac-address 000C.851C.ED81
type 7960
button 1:2
!

```

```

!
!
ephone 3
!
!
ephone 4
!
!
alias exec c conf t
alias exec s sh run
!
line con 0
  exec-timeout 0 0
  privilege level 15
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
!
end

```

Legacy DDR (Dialer Map): Example

The following example shows how to associate the class parameter for legacy DDR.

```

Router# show running-config

Building configuration...

Current configuration : 1358 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname host2
!
boot-start-marker
boot-end-marker
!
card type t1 1
!
username client password 0 lab
memory-size iomem 10
no network-clock-participate aim 0
no network-clock-participate aim 1
no aaa new-model
ip subnet-zero
!
ip cef
!
ip ips po max-events 100
no ftp-server write-enable
isdn switch-type primary-ni
!
controller T1 1/0
framing esf
linecode b8zs
cablelength long 0db

```

```

pri-group timeslots 1-24
!
controller T1 1/1
framing sf
linecode ami
cablelength long 0db
!
interface FastEthernet0/0
ip address 10.10.193.77 255.255.0.0
duplex auto
speed auto
!
interface FastEthernet0/1
ip address 192.168.10.1 255.255.255.0
shutdown
duplex auto
speed auto
!
interface Serial1/0:23
ip address 192.168.254.2 255.255.255.0
encapsulation ppp
dialer map ip 172.22.82.2 name gw3845 class dial1 20009
dialer-group 2
isdn switch-type primary-ni
ppp authentication chap
!
no ip classless
ip route 10.10.1.0 255.255.255.0 192.168.254.1
ip route 172.16.254.0 255.255.255.0 10.10.0.1
!
ip http server
no ip http secure-server
!
dialer-list 2 protocol ip permit
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
!
end

```

Dialer Profiles: Example

The following example shows how to associate the class parameter for dialer profiles.

```

Router# show running-config

Building configuration...

Current configuration : 1689 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname host3

```

```

!
boot-start-marker
boot-end-marker
!
card type t1 1
no logging console
!
username uut password 0 lab
no network-clock-participate aim 0
no network-clock-participate aim 1
no aaa new-model
ip subnet-zero
!
ip cef
!
ip ips po max-events 100
no ftp-server write-enable
isdn switch-type primary-ni
!
controller T1 1/0
framing esf
linecode b8zs
cablelength long 0db
pri-group timeslots 1-24
!
controller T1 1/1
framing sf
linecode ami
cablelength long 0db
!
no crypto isakmp enable
!
interface FastEthernet0/0
ip address 10.10.193.88 255.255.0.0
duplex auto
speed auto
!
interface FastEthernet0/1
ip address 10.10.1.1 255.255.255.0
duplex auto
speed auto
!
interface Serial0/3/0
no ip address
clockrate 2000000
!
interface Serial0/3/1
no ip address
clockrate 2000000
!
interface Serial1/0:23
no ip address
encapsulation ppp
dialer pool-member 1
isdn switch-type primary-ni
isdn protocol-emulate network
isdn T310 30000
isdn bchan-number-order descending
ppp authentication chap
!
interface Dialer2
ip address 192.168.252.1 255.255.255.0
dialer pool 1
dialer string 4087777 class test

```

```

dialer-group 1
!
ip default-gateway 5.5.5.6
ip classless
ip route 172.16.254.254 255.255.255.255 10.3.0.1 !
ip http server
!
!
map-class dialer test
dialer trunkgroup 1
dialer preemption level flash
dialer-list 1 protocol ip permit
snmp-server community public RO
snmp-server enable traps tty
!
dialer-list 1 protocol ip permit
!
control-plane
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 8000
end

```

Maximum Number of Data and Voice Calls on the Dial-Out Trunk Group: Example

The following sample configuration shows a maximum number of 500 data and voice calls configured on the trunk group, includes all B channels in the trunk group, and associates dialer *test* with the trunk group.

Router# **show running-config**

Building configuration...

```

Current configuration : 2283 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname host4
!
boot-start-marker
boot-end-marker
!
card type t1 1 1
no logging console
!
no aaa new-model
!
resource manager
!
no network-clock-participate slot 1
ip subnet-zero
!

```



```
ip cef
!
no ftp-server write-enable
isdn switch-type primary-ni
!
trunk group 1
max-calls any 500
!
voice-card 0
dspfarm
!
voice-card 1
dspfarm
!
controller T1 1/0
framing esf
linecode b8zs
!
controller T1 1/0/0
framing esf
linecode b8zs
pri-group timeslots 1-12,24
!
controller T1 1/0/1
framing esf
linecode b8zs
!
interface GigabitEthernet0/0
ip address 10.10.212.212 255.255.0.0
duplex auto
speed auto
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
!
interface Serial1/0/0:23
no ip address
dialer pool-member 1
isdn switch-type primary-ni
isdn protocol-emulate network
isdn T310 30000
isdn bchan-number-order descending
isdn integrate calltype all
trunk-group 1 1
no cdp enable
!
interface Dialer0
ip address 192.168.254.1 255.255.255.0
dialer pool 1
dialer string 4081234 class test
dialer-group 1
!
interface Dialer1
ip address 192.168.253.1 255.255.255.0
dialer pool 1
dialer string 4085678 class test
dialer-group 1
!
interface Dialer2
ip address 192.168.252.1 255.255.255.0
dialer pool 1
dialer string 4087777 class test
```

```

dialer-group 1
!
ip classless
ip route 192.168.10.0 255.255.255.0 Dialer0
ip route 192.168.11.0 255.255.255.0 Dialer1
ip route 192.168.12.0 255.255.255.0 Dialer2
ip route 172.16.254.254 255.255.255.255 GigabitEthernet0/0
!
ip http server
!
map-class dialer test
dialer trunkgroup 1
dialer-list 1 protocol ip permit
!
control-plane
!
voice-port 1/0/0:23
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/0/2
!
voice-port 2/0/3
!
voice-port 2/0/4
!
voice-port 2/0/5
!
voice-port 2/0/6
!
voice-port 2/0/7
!
dial-peer voice 100 pots
destination-pattern 1001
port 2/0/0
forward-digits all
!
dial-peer voice 2001 pots
destination-pattern 200.
port 1/0/0:23
forward-digits all
!
dial-peer voice 101 pots
destination-pattern 1002
port 2/0/1
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
!
end

```

Dial-Peer Configuration: Example

Data dial peers enable the configuration and order assignment of dial peers so that the gateway can identify incoming calls as voice or data. The incoming called number specifies the number associated with the data dial peer. The following example shows a configuration for the voice and data dial-peers and incoming called number.

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

```
Building configuration...
```

```
Current configuration : 1978 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname host6
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource manager
!
no network-clock-participate slot 1
ip subnet-zero
!
ip cef
!
no ftp-server write-enable
isdn switch-type primary-ni
!
trunk group 1
max-calls any 2
!
voice-card 0
dspfarm
!
voice-card 1
dspfarm
!
controller T1 1/1/0
framing esf
linecode b8zs
pri-group timeslots 1-12,24
trunk-group 1 timeslots 2
!
controller T1 1/1/1
framing esf
linecode b8zs
!
interface FastEthernet0/0
ip address 10.10.193.90 255.255.0.0
duplex half
speed 10
!
interface FastEthernet0/1
no ip address
shutdown
```

```

duplex auto
speed auto
!
interface FastEthernet0/1/0
no ip address
shutdown
!
interface FastEthernet0/1/1
no ip address
shutdown
!
interface FastEthernet0/1/2
no ip address
shutdown
!
interface FastEthernet0/1/3
no ip address
shutdown
!
interface Serial1/1/0:23
no ip address
dialer pool-member 2
isdn switch-type primary-ni
isdn integrate calltype all
no cdp enable
!
interface Vlan1
no ip address
!
interface Dialer0
ip address 192.168.254.2 255.255.255.0
dialer pool 2
dialer string 6501234
dialer-group 2
!
ip classless
ip route 10.10.1.0 255.255.255.0 Dialer0
ip route 172.16.254.0 255.255.255.0 10.10.0.1
!
ip http server
!
dialer-list 2 protocol ip permit
!
control-plane
!
voice-port 0/2/0
!
voice-port 0/2/1
!
voice-port 0/2/2
!
voice-port 0/2/3
!
voice-port 1/1/0:23
!
dial-peer voice 100 pots
destination-pattern 2001
port 0/2/0
forward-digits all
!
dial-peer voice 10 pots
incoming called-number .
direct-inward-dial
port 1/1/0:23

```

```

!
dial-peer data 50 pots
incoming called-number 408T
!
dial-peer voice 101 pots
destination-pattern 2002
port 0/2/1
forward-digits all
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
!
end

```

Disconnect Cause: Example

This example shows the DisconnectCause information for a preemption call.

Router# **show call history voice**

```

Telephony call-legs: 2
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 2
GENERIC:
SetupTime=281680 ms
Index=1
PeerAddress=7002
PeerSubAddress=
PeerId=20002
PeerIfIndex=161
LogicalIfIndex=160
DisconnectCause=8
DisconnectText=preemption (8)
ConnectTime=286160 ms
DisconnectTime=441190 ms
CallDuration=00:02:35 sec
CallOrigin=2
ReleaseSource=7
InternalErrorCode=1.1.8.11.35.0
ChargedUnits=0
InfoType=speech
TransmitPackets=0
TransmitBytes=0
ReceivePackets=6910
ReceiveBytes=1105600
TELE:
ConnectionId=[0x4E9D9EF1 0x23E411DA 0x8002A31F 0xB25BECEF]
IncomingConnectionId=[0x4E9D9EF1 0x23E411DA 0x8002A31F 0xB25BECEF]
CallID=1
TxDuration=0 ms
VoiceTxDuration=0 ms
FaxTxDuration=0 ms
CoderTypeRate=g711ulaw
NoiseLevel=0
ACOMLevel=0

```

```

SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=7002
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=7002
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=2000
GwReceivedCallingNumber=7002
GwReceivedCallingOctet3=0x0
GwReceivedCallingOctet3a=0x0 GENERIC:
SetupTime=282800 ms
Index=2
PeerAddress=2000
PeerSubAddress=
PeerId=2001
PeerIfIndex=144
LogicalIfIndex=42
DisconnectCause=8
DisconnectText=preemption (8)
ConnectTime=286160 ms
DisconnectTime=441210 ms
CallDuration=00:02:35 sec
CallOrigin=1
ReleaseSource=7
InternalErrorCode=1.1.8.11.35.0
ChargedUnits=0
InfoType=speech
TransmitPackets=6910
TransmitBytes=1160880
ReceivePackets=6917
ReceiveBytes=1106720
TELE:
ConnectionId=[0x4E9D9EF1 0x23E411DA 0x8002A31F 0xB25BECEF]
IncomingConnectionId=[0x4E9D9EF1 0x23E411DA 0x8002A31F 0xB25BECEF]
CallID=2
TxDuration=0 ms
VoiceTxDuration=0 ms
FaxTxDuration=0 ms
CoderTypeRate=g711ulaw
NoiseLevel=-41
ACOMLevel=26
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
AlertTimepoint=282820 ms
Target tg label=1
OriginalCallingNumber=7002
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=7002

```

```

TranslatedCallingOctet=0x0
TranslatedCalledNumber=2000
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=2000
GwOutpulsedCalledNumber=2000
GwOutpulsedCalledOctet3=0x80
GwReceivedCallingNumber=7002
GwReceivedCallingOctet3=0x0
GwReceivedCallingOctet3a=0x0
GwOutpulsedCallingNumber=7002
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x0
DSPIdentifier=0/1:1

```

Additional References

The following sections provide references related to configuring integrated data and voice for ISDN interfaces.

Related Documents

Related Topic	Document Title
<i>Cisco IOS Voice Configuration Library</i> , including library preface and glossary, other feature documents, and troubleshooting documentation.	<i>Cisco IOS Voice Configuration Library</i>
Voice command reference	<i>Cisco IOS Voice Command Reference</i>
Cisco IOS ISDN voice technologies	<i>Cisco IOS ISDN Voice Configuration Guide</i>
Cisco dial technologies	• <i>Cisco IOS Dial Technologies Configuration Guide</i> • <i>Cisco IOS Dial Technologies Command Reference</i>
ISDN PRI configuration information	<i>Configuring Network Side ISDN PRI Signaling, Trunking, and Switching</i>
Multilevel precedence and preemption (MLPP) information	<i>Multilevel Precedence and Preemption</i>
ISDN voice interface information.	<i>Configuring ISDN PRI Voice-Interface Support</i>

Standards

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

MIBs

MIB	MIBs Link
- CISCO-VOICE-COMMON-DIAL-CONTROL-MIB - CISCO-VOICE-DIAL-CONTROL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

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

Command Reference

This section documents the following new and modified commands:

New Commands

- [dialer preemption level](#)
- [dialer trunkgroup](#)
- [isdn integrate calltype all](#)
- [preemption enable](#)
- [preemption guard timer](#)

- [preemption level](#)
- [preemption tone timer](#)

Modified Commands

- [debug voip ccapi](#)
- [dial-peer data](#)
- [dial-peer search type](#)
- [show trunk group](#)

debug voip ccapi

To troubleshoot the call control application programming interface (CCAPI) contents, use the **debug voip ccapi** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug voip ccapi [**all** | **default** | **detail** | **error** [**call** [**informational**] | **software** [**informational**]] | **individual** *range* | **inout** | **function** | **protoheaders** | **service**]

no debug voip ccapi

Syntax Description	
all	(Optional) Displays all CCAPI debugging messages.
default	(Optional) Displays CCAPI error and inout information. This option also runs if no keywords are added.
detail	(Optional) Displays CCAPI background messages.
error	(Optional) Displays CCAPI error messages. The debug voip ccapi error command traces the error logs in the call control API. Error logs are generated during normal call processing, if there are insufficient resources, or if there are problems in the underlying network-specific code, the higher call session application, or the call control API itself. This debug command shows error events or unexpected behavior in system software. Usually no events will be generated.
call	(Optional) Displays call processing errors.
informational	(Optional) Displays minor errors and major errors. Without the informational keyword, only major errors are displayed.
software	(Optional) Displays software errors.
individual	(Optional) Enables individual CCAPI debug outputs.
<i>range</i>	For the individual keyword, the range is an integer value from 1 to 146. For specific range values, see Table 1 .
inout	(Optional) Displays CCAPI in/out functions. The debug voip ccapi inout 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. You can use the output from this command to understand how calls are being handled by the router. This command shows how a call flows through the system. Using this debug level, you can see the call setup and teardown operations performed on both the telephony and network call legs.
function	(Optional) Displays CCAPI function tracing.
protoheaders	(Optional) Displays CCAPI protocol headers passing information.
service	(Optional) Logs debug messages that are not call related.

Table 1 **CCAPI Individual Debug Values**

Value	CCAPI Debug Function
1	CC_IDMSG_API_DISPLAY_IES
2	CC_IDMSG_SETUP_IND_COMM_2
3	CC_IDMSG_SETUP_IND_COMM_3
4	CC_IDMSG_SETUP_IND_COMM_4
5	CC_IDMSG_ALERT_IND_5
6	CC_IDMSG_ALERT_IND_6
7	CC_IDMSG_CONNECT_IND_7
8	CC_IDMSG_CONNECT_IND_8
9	CC_IDMSG_RECONNECT_IND_9
10	CC_IDMSG_DISCONNECTED_IND_10
11	CC_IDMSG_DISCONNECTED_IND_11
12	CC_IDMSG_DISCONNECTED_IND_12
13	CC_IDMSG_DISCONNECT_DONE_IND_13
14	CC_IDMSG_DISCONNECT_DONE_IND_14
15	CC_IDMSG_DISCONNECT_DONE_IND_15
16	CC_IDMSG_PRE_DISC_CAUSE_16
17	CC_IDMSG_PRE_DISC_CAUSE_17
18	CC_IDMSG_DIGIT_BEGIN_IND_18
19	CC_IDMSG_DIGIT_END_IND_19
20	CC_IDMSG_DIGIT_END_IND_20
21	CC_IDMSG_DIGIT_END_NO_TERM_21
22	CC_IDMSG_TONE_IND_22
23	CC_IDMSG_FEATURE_IND_23
24	CC_IDMSG_MODIFY_DONE_IND_24
25	CC_IDMSG_MODIFY_MODE_DONE_IND_25
26	CC_IDMSG_INBAND_MSG_RCVD_IND_26
27	CC_IDMSG_INBAND_MSG_DONE_IND_27
28	CC_IDMSG_UPD_CALL_INFO_IND_28
29	CC_IDMSG_GEN_NTK_ALERT_EVENT_29
30	CC_IDMSG_VOICE_MODE_EVENT_30
31	CC_IDMSG_VOICE_MODE_EVENT_31
32	CC_IDMSG_DIALING_COMPLETE_IND_32
33	CC_IDMSG_DIGITS_DONE_IND_33
34	CC_IDMSG_DIGITS_DONE_IND_34
35	CC_IDMSG_VBD_XMIT_DONE_IND_35
36	CC_IDMSG_FWD_SETUP_IND_36

Table 1 **CCAPI Individual Debug Values (Continued)**

Value	CCAPI Debug Function
37	CC_IDMSG_RSVP_DONE_IND_37
38	CC_IDMSG_AUDIT_RSP_IND_38
39	CC_IDMSG_XFR_STATUS_IND_39
40	CC_IDMSG_XFR_STATUS_IND_40
41	CC_IDMSG_XFR_DONE_IND_41
42	CC_IDMSG_XFR_DONE_IND_42
43	CC_IDMSG_XFR_DONE_IND_43
44	CC_IDMSG_TGT_CID_ACTIVE_RCD_44
45	CC_IDMSG_MODIFY_MEDIA_IND_45
46	CC_IDMSG_MODIFY_MEDIA_ACK_IND_46
47	CC_IDMSG_MODIFY_MEDIA_REJ_IND_47
48	CC_IDMSG_MODEM_CALL_START_IND_48
49	CC_IDMSG_MODEM_CALL_DONE_IND_49
50	CC_IDMSG_ACCT_STATUS_IND_50
51	CC_IDMSG_NW_STATUS_IND_51
52	CC_IDMSG_DESTINFO_IND_52
53	CC_IDMSG_LOOPBACK_DONE_IND_53
54	CC_IDMSG_RT_PACKET_STATS_IND_54
55	CC_IDMSG_CUT_PROGRESS_IND_55
56	CC_IDMSG_CUT_PROGRESS_IND_56
57	CC_IDMSG_PROCEEDING_IND_57
58	CC_IDMSG_FACILITY_IND_58
59	CC_IDMSG_INFO_IND_59
60	CC_IDMSG_PROGRESS_IND_60
61	CC_IDMSG_USERINFO_IND_61
62	CC_IDMSG_DISC_PROG_IND_62
63	CC_IDMSG_DISC_PROG_IND_63
64	CC_IDMSG_PING_DONE_IND_64
65	CC_IDMSG_COT_TEST_DONE_IND_65
66	CC_IDMSG_PROCESS_DONE_IND_66
67	CC_IDMSG_ASSOCIATED_IND_67
68	CC_IDMSG_SUSPEND_IND_68
69	CC_IDMSG_SUSPEND_ACK_IND_69
70	CC_IDMSG_SUSPEND_REJ_IND_70
71	CC_IDMSG_RESUME_IND_71
72	CC_IDMSG_RESUME_ACK_IND_72

Table 1 **CCAPI Individual Debug Values (Continued)**

Value	CCAPI Debug Function
73	CC_IDMSG_RESUME_REJ_IND_73
74	CC_IDMSG_IF_SETUP_REQ_PRIV_74
75	CC_IDMSG_IF_SETUP_REQ_PRIV_75
76	CC_IDMSG_IF_ALLOCATE_DSP_76
77	CC_IDMSG_CONNECT_77
78	CC_IDMSG_CONNECT_78
79	CC_IDMSG_PING_79
80	CC_IDMSG_DISCONNECT_80
81	CC_IDMSG_DISCONNECT_81
82	CC_IDMSG_DISCONNECT_82
83	CC_IDMSG_ALERT_83
84	CC_IDMSG_ALERT_84
85	CC_IDMSG_CUT_PROGRESS_85
86	CC_IDMSG_CUT_PROGRESS_86
87	CC_IDMSG_CUT_PROGRESS_87
88	CC_IDMSG_DISC_PROG_88
89	CC_IDMSG_DISC_PROG_89
90	CC_IDMSG_SET_PEER_90
91	CC_IDMSG_SET_PEER_91
92	CC_IDMSG_PROCEEDING_92
93	CC_IDMSG_SETUP_REQ_93
94	CC_IDMSG_SETUP_REQ_94
95	CC_IDMSG_SETUP_REQ_95
96	CC_IDMSG_SETUP_REQ_96
97	CC_IDMSG_SETUP_REQ_97
98	CC_IDMSG_SETUP_REQ_98
99	CC_IDMSG_SETUP_REQ_99
100	CC_IDMSG_SETUP_REQ_100
101	CC_IDMSG_SETUP_REQ_101
102	CC_IDMSG_SETUP_ACK_102
103	CC_IDMSG_FACILITY_103
104	CC_IDMSG_TRANSFER_REQ_104
105	CC_IDMSG_GET_CONSULT_ID_105
106	CC_IDMSG_FORWARD_TO_106
107	CC_IDMSG_INFO_107
108	CC_IDMSG_NOTIFY_108

Table 1 **CCAPI Individual Debug Values (Continued)**

Value	CCAPI Debug Function
109	CC_IDMSG_PROGRESS_109
110	CC_IDMSG_PRE_DISC_110
111	CC_IDMSG_PRE_DISC_111
112	CC_IDMSG_USER_INFO_112
113	CC_IDMSG_MODIFY_113
114	CC_IDMSG_DIGIT_114
115	CC_IDMSG_DIGIT_DIAL_115
116	CC_IDMSG_DIGIT_DIAL_STOP_116
117	CC_IDMSG_FEATURE_117
118	CC_IDMSG_FEATURE_ENABLE_118
119	CC_IDMSG_ASSOCIATE_STREAM_119
120	CC_IDMSG_ASSOCIATE_STREAM_120
121	CC_IDMSG_DISASSOCIATE_STREAM_121
122	CC_IDMSG_DISASSOCIATE_STREAM_122
123	CC_IDMSG_GENERATE_TONE_INFO_123
124	CC_IDMSG_SET_DIGIT_TIMEOUTS_124
125	CC_IDMSG_SET_DIGIT_TIMEOUTS_125
126	CC_IDMSG_SUSPEND_126
127	CC_IDMSG_SUSPEND_ACK_127
128	CC_IDMSG_SUSPEND_REJ_128
129	CC_IDMSG_RESUME_129
130	CC_IDMSG_RESUME_ACK_130
131	CC_IDMSG_RESUME_REJ_131
132	CC_IDMSG_UPDATE_REDIRECT_NUM_132
133	CC_IDMSG_BABBLER_AUDIT_133
134	CC_IDMSG_CONFERENCE_CREATE_134
135	CC_IDMSG_CONFERENCE_CREATE_135
136	CC_IDMSG_CONFERENCE_CREATE_136
137	CC_IDMSG_CONFERENCE_DESTROY_137
138	CC_IDMSG_CONFERENCE_DESTROY_138
139	CC_IDMSG_CONFERENCE_DESTROY_139
140	CC_IDMSG_LOOPBACK_140
141	CC_IDMSG_COT_TEST_141
142	CC_IDMSG_HANDOFF_142
143	CC_IDMSG_APP_RETURN_143
144	CC_IDMSG_T38_FAX_START_144

Table 1 CCAPI Individual Debug Values (Continued)

Value	CCAPI Debug Function
145	CC_IDMSG_T38_FAX_DONE_145
146	CC_IDMSG_CALL_PREEMPT_IND_146

Command Default Debugging is not enabled.

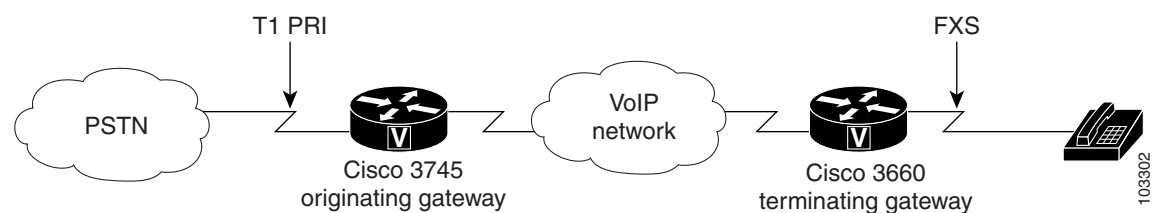
Command Modes Privileged EXEC

Command History	Release	Modification
	11.3(6)NA2	This command was introduced.
	12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.
	12.3(8)T	The all , default , detail , call , informational , software , individual , function , protoheaders , and service keywords were added.
	12.4(4)XC	The range for the individual keyword was extended to 146, to include logs for call preemption indication information.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples The following examples show output for variations of the **debug voip ccapi** command:

- [debug voip ccapi detail on the Originating Gateway: Example, page 48](#)
- [debug voip ccapi detail on the Terminating Gateway: Example, page 49](#)
- [debug voip ccapi inout on the Originating Gateway: Example, page 51](#)
- [debug voip ccapi service on the Terminating Gateway: Example, page 55](#)
- [debug voip ccapi individual 146 on the Originating Gateway: Example, page 55](#)

For these examples, the topology shown in [Figure 2](#) is used.

Figure 2 Network Topology for debug voip ccapi Output Examples

debug voip ccapi detail on the Originating Gateway: Example

Router# **debug voip ccapi detail**

voip ccapi detail debugging is on

Router#

```
*Apr 18 20:35:35.779: //-1/ABCE697D8005/CCAPI/cc_api_call_setup_ind_common:
  Interface Type=13, Protocol=0
*Apr 18 20:35:35.779: //-1/ABCE697D8005/CCAPI/ccCheckClipClir:
  Calling Party Number Is User Provided
*Apr 18 20:35:35.779: //11/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=0, Call Entry(Call Count On=FALSE, Incoming Call=TRUE)
*Apr 18 20:35:35.779: //11/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=1
```

The following event shows that the CallEntry ID 11 is used for the incoming call leg.

```
*Apr 18 20:35:35.779: //11/ABCE697D8005/CCAPI/cc_insert_guid_pod_entry:
  Incoming=TRUE, Call Id=11
*Apr 18 20:35:35.779: //11/ABCE697D8005/CCAPI/cc_setupind_registration_lookup:
  Matching Parameters; Called Number=83103, Call Transfer Consult Id=
*Apr 18 20:35:35.779: //11/ABCE697D8005/CCAPI/cc_setupind_registration_lookup:
  No Matching Node
*Apr 18 20:35:35.779: //11/ABCE697D8005/CCAPI/ccCheckClipClir:
  Calling Party Number Is User Provided
*Apr 18 20:35:35.779: //12/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=1, Call Entry(Call Count On=FALSE, Incoming Call=FALSE)
```

The following event shows that the incoming call leg with CallEntry ID 11 is bound to the outgoing call leg with CallEntry ID 12.

```
*Apr 18 20:35:35.779: //11/ABCE697D8005/CCAPI/cc_peer_bind:
  Bind=TRUE, Binder Call Id=11, Bindee Call Id=12
```

The next event shows that CallEntry ID 12 is used for the outgoing call leg.

```
*Apr 18 20:35:35.779: //12/ABCE697D8005/CCAPI/cc_insert_guid_pod_entry:
  Incoming=FALSE, Call Id=12
*Apr 18 20:35:35.779: //-1/xxxxxxxxxxxx/CCAPI/cc_api_supported_data:
  data_mode=0x10082
```

The next event shows an IP address for a remote device on the outgoing call leg, which indicates that this is the VoIP call leg.

```
*Apr 18 20:35:35.779: //12/ABCE697D8005/CCAPI/cc_incr_if_call_volume:
  Remote IP Address=172.16.13.81, HwIdb=FastEthernet0/0
*Apr 18 20:35:35.779: //12/ABCE697D8005/CCAPI/cc_incr_if_call_volume:
  Total Call Count=1, Voip Call Count=1, MMoip Call Count=0
*Apr 18 20:35:35.795: //11/ABCE697D8005/CCAPI/ccCallGetContext:
  Context=0x652C0168, Call Id=11
*Apr 18 20:36:31.419: //11/ABCE697D8005/CCAPI/ccCallDisconnect:
  Start Calling Accounting;
  Call Entry(Incoming=TRUE)
*Apr 18 20:36:31.419: //11/ABCE697D8005/CCAPI/ccCallDisconnect:
  Cause Value=16, Call Entry(Disconnect Cause=16)
*Apr 18 20:36:31.419: //11/ABCE697D8005/CCAPI/ccCallDisconnect:
  Call Entry(Disconnect Cause=16)
```


At this point, the CallEntry ID changes as the call accounting process begins. The accounting data is sent over the outgoing call leg. The GUID, which identifies the unique call, remains the same.

```
*Apr 18 20:36:31.419: //12/ABCE697D8005/CCAPI/ccCallDisconnect:
  Start Calling Accounting;
  Call Entry(Incoming=FALSE)
*Apr 18 20:36:31.419: //12/ABCE697D8005/CCAPI/ccCallDisconnect:
  Cause Value=16, Call Entry(Disconnect Cause=0)
```

The change of the CallEntry ID indicates that the call is using the incoming call leg, which is the POTS call leg in this case.

```
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Data Bitmask=0x1, Call Id=11
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Flag=FALSE
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/cc_delete_guid_pod_entry:
  Incoming=TRUE
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/cc_delete_call_entry:
  ccFreeRawMsgInfo=0x63FF8198
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/cc_delete_call_entry:
  Total Call Count=1, Call Entry(Call Count On=FALSE, Incoming Call=TRUE)
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/cc_delete_call_entry:
  Total Call Count=0
*Apr 18 20:36:31.423: //11/ABCE697D8005/CCAPI/cc_delete_call_entry:
  Deleting profileTable[0x64F44700]
```

The next line shows the impairment calculation. This is the only CCAPI debug command that shows impairment.

```
*Apr 18 20:36:31.423: //-1/ABCE697D8005/CCAPI/g113_calculate_impairment:
  (delay=91(ms), loss=0%), Io=0 Iq=0 Idte=0 Idd=2 Ie=10 Itot=12
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Data Bitmask=0x1, Call Id=12
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Flag=FALSE
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/cc_decr_if_call_volume:
  Remote IP Address=172.16.13.81, Hwldb=FastEthernet0/0
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/cc_decr_if_call_volume:
  Total Call Count=0, Voip Call Count=0, MMoip Call Count=0
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/cc_delete_guid_pod_entry:
  Incoming=FALSE
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/cc_delete_call_entry:
  Total Call Count=0, Call Entry(Call Count On=FALSE, Incoming Call=FALSE)
*Apr 18 20:36:31.423: //12/ABCE697D8005/CCAPI/cc_delete_call_entry:
  Deleting profileTable[0x652E3310]
*Apr 18 20:36:31.427: //12/xxxxxxxxxxxx/CCAPI/cc_get_call_entry:
  Call Entry Is Not Found
```

debug voip ccapi detail on the Terminating Gateway: Example

Router# **debug voip ccapi detail**

```
voip ccapi detail debugging is on
Router#
*May 1 18:58:26.251: //-1/xxxxxxxxxxxx/CCAPI/cc_api_supported_data:
  data_mode=0x10082
*May 1 18:58:26.255: //8/xxxxxxxxxxxx/CCAPI/cc_get_call_entry:
  Call Entry Is Not Found
*May 1 18:58:26.255: //-1/ABCE697D8005/CCAPI/cc_api_call_setup_ind_common:
  Interface Type=0, Protocol=1
*May 1 18:58:26.255: //-1/ABCE697D8005/CCAPI/ccCheckClipClir:
  Calling Party Number Is User Provided
```

The following line shows the attributes of the calling number:

```
*May 1 18:58:26.255: //-1/ABCE697D8005/CCAPI/cc_api_call_setup_ind_common:
  After Number Translation Checking:
    Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed),
    Called Number=3600(TON=Unknown, NPI=Unknown)
*May 1 18:58:26.255: //8/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=0, Call Entry(Call Count On=FALSE, Incoming Call=TRUE)
*May 1 18:58:26.255: //8/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=1
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/cc_insert_guid_pod_entry:
  Incoming=TRUE, Call Id=8
```

The following line shows the IP address of the originating gateway:

```
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/cc_incr_if_call_volume:
  Remote IP Address=172.16.13.175, HwIdb=FastEthernet0/0
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/cc_incr_if_call_volume:
  Total Call Count=1, Voip Call Count=1, MMoip Call Count=0
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/cc_setupind_registration_lookup:
  Matching Parameters; Called Number=3600, Call Transfer Consult Id=
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/cc_setupind_registration_lookup:
  No Matching Node
*May 1 18:58:26.255: //8/ABCE697D8005/CCAPI/ccCheckClipClir:
  Calling Party Number Is User Provided
*May 1 18:58:26.259: //9/xxxxxxxxxxxx/CCAPI/cc_insert_call_entry:
  Total Call Count=1, Call Entry(Call Count On=FALSE, Incoming Call=FALSE)
*May 1 18:58:26.259: //8/ABCE697D8005/CCAPI/cc_peer_bind:
  Bind=TRUE, Binder Call Id=8, Bindee Call Id=9
*May 1 18:58:26.259: //9/ABCE697D8005/CCAPI/cc_insert_guid_pod_entry:
  Incoming=FALSE, Call Id=9
*May 1 18:58:26.259: //9/ABCE697D8005/CCAPI/cc_set_voice_port_value:
  CC_IF_TELEPHONY: Echo=0, Playout=0
*May 1 18:58:26.263: //9/ABCE697D8005/CCAPI/ccCallGetContext:
  Context=0x64B6BB5C, Call Id=9
*May 1 18:59:21.871: //8/ABCE697D8005/CCAPI/ccCallDisconnect:
  Start Calling Accounting;
  Call Entry(Incoming=TRUE)
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/ccCallDisconnect:
  Cause Value=16, Call Entry(Disconnect Cause=16)
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/ccCallDisconnect:
  Call Entry(Disconnect Cause=16)
*May 1 18:59:21.875: //9/ABCE697D8005/CCAPI/ccCallDisconnect:
  Start Calling Accounting;
  Call Entry(Incoming=FALSE)
*May 1 18:59:21.875: //9/ABCE697D8005/CCAPI/ccCallDisconnect:
  Cause Value=16, Call Entry(Disconnect Cause=0)
```

The next line shows the impairment calculation. This is the only CCAPI debug command that shows impairment.

```
*May 1 18:59:21.875: //-1/ABCE697D8005/CCAPI/g113_calculate_impairment:
  (delay=99(ms), loss=0%), Io=0 Iq=0 Idte=0 Idd=2 Ie=10 Itot=12
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Data Bitmask=0x1, Call Id=8
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
  Flag=FALSE
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_decr_if_call_volume:
  Remote IP Address=172.16.13.175, HwIdb=FastEthernet0/0
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_decr_if_call_volume:
  Total Call Count=0, Voip Call Count=0, MMoip Call Count=0
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_delete_guid_pod_entry:
```

```

Incoming=TRUE
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_delete_call_entry:
ccFreeRawMsgInfo=0x644EB850
Router#
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_delete_call_entry:
Total Call Count=1, Call Entry(Call Count On=FALSE, Incoming Call=TRUE)
*May 1 18:59:21.875: //8/ABCE697D8005/CCAPI/cc_delete_call_entry:
Total Call Count=0
*May 1 18:59:21.879: //8/ABCE697D8005/CCAPI/cc_delete_call_entry:
Deleting profileTable[0x64B78600]
*May 1 18:59:21.879: //8/xxxxxxxxxxxx/CCAPI/cc_get_call_entry:
Call Entry Is Not Found
*May 1 18:59:21.879: //8/xxxxxxxxxxxx/CCAPI/cc_get_call_entry:
Call Entry Is Not Found
Router#
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
Data Bitmask=0x1, Call Id=9
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/ccCallGetVoipFlag:
Flag=FALSE
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/cc_api_call_disconnect_done:
Prefix Is Not Defined From Peer; Peer=3600, Called Number=3600
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/cc_delete_guid_pod_entry:
Incoming=FALSE
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/cc_delete_call_entry:
Total Call Count=0, Call Entry(Call Count On=FALSE, Incoming Call=FALSE)
*May 1 18:59:24.587: //9/ABCE697D8005/CCAPI/cc_delete_call_entry:
Deleting profileTable[0x6453F228]

```

debug voip ccapi inout on the Originating Gateway: Example

Router# **debug voip ccapi inout**

```

voip ccapi inout debugging is on
Router#
*Apr 18 20:42:19.347: //-1/9C5A9CA88009/CCAPI/cc_api_display_ie_subfields:
cc_api_call_setup_ind_common:
acme-username=
----- ccCallInfo IE subfields -----
acme-ani=4085550111
acme-anitype=2
acme-aniplan=1
acme-anipi=0
acme-anisi=1
dest=83103
acme-desttype=0
acme-destplan=0
acme-rdn=
acme-rdntype=-1
acme-rdnplan=-1
acme-rdnpi=-1
acme-rdnsi=-1
acme-redirectreason=-1

```

The following lines show information about the calling and called numbers. The network presentation indicator (NPI) shows the type of transmission. The Incoming Dial-Peer field shows that the incoming dial peer has been matched.

```

*Apr 18 20:42:19.347: //-1/9C5A9CA88009/CCAPI/cc_api_call_setup_ind_common:
Interface=0x64F26F10, Call Info(
Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed),
Called Number=83103(TON=Unknown, NPI=Unknown),
Calling Translated=FALSE, Subscriber Type Str=RegularLine, FinalDestinationFlag=TRUE,

```

```

Incoming Dial-peer=1, Progress Indication=NULL(0), Calling IE Present=TRUE,
Source Trkgrp Route Label=, Target Trkgrp Route Label=, CLID Transparent=FALSE), Call
Id=-1
*Apr 18 20:42:19.347: //-1/9C5A9CA88009/CCAPI/ccCheckClipClir:
  In: Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed)
*Apr 18 20:42:19.347: //-1/9C5A9CA88009/CCAPI/ccCheckClipClir:
  Out: Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed)

```

In the following event, the call leg is established. The CallEntry ID field changes from -1 to 19.

```

*Apr 18 20:42:19.347: //19/9C5A9CA88009/CCAPI/cc_api_call_setup_ind_common:
  Set Up Event Sent;
  Call Info(Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed),
  Called Number=83103(TON=Unknown, NPI=Unknown))
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/cc_process_call_setup_ind:
  Event=0x63FF4730
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCallSetContext:
  Context=0x652A9858
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/cc_process_call_setup_ind:
  >>>CCAPI handed cid 19 with tag 1 to app "Default"
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCallProceeding:
  Progress Indication=NULL(0)
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCallSetupRequest:
  Destination=, Calling IE Present=TRUE, Mode=0,
  Outgoing Dial-peer=3600, Params=0x652AA4A8, Progress Indication=NULL(0)
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCheckClipClir:
  In: Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed)
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCheckClipClir:
  Out: Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed)
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCallSetupRequest:
  Destination Pattern=360., Called Number=3600, Digit Strip=FALSE
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccCallSetupRequest:
  Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed),
  Called Number=3600(TON=Unknown, NPI=Unknown),
  Redirect Number=, Display Info=
  Account Number=, Final Destination Flag=TRUE,
  Guid=9C5A9CA8-5243-11D6-8009-00059A3A15A0, Outgoing Dial-peer=3600
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/cc_api_display_ie_subfields:
  ccCallSetupRequest:
  cisco-username=
  ----- ccCallInfo IE subfields -----
  cisco-ani=4085550111
  cisco-anitype=2
  cisco-aniplan=1
  cisco-anipi=0
  cisco-anisi=1
  dest=3600
  cisco-desttype=0
  cisco-destplan=0
  cisco-rdn=
  cisco-rdntype=-1
  cisco-rdnplan=-1
  cisco-rdnpi=-1
  cisco-rdnsi=-1
  cisco-redirectreason=-1

```

In the following lines, the outgoing dial peer is matched:

```
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccIFCallSetupRequestPrivate:
  Interface=0x63EAF24C, Interface Type=1, Destination=, Mode=0x0,
  Call Params(Calling Number=4085550111(TON=National, NPI=ISDN, Screening=User, Passed,
Presentation=Allowed),
  Called Number=3600(TON=Unknown, NPI=Unknown), Calling Translated=FALSE,
  Subscriber Type Str=RegularLine, FinalDestinationFlag=TRUE, Outgoing Dial-peer=3600,
Call Count On=FALSE,
  Source Trkgrp Route Label=, Target Trkgrp Route Label=, tg_label_flag=0, Application
Call Id=)
*Apr 18 20:42:19.351: //20/9C5A9CA88009/CCAPI/ccIFCallSetupRequestPrivate:
  SPI Call Setup Request Is Success; Interface Type=1, FlowMode=1
*Apr 18 20:42:19.351: //20/9C5A9CA88009/CCAPI/ccCallSetContext:
  Context=0x652AA458
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccSaveDialpeerTag:
  Outgoing Dial-peer=3600
*Apr 18 20:42:19.351: //19/9C5A9CA88009/CCAPI/ccSaveDialpeerTag:
  Outgoing Dial-peer=3600
*Apr 18 20:42:19.367: //20/9C5A9CA88009/CCAPI/cc_api_call_proceeding:
  Interface=0x652F6388, Progress Indication=NULL(0)
```

The following lines show call progress. The progress and signal indications are shown.

```
*Apr 18 20:42:19.371: //20/9C5A9CA88009/CCAPI/cc_api_call_cut_progress:
  Interface=0x652F6388, Progress Indication=INBAND(8), Signal Indication=SIGNAL
RINGBACK(1),
  Cause Value=0
*Apr 18 20:42:19.371: //20/9C5A9CA88009/CCAPI/cc_api_call_cut_progress:
  Call Entry(Responded=TRUE)
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/ccCallCutProgress:
  Progress Indication=INBAND(8), Signal Indication=SIGNAL RINGBACK(1), Cause Value=0
  Voice Call Send Alert=FALSE, Call Entry(AlertSent=FALSE)
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/ccCallCutProgress:
  Call Entry(Responded=TRUE)
```

The following lines show the tone generation information:

```
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/ccGenerateToneInfo:
  Stop Tone On Digit=FALSE, Tone=NULL,
  Tone Direction=Network, Params=0x0, Call Id=19
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/ccConferenceCreate:
  Conference Id=0x652F723C, Call Id1=19, Call Id2=20, Tag=0x0
*Apr 18 20:42:19.371: //20/xxxxxxxxxxxx/CCAPI/cc_api_bridge_done:
  Conference Id=0x6, Source Interface=0x63EAF24C, Source Call Id=20,
  Destination Call Id=19, Disposition=0x0, Tag=0x0
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/ccConferenceCreate:
  Call Entry(Conference Id=0x6, Destination Call Id=20)
*Apr 18 20:42:19.371: //20/9C5A9CA88009/CCAPI/ccConferenceCreate:
  Call Entry(Conference Id=0x6, Destination Call Id=19)
*Apr 18 20:42:19.371: //19/xxxxxxxxxxxx/CCAPI/cc_api_bridge_done:
  Conference Id=0x6, Source Interface=0x64F26F10, Source Call Id=19,
  Destination Call Id=20, Disposition=0x0, Tag=0x0
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/cc_generic_bridge_done:
  Conference Id=0x6, Source Interface=0x64F26F10, Source Call Id=19,
  Destination Call Id=20, Disposition=0x0, Tag=0x0
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/cc_api_caps_ind:
  Destination Interface=0x63EAF24C, Destination Call Id=20, Source Call Id=19,
  Caps(Codec=0x2887F, Fax Rate=0xBF, Vad=0x3,
  Modem=0x2, Codec Bytes=0, Signal Type=3)
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/cc_api_caps_ind:
  Caps(Playout Mode=1, Playout Initial=60(ms), Playout Min=40(ms),
  Playout Max=300(ms), Fax Nom=300(ms))
*Apr 18 20:42:19.371: //19/9C5A9CA88009/CCAPI/cc_process_notify_bridge_done:
  Conference Id=0x6, Call Id1=19, Call Id2=20
```

```
*Apr 18 20:42:19.375: //20/9C5A9CA88009/CCAPI/cc_api_caps_ind:
  Destination Interface=0x64F26F10, Destination Call Id=19, Source Call Id=20,
  Caps(Codec=0x4, Fax Rate=0x1, Vad=0x2,
  Modem=0x2, Codec Bytes=20, Signal Type=2)
*Apr 18 20:42:19.375: //20/9C5A9CA88009/CCAPI/cc_api_caps_ind:
  Caps(Playout Mode=1, Playout Initial=60(ms), Playout Min=40(ms),
  Playout Max=300(ms), Fax Nom=300(ms))
```

The following lines show codec information:

```
*Apr 18 20:42:19.375: //20/9C5A9CA88009/CCAPI/cc_api_caps_ack:
  Destination Interface=0x64F26F10, Destination Call Id=19, Source Call Id=20,
  Caps(Codec=g729r8(0x4), Fax Rate=FAX_RATE_NONE(0x1), Vad=ON(0x2),
  Modem=ON(0x2), Codec Bytes=20, Signal Type=2, Seq Num Start=6872)
*Apr 18 20:42:19.375: //19/9C5A9CA88009/CCAPI/cc_api_caps_ack:
  Destination Interface=0x63EAF24C, Destination Call Id=20, Source Call Id=19,
  Caps(Codec=g729r8(0x4), Fax Rate=FAX_RATE_NONE(0x1), Vad=ON(0x2),
  Modem=ON(0x2), Codec Bytes=20, Signal Type=2, Seq Num Start=6872)
*Apr 18 20:42:19.375: //19/9C5A9CA88009/CCAPI/cc_api_voice_mode_event:
  Call Id=19
*Apr 18 20:42:19.375: //19/9C5A9CA88009/CCAPI/cc_api_voice_mode_event:
  Call Entry(Context=0x652A9858)
```

The following lines show progress indication information. In this case, the event shows that the destination is not ISDN.

```
*Apr 18 20:42:26.855: //20/9C5A9CA88009/CCAPI/cc_api_call_connected:
  Interface=0x652F6388, Data Bitmask=0x0, Progress Indication=DESTINATION IS NON ISDN(2),
  Connection Handle=0
*Apr 18 20:42:26.855: //20/9C5A9CA88009/CCAPI/cc_api_call_connected:
  Call Entry(Connected=TRUE, Responded=TRUE, Retry Count=0)
*Apr 18 20:42:26.855: //19/9C5A9CA88009/CCAPI/ccCallConnect:
  Progress Indication=DESTINATION IS NON ISDN(2), Data Bitmask=0x0
*Apr 18 20:42:26.855: //19/9C5A9CA88009/CCAPI/ccCallConnect:
  Call Entry(Connected=TRUE, Responded=TRUE)
*Apr 18 20:42:26.855: //20/9C5A9CA88009/CCAPI/ccSaveDialpeerTag:
  Incoming Dial-peer=1
*Apr 18 20:42:26.859: //19/9C5A9CA88009/CCAPI/ccSaveDialpeerTag:
  Outgoing Dial-peer=3600
*Apr 18 20:42:26.859: //20/9C5A9CA88009/CCAPI/ccCallFeature:
  Feature Type=24, Call Id=20
```

This event shows that the call is disconnected.

```
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/cc_api_call_disconnected:
  Cause Value=16, Interface=0x64F26F10, Call Id=19
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/cc_api_call_disconnected:
  Call Entry(Responded=TRUE, Cause Value=16, Retry Count=0)
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/ccConferenceDestroy:
  Conference Id=0x6, Tag=0x0
*Apr 18 20:43:16.795: //20/xxxxxxxxxxxx/CCAPI/cc_api_bridge_drop_done:
  Conference Id=0x6, Source Interface=0x63EAF24C, Source Call Id=20,
  Destination Call Id=19, Disposition=0x0, Tag=0x0
*Apr 18 20:43:16.795: //19/xxxxxxxxxxxx/CCAPI/cc_api_bridge_drop_done:
  Conference Id=0x6, Source Interface=0x64F26F10, Source Call Id=19,
  Destination Call Id=20, Disposition=0x0, Tag=0x0
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/cc_generic_bridge_done:
  Conference Id=0x6, Source Interface=0x64F26F10, Source Call Id=19,
  Destination Call Id=20, Disposition=0x0, Tag=0x0
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/ccCallDisconnect:
  Cause Value=16, Tag=0x0, Call Entry(Previous Disconnect Cause=0, Disconnect Cause=16)
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/ccCallDisconnect:
  Cause Value=16, Call Entry(Responded=TRUE, Cause Value=16)
*Apr 18 20:43:16.795: //19/9C5A9CA88009/CCAPI/cc_api_get_transfer_info:
```

```

Transfer Number Is Null
*Apr 18 20:43:16.795: //20/9C5A9CA88009/CCAPI/ccCallDisconnect:
Cause Value=16, Tag=0x0, Call Entry(Previous Disconnect Cause=0, Disconnect Cause=0)
*Apr 18 20:43:16.795: //20/9C5A9CA88009/CCAPI/ccCallDisconnect:
Cause Value=16, Call Entry(Responded=TRUE, Cause Value=16)
*Apr 18 20:43:16.795: //20/9C5A9CA88009/CCAPI/cc_api_get_transfer_info:
Transfer Number Is Null
*Apr 18 20:43:16.803: //20/9C5A9CA88009/CCAPI/cc_api_call_disconnect_done:
Disposition=0, Interface=0x652F6388, Tag=0x0, Call Id=20,
Call Entry(Disconnect Cause=16, Voice Class Cause Code=0, Retry Count=0)
*Apr 18 20:43:16.803: //20/9C5A9CA88009/CCAPI/cc_api_call_disconnect_done:
Call Disconnect Event Sent
*Apr 18 20:43:16.803: //19/9C5A9CA88009/CCAPI/cc_api_call_disconnect_done:
Disposition=0, Interface=0x64F26F10, Tag=0x0, Call Id=19,
Call Entry(Disconnect Cause=16, Voice Class Cause Code=0, Retry Count=0)
*Apr 18 20:43:16.803: //19/9C5A9CA88009/CCAPI/cc_api_call_disconnect_done:
Call Disconnect Event Sent

```

debug voip ccapi service on the Terminating Gateway: Example

Router# **debug voip ccapi service**

```

voip ccapi service debugging is on
*May 1 19:08:41.803: //-1/xxxxxxxxxxxx/CCAPI/cc_setupind_match_search:
Searching Node;
Called Number=3600, Call Transfer Consult Id=

```

This debug shows noncall related events. In this case, information about the timer is shown.

```

*May 1 19:08:48.027: //-1/xxxxxxxxxxxx/CCAPI/cc_handle_periodic_timer:
Calling The Callback, ccTimerctx=0x63B368C0
*May 1 19:08:48.027: //-1/xxxxxxxxxxxx/CCAPI/ccTimerStart:
ccTimerctx=0x63B368C0
*May 1 19:10:08.615: //-1/xxxxxxxxxxxx/CCAPI/cc_api_icpif:
ExpectFactor=0xA

```

debug voip ccapi individual 146 on the Originating Gateway: Example

This debug shows the preemption tone timer getting initiated.

```

*Aug 24 18:28:16.919: //18958/B37648B6AF48/CCAPI/cc_api_call_preempt_ind:

PreemptionToneTimer=10(s)

```

dial-peer data

To create a data dial peer and to enter dial-peer configuration mode, use the **dial-peer data** command in global configuration mode. To remove a data dial peer, use the **no** form of this command.

dial-peer data *tag* **pots**

no dial-peer data *tag*

Syntax Description

<i>tag</i>	Specifies the dial-peer identifying number. Range is from 1 to 2147483647.
pots	Specifies an incoming POTS dial peer.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.4(4)XC	This command was implemented on the Cisco 2600XM series, Cisco 2800 series, Cisco 3700 series, and Cisco 3800 series.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

A data dial peer should be defined only for incoming data calls. The **incoming called-number** and **shutdown** commands on the data dial peer are allowed. However, the following POTS dial-peer commands are disabled on a data dial peer:

- **answer-address**
- **carrier-id**
- **destination-pattern**
- **information-type**
- **port**
- **trunk-group-label**

Examples

The following example is a data dial-peer configuration:

```
dial-peer data 100 pots
  incoming called-number 100
```


The following example is a voice dial-peer configuration:

```
dial-peer voice 2001 pots
destination-pattern 2001
no digit-strip
port 3/1:1
```

Related Commands

Command	Description
dial-peer search	Optimizes voice or data dial-peer searches.
incoming called-number	Specifies an incoming called number of an MMoIP or POTS dial peer.
shutdown (dial peer)	Changes the administrative state of a selected dial peer from up to down.

dial-peer search type

To optimize voice or data dial-peer searches, use the **dial-peer search type** command in global configuration mode. To disable the search parameters, use the **no** form of this command.

dial-peer search type { **data voice** | **voice data** | **none** }

no dial-peer search type

Syntax Description	data	Searches for data dial peers.
	none	Searches for all dial peers by order of input.
	voice	Searches for voice dial peers.

Command Default	data and voice
------------------------	------------------------------

Command Modes	Global configuration
----------------------	----------------------

Command History	Release	Modification
	12.2(13)T	This command was introduced.
	12.4(4)XC	This command was implemented on the Cisco 2600XM series, Cisco 2800 series, Cisco 3700 series, and Cisco 3800 series.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines	The search defines the search preference explicitly. If the data and voice keywords are specified, data dial peers are searched first. If no data dial peers are found, the voice dial peers are searched.
-------------------------	--

Examples	The following is sample output that shows that data dial peers are searched first. Then voice dial peers are searched if no data dial peers can be matched for an incoming call:
-----------------	--

```
dial-peer search type data voice
```

The following is sample output that shows that voice dial peers are searched first. Then data dial peers are searched if no voice dial peers can be matched for an incoming call:

```
dial-peer search type voice data
```

Related Commands	Command	Description
	dial-peer data	Enable a gateway to process incoming data calls first by assigning the POTS dial peer as data.

dialer preemption level

To set the precedence for voice calls to be preempted by a dial-on demand routing (DDR) call for the dialer map, use the **dialer preemption level** command in map-class dialer configuration mode. To remove the preemption setting, use the **no** form of this command.

dialer preemption level { **flash-override** | **flash** | **immediate** | **priority** | **routine** }

no dialer preemption level { **flash-override** | **flash** | **immediate** | **priority** | **routine** }

Syntax Description

flash-override	Sets the precedence for DDR calls to preemption level 0 (highest).
flash	Sets the precedence for DDR calls to preemption level 1.
immediate	Sets the precedence for DDR calls to preemption level 2.
priority	Sets the precedence for DDR calls to preemption level 3.
routine	Sets the precedence for DDR calls to preemption level 4 (lowest). This is the default.

Command Default

The preemption level default is **routine** (lowest).

Command Modes

Map-class dialer configuration

Command History

Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples

The following example sets a preemption level of priority (level 3) for the dialer map-class dial1.

```
Router(config)# map-class dialer dial1
Router(config-map-class)# dialer preemption level priority
```

Related Commands

Command	Description
dialer map	Configures a serial interface or ISDN interface to call one or multiple sites or to receive calls from multiple sites.
dialer trunkgroup	Defines the dial-on-demand trunk group label for the dialer interface.
map-class dialer	Defines a class of shared configuration parameters associated with the dialer map command for outgoing calls from an ISDN interface and for PPP callback.
preemption enable	Enables preemption capabilities on a trunk group.

Command	Description
preemption level	Sets the preemption level of the selected outbound dial peer. Voice calls can be preempted by a DDR call with higher preemption level.
preemption tone timer	Defines the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call.

dialer trunkgroup

To define the dial-on-demand trunk group label for the dialer interface, use the **dialer trunkgroup** command in map-class dialer configuration mode. To remove the trunk group label, use the **no** form of this command.

dialer trunkgroup *label*

no dialer trunkgroup *label*

Syntax Description

<i>label</i>	Unique name for the dialer interface trunk group. Valid names contain a maximum of 63 alphanumeric characters.
--------------	--

Command Default

No dialer trunk group is defined.

Command Modes

Map-class dialer configuration

Command History

Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples

The following example creates a trunk group named 20 for dialer map-class dial1.

```
Router(config)# map-class dialer dial1
Router(config-map-class)# dialer trunkgroup 20
```

Related Commands

Command	Description
dialer map	Configures a serial interface or ISDN interface to call one or multiple sites or to receive calls from multiple sites.
map-class dialer	Defines a class of shared configuration parameters associated with the dialer map command for outgoing calls from an ISDN interface and for PPP callback.
show dialer	Displays general diagnostic information for interfaces configured for dial-on-demand routing (DDR).
trunk group	Defines a trunk group (global configuration) and enters trunk group configuration mode.

isdn integrate calltype all

To enable integrated mode on an ISDN PRI interface, use the **isdn integrate calltype all** command in interface configuration mode. To disable integrated mode, use the **no** form of this command.

isdn integrate calltype all

no isdn integrate calltype all

Syntax Description

This command has no arguments or keywords.

Command Default

Integrated mode is disabled on the interface.

Command Modes

Interface configuration

Command History

Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

Configure this command from a PRI interface only. This command is not supported from a BRI interface. Any incoming calls from an interface that has been configured for integrate calltype all is rejected with cause-code **invalid number 0x1C** if inbound dial-peer is not selected.

Examples

In the following example, the interface is shut down.

```
Router(config)# interface Serial14/1:15
Router(config-if)# shutdown
```

In the following example, integrated mode is enabled.

```
Router(config)# interface Serial14/1:15
Router(config-if)# isdn integrate calltype all
% This command line will enable the Serial Interface to "integrated service" mode.
% The "isdn incoming-voice voice" setting will be removed from the interface.
% Continue? [confirm]
```

When you confirm, the default incoming-voice configuration is removed from the interface, and the interface is now in integrated service mode. The interface does not reset back to voice mode if an incoming call is originated from the interface.

In the following example, the interface is set to active.

```
Router(config)# interface Serial14/1:15
Router(config-if)# no shutdown
```

Related Commands

Command	Description
dial-peer data <i>tag</i> pots	Creates a data dial peer and enters dial-peer configuration mode.
dial-peer search type	Optimizes voice or data dial-peer searches.
isdn incoming-voice	Routes all incoming voice calls to the modem and determine how they will be treated.

preemption enable

To enable preemption capability on a trunk group, use the **preemption enable** command in trunk group configuration mode. To disable preemption capabilities, use the **no** form of this command.

preemption enable

no preemption enable

Syntax Description

This command has no arguments or keywords.

Command Default

Preemption is disabled on the trunk group.

Command Modes

Trunk group configuration

Command History

Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples

The following command example enables preemption capabilities on trunk group test:

```
Router(config)# trunk group test
Router(config-trunk-group)# preemption enable
```

Related Commands

Command	Description
isdn integrate all	Enables integrated mode on an ISDN PRI interface.
max-calls	Sets the maximum number of calls that a trunk group can handle.
preemption guard timer	Defines time for a DDR call and allows time to clear the last call from the channel.
preemption level	Sets the preemption level of the selected outbound dial peer. Voice calls can be preempted by a DDR call with higher preemption level.
preemption tone timer	Defines the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call.

preemption guard timer

To define the time for a DDR call and to allow time to clear the last call from the channel, use the **preemption guard timer** command in trunk group configuration mode. To disable the preemption guard time, use the **no** form of this command.

preemption guard timer *value*

no preemption guard timer

Syntax Description	<i>value</i>	Number, in milliseconds for the preemption guard timer. The range is 60 to 500. The default is 60.
---------------------------	--------------	--

Command Default	No preemption guard timer is configured.
------------------------	--

Command Modes	Trunk group configuration
----------------------	---------------------------

Command History	Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples	The following set of commands configures a 60-millisecond preemption guard timer on the trunk group dial2.
-----------------	--

```
Router(config)# trunk group dial2
Router(config-trunk-group)# preemption enable
Router(config-trunk-group)# preemption guard timer 60
```

Related Commands	Command	Description
	isdn integrate all	Enables integrated mode on an ISDN PRI interface.
	max-calls	Sets the maximum number of calls that a trunk group can handle.
	preemption enable	Enables preemption capabilities on a trunk group.
	preemption level	Sets the preemption level of the selected outbound dial-peer. Voice calls can be preempted by a DDR call with higher preemption level.
	preemption tone timer	Sets the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call.

preemption level

To set the precedence for voice calls to be preempted by a dial-on demand routing (DDR) call for the trunk group, use the **preemption level** command in dial-peer configuration mode. To restore the default preemption level setting, use the **no** form of this command

preemption level { **flash-override** | **flash** | **immediate** | **priority** | **routine** }

no preemption level

Syntax Description

flash-override	Sets the precedence for voice calls to preemption level 0 (highest).
flash	Sets the precedence for voice calls to preemption level 1.
immediate	Sets the precedence for voice calls to preemption level 2.
priority	Sets the precedence for voice calls to preemption level 3.
routine	Sets the precedence for voice calls to preemption level 4 (lowest). This is the default.

Command Default

The preemption level default is **routine** (lowest).

Command Modes

Dial-peer configuration

Command History

Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples

The following command example sets a preemption level of flash (level 1) on POTS dial-peer 20:

```
Router(config)# dial-peer voice 20 pots
Router(config-dial-peer)# preemption level flash
```

Related Commands

Command	Description
dialer preemption level	Sets the precedence for voice calls to be preempted by a DDR call for the dialer map.
isdn integrate all	Enables integrated mode on an ISDN PRI interface.
max-calls	Sets the maximum number of calls that a trunk group can handle.
preemption enable	Enables preemption capabilities on a trunk group.
preemption guard timer	Defines time for a DDR call and allows time to clear the last call from the channel.
preemption tone timer	Defines the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call.

preemption tone timer

To set the expiry time for the preemption tone for the outgoing call being preempted by a DDR backup call, use the **preemption tone timer** command in trunk group configuration mode. To clear the expiry time, use the **no** form of this command. To set the expiry time to the default (10 seconds), use the **default** form of this command.

preemption tone timer *seconds*

no preemption tone timer

default preemption tone timer

Syntax Description	<i>seconds</i>	Number of seconds for the preemption tone. The range is 4 to 30 seconds. The default value is 10.
---------------------------	----------------	---

Command Default	No preemption tone timer is configured.
------------------------	---

Command Modes	Trunk group configuration
----------------------	---------------------------

Command History	Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples The following set of commands configures a 20-second preemption tone timer on trunk group dial2.

```
Router(config)# trunk group dial2
Router(config-trunk-group)# preemption enable
Router(config-trunk-group)# preemption tone timer 20
```

The default value is 10 seconds. To change back to the default value, enter the following command:

```
Router(config-trunk-group)# default preemption tone timer
```

Related Commands	Command	Description
	isdn integrate all	Enables integrated mode on an ISDN PRI interface.
	max-calls	Sets the maximum number of calls that a trunk group can handle.
	preemption enable	Enables preemption capabilities on a trunk group.
	preemption level	Sets the preemption level of the selected outbound dial peer. Voice calls can be preempted by a DDR call with higher preemption level.

show trunk group

To display one or more trunk groups, use the **show trunk group** command in privileged EXEC mode.

show trunk group [*name* | **sort** {**ascending** | **descending**}]

Syntax Description	<i>name</i>	(Optional) Trunk group to display.
	sort	(Optional) Sorts the output by trunk group number, in ascending or descending order.
	ascending	Specifies ascending display order for the trunk groups. This is the default.
	descending	Specifies descending display order for the trunk groups.

Command Default Ascending order

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(11)T	This command was introduced.
	12.3(11)T	This command was enhanced to support dial-out trunk groups.
	12.4(4)XC	This command was implemented on the Cisco 2600XM series, Cisco 2800 series, Cisco 3700 series, and Cisco 3800 series.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

Use the *name* argument to display the information for a specific trunk group.

Use the **sort** keyword to display all the trunks groups in either ascending or descending order.

Examples

The following sample output shows that for trunk group 1, preemption is enabled, with a preemption tone timer of 10 seconds, and the preemption level is flash.

```
Router# show trunk group 1
Trunk group: 1
  Description:
    trunk group label: 1

  Translation profile (Incoming):
  Translation profile (Outgoing):

  Preemption is enabled
  Preemption Tone Timer is 10 seconds
  Preemption Guard Timer is 60 milliseconds

  Hunt Scheme is least-used
  Max Calls (Incoming):   NOT-SET (Any)   NOT-SET (Voice) NOT-SET
(Data)
```

```

Max Calls (Outgoing):  NOT-SET (Any)  NOT-SET (Voice) NOT-SET
(Data)
Retries: 0

Trunk Se0/3/0:15      Preference DEFAULT
    Member Timeslots : 1-5
    Total channels available : 5
    Data = 0, Voice = 0, Modem = 0, Pending = 0, Free = 5
Trunk Se0/3/1:15      Preference DEFAULT
    Member Timeslots : 1-2
    Total channels available : 0
    Data = 0, Voice = 0, Modem = 0, Pending = 0, Free = 0
Trunk Se1/0/0:15      Preference DEFAULT
    Member Timeslots : 1-31
    Total channels available : 0
    Data = 0, Voice = 0, Modem = 0, Pending = 0, Free = 0
Trunk Se1/0/1:15      Preference DEFAULT
    Member Timeslots : 1-10
    Total channels available : 0
    Data = 0, Voice = 0, Modem = 0, Pending = 0, Free = 0

Total calls for trunk group: Data = 0, Voice = 0, Modem = 0
                             Pend = 0, Free = 5

Preemption Call Type:  Active  Pending
Flash-Override        NA      0
Flash                 0       0
Immediate             0       0
Priority               0       0
Routine               0       0

Total                 0       0

Active preemption call-type shows the number of calls
of each priority level which can be preempted by
higher preemption level calls.

Pending preemption call-type shows the number of calls
of each priority level which are pending for the completion
of call preemption.

advertise_flag 0x00000040, capacity timer 25 sec tripl_config_mask 0x00000000
AC_curr 5, FD_curr 0, SD_curr 0

succ_curr 0 tot_curr 1
succ_report 0 tot_report 1
changed 1 replacement position 0

```

Table 2 describes the significant fields shown in the output. Fields are listed in alphabetical order.

Table 2 *show trunk group Field Descriptions*

Field	Description
Data	Number of currently used data channels on the trunk or total data calls used by the trunk group.
Description	Description of the trunk group if entered with the description (trunk group) command.
Free	Number of currently available channels on the trunk or total available calls for the trunk group.

Table 2 *show trunk group Field Descriptions (Continued)*

Field	Description
Hunt Scheme	Name of the idle channel hunt scheme used for this trunk group.
Max calls (incoming)	Maximum number of incoming calls handled by this trunk group.
Max calls (outgoing)	Maximum number of outgoing calls handled by this trunk group.
Member timeslots	Member timeslots for this trunk.
Pending	Number of pending channels.
Preemption Call Type	List of preemption levels for active and pending calls.
Preemption is	Indicates whether preemption is enabled or disabled.
Preemption level	The preemption level for voice calls to be preempted by a DDR call.
Preemption tone timer	The expiry time for the preemption tone for the outgoing calls being preempted by a DDR call.
Preference	Preference of the trunk in the trunk group. If DEFAULT appears, the trunk does not have a defined preference.
Retries	Number of times the gateway tries to complete the call on the same trunk group.
Total calls for trunk group	List of the total calls across all trunks in the trunk group.
Total channels available	Number of available channels for the trunk.
Translation profile (Incoming)	List of incoming translation profiles.
Translation profile (Outgoing)	List of outgoing translation profiles.
Trunk group	ID of the trunk group member.
trunk group label	Name of the trunk group.
Voice	Number of currently used voice channels on the trunk or total voice calls used by the trunk group.

Related Commands

Command	Description
description (trunk group)	Includes a specific description of the trunk group interface.
hunt-scheme least-idle	Specifies the method for selecting an available incoming or outgoing channel.
trunk group	Initiates a trunk group definition.
trunk group timeslots	Directs an outbound synchronous or asynchronous call initiated by DDR to use specific DS0 channels of an ISDN circuit.

Feature Information for Integrating Data and Voice Services for ISDN PRI Interfaces

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

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



Note

The following table lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

History for the Integrating Data and Voice Services for ISDN PRI Interfaces Feature

Release	Modification
12.4(4)XC	This feature was introduced.
12.4(9)T	This feature was integrated into Cisco IOS Release 12.4(9)T.

Glossary

multi-level precedence and preemption (MLPP)—A service that allows placement of priority calls. Precedence designates the priority level that is associated with a call. Preemption designates the process of terminating lower-precedence calls so that a call of higher precedence can be extended.



Note

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

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

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

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

