Digital Private Network Signaling System Backhaul

This feature introduces support for Digital Private Network Signaling System (DPNSS) Layer 2 functionality on the Cisco gateway (GW) for Cisco IOS Release 12.2(15)ZJ and 12.3(4)T. This feature supports Layer 3 backhauling to a Cisco PGW2200 using DPNSS and Digital Access Signaling System (DASS) User Adaptation (DUA) over Stream Control Transmission Protocol (SCTP).

DPNSS was developed by British Telecom and is used in the United Kingdom and some parts of Europe. DPNSS is a standard and open protocol used between PBXs in a private network that enables complex features to work on a network basis. This feature applies the DPNSS backhaul solution on Cisco gateways to provide connectivity and services to the PBXs that are running the DPNSS protocol.

Note The DPNSS protocol can run on both T1 and E1 interfaces, but only E1 interfaces are supported by this feature.

The DPNSS Backhaul feature includes the following benefits:

- DUA works with existing Q.931 or DPNSS and DASS-2 protocols on an application server process (ASP), in this case, the Cisco PGW2200.
- The ISDN User Adaptation layer (IUA) with DUA and SCTP protocol stacks are written to be portable across operating systems and products.
- Memory allocation and system performance are not negatively affected by this feature.

Feature History for the DPNSS Backhaul Feature

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

Finding Support Information for Platforms and Cisco IOS Software Images

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

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

Contents

Prerequisites for DPNSS Backhaul

- You must have a suitable customer premises equipment (CPE) gateway with DPNSS backhaul capability and Media Gateway Control Protocol (MGCP) control for bearer circuit connections.
- You must be running Cisco PGW2200 release 9.4 or a later release.

Restrictions for DPNSS Backhaul

- The DPNSS Backhaul feature does not support Layer 3 debugging at the GW.

Information About DPNSS Backhaul

Before you configure the DPNSS Backhaul feature, you should understand the following concepts:

DPNSS Layer 2

DPNSS Layer 2 functions like Q.921 by transferring information between two DPNSS Layer 3 entities in sequence using the D channel. DPNSS Layer 2 has multiple instances, one for each channel (in contrast to Q.921 for ISDN PRI, which has one Layer 2 instance independent of individual channels).

Up to 30 real and 30 virtual channels can be supported on E1 interfaces, and up to 23 real and 23 virtual channels are supported on T1 interfaces. A real channel is a bearer channel that can be used to convey voice and data, and a virtual channel is an informational channel used for some supplementary services that has no physical realization outside the D channel.

DPNSS data link layer (Layer 2) function is carried out by the GW, while Layer 3 processing occurs at the Cisco PGW2200.

DPNSS Backhaul Using DUA

The DPNSS Backhaul feature implements IUA and DUA with SCTP for an ASP. The DUA protocol is used for backhauling DPNSS Layer 2 messages between the GW and the Cisco PGW2200 over IP using SCTP. IUA is used for backhauling ISDN Layer 3 messages.

DUA supports signaling gateways (SGs) on a single protocol stack that terminates ISDN and DPNSS connections. Each interface identifier (D channel) can be an ISDN or a DPNSS D channel. A flag is used when sending a message to indicate whether the message is IUA or DUA.

Figure 1 shows Q.931 backhaul using DUA over SCTP.

Figure 1 Layer 3 Backhaul

The architectural diagram shown in Figure 2 depicts various layers used for DPNSS backhaul.

Figure 2 DPNSS Backhaul Layers

DUA Message Classes

New message classes have been created for DUA. All IUA messages are classified in various classes, for instance, message class 5, which defines Q.921 and Q.931 boundary primitives transport (QPTM).

Because DPNSS L2-to-L3 primitives need to be handled in a different way from the Q.921-to-Q.931, a new message class has been created: 10—DPNSS Boundary Primitives Transport Messages (DPTM).

DUA queries the status of data link connections (DLCs), so an additional API call is available for this function that is used only for DUA. There is an additional call back notification when DLC status is received. DUA also has an additional callback notification in order to identify "channel not configured." This notification assists the operator in identifying configuration errors.

For more information about DUA, refer to the Internet Engineering Task Force (IETF) draft, *DPNSS/DASS 2 Extensions to the IUA Protocol*.

IUA over SCTP

IETF Signaling Transport (Sigtran) standard, IUA with SCTP, acts as the call signaling IP transport mechanism in a Voice Gateway solution. SCTP and IUA are used for Signaling System 7 (SS7) Interconnect solutions, which allows required flexibility in connecting Inter-Machine Trunks (IMTs) from more than one PSTN switch (multiple trunk groups) to the Cisco 2600 series, Cisco 3600 series, and other platforms that support ISDN PRI and IUA backhaul.

IUA and SCTP protocols provide the following support:

- Trunk groups are defined on a T1/E1 interface basis.
- All digital signal level 0 (DS0) bearer channels in a specific T1/E1 interface are included in the same trunk group and cannot be split into different trunk groups.
- Multiple T1/E1 interfaces on the same gateway can be provisioned in a single trunk group or split into multiple trunk groups. The maximum number of trunk groups that a platform can support is equal to the maximum number of T1/E1 interfaces that the platform can configure.

The IUA module creates the associations between the SGs and the media gateway controller (MGC) based on configuration commands. IUA also manages multiple ASPs as defined in the IETF IUA specification. IUA performs the following additional functions:

- Requests SCTP associations based on configuration information.
- Manages the destination address list and requests a new primary destination in the event of a failure.
- Manages the ASP state machine for each association.
- Manages the application server (AS) state machine across all ASPs associated with a single application.
- Provides service for multiple applications simultaneously to handle different Layer 3 signaling protocols (Q.931 and Q.SIG, for example) or to communicate with different sets of call agents.

For more information about IUA with SCTP, refer to the *Support for IUA with SCTP for Cisco Access Servers* and *PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer* feature documents on Cisco.com.

How to Configure DPNSS Backhaul

This section contains the following procedures:

Configuring DPNSS Backhaul

To configure DPNSS backhaul, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot*/*port*:*ds0*
4. **isdn bind-l3 iua-backhaul** *as-name*
5. **isdn switch-type** {**primary-dpnss**}
6. **isdn dpnss pbxA**
7. **isdn dpnss layer2 timers** [**Tretry** *timer-value*] [**Ttest** *timer-value*]
8. **isdn dpnss layer2 retry max-count** *range*
9. **isdn dpnss layer2 test frame**
10. **no shutdown**
11. **exit**

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*:*ds0* **Example:** `Router(config)# interface serial 2/0:15`	Enters interface configuration mode on the selected slot and port.
Step 4	**isdn bind-l3 iua-backhaul** *as-name* **Example:** `Router(config-if)# isdn bind-l3 iua-backhaul as1`	Binds Layer 3 for DPNSS to DUA.
Step 5	**isdn switch-type primary-dpnss** **Example:** `Router(config-if)# isdn switch-type primary-dpnss`	Sets the primary ISDN switch type to DPNSS.

	Command or Action	Purpose
Step 6	**isdn dpnss pbxA** **Example:** `Router(config-if)# isdn dpnss pbxA`	Sets ISDN DPNSS to act as PBX A.
Step 7	**isdn dpnss layer2 timers** [**Tretry** *timer-value*] [**Ttest** *timer-value*] **Example:** `Router(config-if)# isdn dpnss layer2 timers Tretry 500 Ttest 5`	Sets the Tretry and Ttest timers. • When the Tretry timer expires, the L2 frame is retransmitted if unacknowledged. Valid retry time values range from 5 ms to 10 seconds. Default is 500 ms. • When the Ttest timer expires, frames are sent on all the DLCs. Valid test time values range from 1 to 60 minutes. Default is 5 minutes.
Step 8	**isdn dpnss layer2 retry max-count** *range* **Example:** `Router(config-if)# isdn dpnss layer2 retry max-count 4`	Selects the number of times that a frame is retried if unacknowledged. • Maximum count range is from 0 to 64. Default is 4.
Step 9	**isdn dpnss layer2 test frame** **Example:** `Router(config-if)# isdn dpnss layer2 test frame`	(Optional) Allows test frames to be sent periodically.
Step 10	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Saves the configuration.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and completes the configuration.

Troubleshooting Tips

- To display all DPNSS Layer 2 debugging message exchanges between the GW and the Cisco PGW2200 and between the GW and the PBX, enter the **debug isdn q921** command.
- To display the DPNSS L2 frame information, enter the **debug isdn q921 frame** command.
- To display L2 related debugging information, enter the **debug isdn q921 detail** command.
- To debug DUA packets, enter the **debug isdn api backhaul** command.

Verifying DPNSS Backhaul

To verify DPNSS backhaul, follow these steps:

	Command or Action	Purpose
Step 1	**show running-config** **Example:** `Router# show running-config`	Verifies that DPNSS backhaul is configured. See the "DPNSS Backhaul Configuration Example" section on page 7.
Step 2	**show isdn status** **Example:** `Router# show isdn status`	Verifies that shutting down the interface or controller has brought Layer 2 down (OOS) for all the DLCs on the interface. See the "DLC Layer 2 Status Example" section on page 11.

Configuration Examples for DPNSS Backhaul

This section contains the following examples:

DPNSS Backhaul Configuration Example

The following **show running-config** output examples provide two overviews of how to create an IUA association and how the IUA configuration is associated with the ISDN component.

Cisco 3640 Configured for Many Associations

```
Router# show running-config
!
version 12.2
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
!
hostname
!
logging queue-limit 100
enable secret 5 $1$lOJK$U8UvBbI/68/YhAURdXBZ4/
!
username nas1 password 7 06110A2D4F41041C
clock timezone PST -8
voice-card 2
!
voice-card 3
!
ip subnet-zero
!
!
no ip domain lookup
ip host dpnss-pgw 10.0.58.120
ip host d 170.00.0.000
ip host dirt 170.00.1.000
!
isdn switch-type basic-5ess
!
!
```

```
!
!
!
!
!
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
iua
   ASP asp2 IP-Precedence 10.0.58.000 1
   AS dpnss 10.0.58.00 9900
   ASP asp3 AS dpnss 10.0.58.000 9900
   ASP asp3 IP-Precedence 10.0.58.000 1
!
!
controller E1 2/0
 pri-group timeslots 1-31 service mgcp
!
controller E1 2/1
!
controller T1 3/0
 shutdown
 framing esf
 clock source internal
 linecode b8zs
 cablelength short 133
!
controller T1 3/1
!
controller T1 3/1
 shutdown
 framing esf
 clock source internal
 linecode b8zs
 cablelength short 133
!
!
!
interface Ethernet0/0
 ip address 10.0.58.65 255.000.000.0
 half-duplex
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
interface Serial2/0:15
 no ip address
 no logging event link-status
```

```
 isdn switch-type primary-dpnss
 isdn incoming-voice modem
 isdn bind-l3 iua-backhaul dpnss
!
no ip http server
no ip classless
ip route 0.0.0.0 0.0.0.0 10.0.58.1
!
!
!
dialer-list 1 protocol ip permit
!
call rsvp-sync
!
voice-port 2/0:15
!
mgcp
mgcp vad
mgcp package-capability as-package
mgcp default-package as-package
no mgcp validate domain-name
!
mgcp profile default
!
!
!
dial-peer cor custom
!
!
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0
 no login
line vty 1 4
 login
!
!
end
```

Cisco 3745 Configured for Basic Operation

```
Router# show running-config
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname
!
logging queue-limit 100
enable secret 5 $S.
enable password
!
voice-card 1
 dspfarm
!
ip subnet-zero
no ip routing
```

```
!
!
no ip domain lookup
!
mpls ldp logging neighbor-changes
isdn switch-type primary-5ess
!
!
!
!
!
!
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
iua
  AS dpnss 10.0.58.00 9902
  ASP asp3 AS dpnss 10.0.58.000 9902
  ASP asp3 IP-Precedence 10.0.58.000 1
!
!

!
controller E1 1/0
 pri-group timeslots 1-31 service mgcp
!
controller E1 1/1
!
!
!
interface FastEthernet0/0
 ip address 10.0.00.71 255.000.000.0
 no ip route-cache
 no ip mroute-cache
 speed auto
 half-duplex
!
interface FastEthernet0/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
!
interface Serial1/0:15
 no ip address
 no logging event link-status
 isdn switch-type primary-dpnss
 isdn incoming-voice modem
 isdn bind-l3 iua-backhaul dpnss
 isdn dpnss pbxA
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.58.1
!
ip pim bidir-enable
!
```

```
!
!
!
!
call rsvp-sync
!
voice-port 1/0:15
!
mgcp
mgcp vad
mgcp package-capability as-package
mgcp default-package as-package
no mgcp validate domain-name
!
mgcp profile default
!
!
!
dial-peer cor custom
!
!
!
!
line con 0
line aux 0
line vty 0 4
 password xx
line vty 0 4
 password xxx
 login
!
end
```

DLC Layer 2 Status Example

The following example shows the Layer 2 status of all DLCs on the interface:

```
Router# show isdn status

Global ISDN Switchtype = basic-5ess
        dsl 1, interface ISDN Switchtype = primary-dpnss
        L2 Protocol = Q.921  L3 Protocol(s) = IUA BACKHAUL
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        Channel 1 :INFORMATION TRANSFER,
        Channel 2 :RESET COMPLETE,
        Channel 3 :INFORMATION TRANSFER,
        Channel 4 :RESET COMPLETE,
        Channel 5 :INFORMATION TRANSFER,
        Channel 6 :RESET COMPLETE,
        Channel 7 :INFORMATION TRANSFER,
        Channel 8 :RESET COMPLETE,
        Channel 9 :INFORMATION TRANSFER,
        Channel 10:RESET COMPLETE,
        Channel 11:INFORMATION TRANSFER,
        Channel 12:RESET COMPLETE,
        Channel 13:INFORMATION TRANSFER,
        Channel 14:RESET COMPLETE,
        Channel 15:INFORMATION TRANSFER,
        Channel 17:INFORMATION TRANSFER,
        Channel 18:RESET COMPLETE,
```

```
Channel 19:INFORMATION TRANSFER,
Channel 20:RESET COMPLETE,
Channel 21:INFORMATION TRANSFER,
Channel 22:RESET COMPLETE,
Channel 23:INFORMATION TRANSFER,
Channel 24:RESET COMPLETE,
Channel 25:INFORMATION TRANSFER,
Channel 26:RESET COMPLETE,
Channel 27:INFORMATION TRANSFER,
Channel 28:RESET COMPLETE,
Channel 29:INFORMATION TRANSFER,
Channel 30:RESET COMPLETE,
Channel 31:INFORMATION TRANSFER,
Channel 33:INFORMATION TRANSFER,
Channel 34:RESET COMPLETE,
Channel 35:INFORMATION TRANSFER,
Channel 36:RESET COMPLETE,
Channel 37:INFORMATION TRANSFER,
Channel 38:RESET COMPLETE,
Channel 39:INFORMATION TRANSFER,
Channel 40:RESET COMPLETE,
Channel 41:INFORMATION TRANSFER,
Channel 42:RESET COMPLETE,
Channel 43:INFORMATION TRANSFER,
Channel 44:RESET COMPLETE,
Channel 45:INFORMATION TRANSFER,
Channel 46:RESET COMPLETE,
Channel 47:INFORMATION TRANSFER,
Channel 49:INFORMATION TRANSFER,
Channel 50:RESET COMPLETE,
Channel 51:INFORMATION TRANSFER,
Channel 52:RESET COMPLETE,
Channel 53:INFORMATION TRANSFER,
Channel 54:RESET COMPLETE,
Channel 55:INFORMATION TRANSFER,
Channel 56:RESET COMPLETE,
Channel 57:INFORMATION TRANSFER,
Channel 58:RESET COMPLETE,
Channel 59:INFORMATION TRANSFER,
Channel 60:RESET COMPLETE,
Channel 61:INFORMATION TRANSFER,
Channel 62:RESET COMPLETE,
Channel 63:INFORMATION TRANSFER,
```

Additional References

For additional information related to DPNSS Backhaul, refer to the following references:

Related Documents

Related Topic	Document Title
Specific command information for Cisco IOS Release 12.3 T	*Cisco IOS Dial Technologies Command Reference, Release 12.3*
Cisco IOS commands to support voice, video, and fax applications	*Cisco IOS Voice Command Reference, Release 12.3 T*
How to configure your Cisco router or access server to support voice, video, and fax applications.	*Cisco IOS Voice Configuration Library, Release 12.3*
Cisco MGC installation and configuration	*Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide*
Cisco MGC software Release 9 informational events, alarms, and log messages	*Cisco Media Gateway Controller Software Release 9 Messages Reference Guide*
Cisco MGC software Release 9 Man-Machine Language (MML) commands	*Cisco Media Gateway Controller Software Release 9 MML Command Reference*
Operating, maintaining, and troubleshooting the core elements of the MGC node	*Cisco Media Gateway Controller Software Release 9 Operations, Maintenance, and Troubleshooting Guide*
Provisioning the Cisco MGC	*Cisco Media Gateway Controller Software Release 9 Provisioning Guide*
Cisco SLT information	Cisco Signaling Link Terminal index page
Distributed system that provides SS7 connectivity for VoIP gateways by using the Cisco PGW2200	*Cisco SS7 Interconnect for Voice Gateways Solution*, Release 2.0
IP Transfer Point (ITP) information	*IP Transfer Point (ITP)*, Release 12.2(2)MB
PRI/Q.921 signaling backhaul	*PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer*
SCTP feature information	*Stream Control Transmission Protocol (SCTP), Release 2*
IUA with SCTP information	*Support for IUA with SCTP for Cisco Access Servers*

Standards

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

MIBs

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

RFCs

RFCs	Title
DUA (unnumbered)	*Digital Private Network Signaling System (DPNSS)/ Digital Access Signaling System 2 (DASS 2) Extensions to the IUA Protocol*

Technical Assistance

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

Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 T command reference publications.

Modified Commands

- **debug isdn q921**
- **isdn bind-l3 iua-backhaul**
- **isdn dpnss**
- **isdn switch-type (PRI)**
- **show isdn**

debug isdn q921

To display data link layer (Layer 2) access procedures that are taking place at the router on the D channel (Link Access Procedure or LAPD) of its ISDN interface, use the **debug isdn q921** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug isdn q921 [**detail** | **frame** | **interface** [**bri** *number*]]

no debug isdn q921 [**detail** | **frame** | **interface**]

Syntax Description

detail	Displays ISDN Q.921 packet detail.
frame	Displays ISDN Q.921 frame contents.
interface	Specifies an interface for debugging.
bri *number*	(Optional) Specifies the BRI interface and selects the interface number. Valid values are from 0 to 6.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.
12.2(15)ZJ	The **detail** and **frame** keywords were added.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

The ISDN data link layer interface provided by the router conforms to the user interface specification defined by ITU-T recommendation Q.921. The **debug isdn q921** command output is limited to commands and responses exchanged during peer-to-peer communication carried over the D channel. This debug information does not include data transmitted over the B channels that are also part of the router ISDN interface. The peers (data link layer entities and layer management entities on the routers) communicate with each other with an ISDN switch over the D channel.

Note The ISDN switch provides the network interface defined by Q.921. This debug command does not display data link layer access procedures taking place within the ISDN network (that is, procedures taking place on the network side of the ISDN connection). Refer to Appendix B, "ISDN Switch Types, Codes, and Values," in the *ISDN Switch Types, Codes, and Values* document on Cisco.com for a list of the supported ISDN switch types.

A router can be the calling or called party of the ISDN Q.921 data link layer access procedures. If the router is the calling party, the command displays information about an outgoing call. If the router is the called party, the command displays information about an incoming call and the keepalives.

The **debug isdn q921** command can be used with the **debug isdn event**, **debug isdn q931**, **debug isdn q921 frame**, and **debug isdn q921 detail** commands at the same time. The displays are intermingled.

Use the **service timestamps debug datetime msec** global configuration command to include the time with each message.

Examples

The following is example output for a single active data link connection (DLC). The debugs turned on are **debug isdn q921**, **debug isdn q921 frame**, and **debug isdn q921 detail**. In the debugs below, "Q921" followed by a colon (:) indicates that **debug isdn q921** has been entered. "Q921" followed by the letter "f" indicates that **debug isdn q921 frame** has been entered. "Q921" followed by the letter "d" indicates that **debug isdn q921 detail** has been entered.

The following output shows that the L2 frame is received. The first two octets form the address field, while the third octet forms the control field. The address field identifies the originator of a frame and whether it is a command or a response. The second octet of the address field identifies the DLC with which the frame is associated. The control field (third octet) contains the frame type code and sequence number information.

```
00:12:10:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
00:12:10:ISDN Se1:15 Q921f:PBXb RX <- 0x0E03EF
```

The following output interprets the octet information. String "PBXb" indicates that the side receiving (RX) this frame is acting as a PBXb (as opposed to PBXa, which is the other possibility). This example also gives information about the type of frame received (SABMR), the associated DLC (1), the frame type code received from the control field (cntl=SABMR), and the sequence number (indicated by nbit, which is 0 in this case).

```
00:12:10:ISDN Se1:15 Q921:PBXb RX <- SABMR dlci=1 cntl=SABMR nbit=0
```

The following output shows information received from the driver (source_id of x200) showing an L2 frame (event x141). This results from the SABMR frame that was received from the peer PBX (v_bit and chan do not have any significance in this case).

```
00:12:10:ISDN Se1:15 Q921d:process_rxdata:Frame sent to L2
00:12:10:ISDN  Q921d:isdn_from_driver_process:event_count 3
00:12:10:ISDN Se1:15 Q921d:dpnss_l2_main:source_id x200 event x141 v_bit x0 chan x0
```

The following output shows that DPNSS L2 for DLC 1 (chan 1) has received an SABMR frame (event x0) in the IDLE state (s_dpnss_idle):

```
00:12:10:ISDN Se1:15 Q921d:s_dpnss_idle:event x0 chan 1
```

The following output shows that for DLC 1 (chan 1 above), a UA frame (event x1) needs to be sent to the driver (dest x200):

```
00:12:10:ISDN Se1:15 Q921d:dpnss_l2_mail:dest x200 event x1 v_bit 1 chan 1 out_pkt
x630531A4
```

The following output shows that for DLC 1, a DL_EST_IND (event x201) needs to be sent to L3 (DUA in this case because of the backhauling) indicating that this DLC is now up (in RESET COMPLETE state):

```
00:12:10:ISDN Se1:15 Q921d:dpnss_l2_mail:dest x300 event x201 v_bit 1 chan 1 out_pkt x0
```

The following output shows that the L2 frame is transmitted (TX):

```
00:12:10:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
00:12:10:ISDN Se1:15 Q921f:PBXb TX -> 0x0E0363
```

The following output shows that string “PBXb” is the side transmitting (TX) and that this frame is acting as PBX B. This example also gives information about the associated DLC (1), the frame type code transmitted from the control field (cntl=UA), and the sequence number (indicated by nbit, which is 0 in this case).

```
00:12:10:ISDN Se1:15 Q921:PBXb TX -> UA dlci=1 cntl=UA nbit=0
```

The following is complete debugging output from a DPNSS call:

```
Jan  8 17:24:43.499:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.499:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:43.499:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:43.499:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.503:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.503:ISDN Se2/0:15 Q921f:PBXa RX <-
        0x44030300102A34232A35302A33333330
Jan  8 17:24:43.503:   303030312334343430303031
Jan  8 17:24:43.503:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
        i=0x00102A34232A35302A3333333030303031233434343030303031
Jan  8 17:24:43.503:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.503:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.507:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
        x141 v_bit x0 chan x0
Jan  8 17:24:43.507:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2
        chan 1
Jan  8 17:24:43.507:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
        v_bit 1 chan 1 out_pkt x63F183D4
Jan  8 17:24:43.507:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.507:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:43.507:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:43.507:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.515:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921f:PBXa RX <-
        0x44030300102A34232A35302A33333330
Jan  8 17:24:43.515:   303030312334343430303031
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
        i=0x00102A34232A35302A3333333030303031233434343030303031
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.515:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
        x141 v_bit x0 chan x0
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2
        chan 1
Jan  8 17:24:43.515:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
        v_bit 1 chan 1 out_pkt x63F183D4
Jan  8 17:24:43.515:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.519:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:43.519:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:43.519:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.599:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x4 event x240
        v_bit x0 chan x2
Jan  8 17:24:43.599:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event
        x240 chan 1
Jan  8 17:24:43.599:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x2
        v_bit 1 chan 1 out_pkt x63EE5780
Jan  8 17:24:43.599:ISDN Se2/1:15 LIFd:LIF_StartTimer:timer (0x63E569A8),
        ticks (500), event (0x1201)
Jan  8 17:24:43.599:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.599:ISDN Se2/1:15 Q921f:PBXa TX ->
        0x46030300102A31232A35302A33333330
Jan  8 17:24:43.599:   303030312334343430303031
Jan  8 17:24:43.599:ISDN Se2/1:15 Q921:PBXa TX -> UI(C) dlci=1 cntl=UI nbit=0
        i=0x00102A31232A35302A3333333030303031233434343030303031
Jan  8 17:24:43.599:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
```

```
Jan  8 17:24:43.623:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.623:ISDN Se2/1:15 Q921f:PBXa RX <- 0x460303
Jan  8 17:24:43.623:ISDN Se2/1:15 Q921:PBXa RX <- UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:43.623:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.623:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.627:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:43.627:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x3
       chan 1
Jan  8 17:24:43.719:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440313092A34232A35302A3434343030
Jan  8 17:24:43.719:   303031232A31382A33312A33312A3331
Jan  8 17:24:43.719:   23
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x092A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.719:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x300 event x241
       v_bit 1 chan 1 out_pkt x63EE5780
Jan  8 17:24:43.719:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63EE57CC
Jan  8 17:24:43.723:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.723:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:24:43.723:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:24:43.723:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.727:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.727:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440313092A34232A35302A3434343030
Jan  8 17:24:43.727:   303031232A31382A33312A33312A3331
Jan  8 17:24:43.727:   23
Jan  8 17:24:43.727:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x092A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:43.727:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.727:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.731:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:43.731:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:43.731:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63EE57CC
Jan  8 17:24:43.731:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.731:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:24:43.731:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:24:43.731:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.739:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440313092A34232A35302A3434343030
Jan  8 17:24:43.739:   303031232A31382A33312A33312A3331
Jan  8 17:24:43.739:   23
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x092A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.739:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:43.739:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63EE57CC
```

```
Jan  8 17:24:43.739:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.743:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:24:43.743:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:24:43.743:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.787:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x4 event x240
       v_bit x0 chan x2
Jan  8 17:24:43.787:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event
       x240 chan 1
Jan  8 17:24:43.787:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x2
       v_bit 1 chan 1 out_pkt x636B1B64
Jan  8 17:24:43.787:ISDN Se2/0:15 LIFd:LIF_StartTimer:timer (0x63A4AFBC),
       ticks (500), event (0x1201)
Jan  8 17:24:43.791:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:43.791:ISDN Se2/0:15 Q921f:PBXa TX ->
       0x460313092A31232A35302A3434343030
Jan  8 17:24:43.791:   30303123
Jan  8 17:24:43.791:ISDN Se2/0:15 Q921:PBXa TX -> UI(C) dlci=1 cntl=UI nbit=1
       i=0x092A31232A35302A343434303030303123
Jan  8 17:24:43.791:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:43.811:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.811:ISDN Se2/0:15 Q921f:PBXa RX <- 0x460313
Jan  8 17:24:43.811:ISDN Se2/0:15 Q921:PBXa RX <- UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:24:43.811:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.811:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.811:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:43.811:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x3
       chan 1
Jan  8 17:24:52.107:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:52.107:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440303052A34232A35302A3434343030
Jan  8 17:24:52.107:   303031232A31382A33312A33312A3331
Jan  8 17:24:52.107:   23
Jan  8 17:24:52.107:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
       i=0x052A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:52.107:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:52.107:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x300 event x241
       v_bit 1 chan 1 out_pkt x63F19CC8
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63F19D14
Jan  8 17:24:52.111:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:52.111:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:52.111:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:52.119:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440303052A34232A35302A3434343030
Jan  8 17:24:52.119:   303031232A31382A33312A33312A3331
Jan  8 17:24:52.119:   23
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
       i=0x052A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:52.119:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:52.119:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63F19D14
```

```
Jan  8 17:24:52.119:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:52.123:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:52.123:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:52.123:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:52.127:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:52.127:ISDN Se2/1:15 Q921f:PBXa RX <-
       0x440303052A34232A35302A3434343030
Jan  8 17:24:52.127:   303031232A31382A33312A33312A3331
Jan  8 17:24:52.127:   23
Jan  8 17:24:52.127:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
       i=0x052A34232A35302A34343430303031232A31382A33312A33312A333123
Jan  8 17:24:52.127:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:52.127:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:52.131:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:52.131:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:24:52.131:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63F19D14
Jan  8 17:24:52.131:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:52.131:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440303
Jan  8 17:24:52.131:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:52.131:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:52.159:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x4 event x240
       v_bit x0 chan x2
Jan  8 17:24:52.159:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event
       x240 chan 1
Jan  8 17:24:52.159:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x2
       v_bit 1 chan 1 out_pkt x63F19CC8
Jan  8 17:24:52.159:ISDN Se2/0:15 LIFd:LIF_StartTimer:timer (0x63A4AFBC),
       ticks (500), event (0x1201)
Jan  8 17:24:52.159:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:24:52.159:ISDN Se2/0:15 Q921f:PBXa TX ->
       0x460303052A35302A3434343030303031
Jan  8 17:24:52.159:   23
Jan  8 17:24:52.159:ISDN Se2/0:15 Q921:PBXa TX -> UI(C) dlci=1 cntl=UI nbit=0
       i=0x052A35302A343434303030303123
Jan  8 17:24:52.159:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:24:52.179:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:52.179:ISDN Se2/0:15 Q921f:PBXa RX <- 0x460303
Jan  8 17:24:52.179:ISDN Se2/0:15 Q921:PBXa RX <- UI(R) dlci=1 cntl=UI nbit=0
Jan  8 17:24:52.179:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:52.183:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:52.183:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:24:52.183:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x3
       chan 1
Jan  8 17:25:31.811:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:31.811:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x300 event x241
       v_bit 1 chan 1 out_pkt x63F1806C
Jan  8 17:25:31.811:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x636710B8
Jan  8 17:25:31.815:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:31.815:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:31.815:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
```

```
Jan  8 17:25:31.815:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:31.819:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:31.819:ISDN Se2/0:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:31.819:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:31.819:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:31.819:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:31.823:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:31.823:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:31.823:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x63F19CC8
Jan  8 17:25:31.823:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:31.823:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:31.823:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:31.823:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:31.831:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:31.831:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:31.831:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x636710B8
Jan  8 17:25:31.835:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:31.835:ISDN Se2/0:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:31.835:ISDN Se2/0:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:31.835:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:31.851:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x4 event x240
       v_bit x0 chan x2
Jan  8 17:25:31.851:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event
       x240 chan 1
Jan  8 17:25:31.851:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x2
       v_bit 1 chan 1 out_pkt x63F1806C
Jan  8 17:25:31.851:ISDN Se2/1:15 LIFd:LIF_StartTimer:timer (0x63E569A8),
       ticks (500), event (0x1201)
Jan  8 17:25:31.851:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:31.855:ISDN Se2/1:15 Q921f:PBXa TX -> 0x4603130830
Jan  8 17:25:31.855:ISDN Se2/1:15 Q921:PBXa TX -> UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:31.855:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:31.875:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:31.875:ISDN Se2/1:15 Q921f:PBXa RX <- 0x460313
Jan  8 17:25:31.875:ISDN Se2/1:15 Q921:PBXa RX <- UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:31.875:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:31.875:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:31.875:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:31.875:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x3
       chan 1
Jan  8 17:25:31.879:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x4 event x240
       v_bit x0 chan x2
Jan  8 17:25:31.879:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event
       x240 chan 1
Jan  8 17:25:31.879:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x2
       v_bit 1 chan 1 out_pkt x63EFC5AC
Jan  8 17:25:31.879:ISDN Se2/0:15 LIFd:LIF_StartTimer:timer (0x63A4AFBC),
       ticks (500), event (0x1201)
Jan  8 17:25:31.879:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
```

```
Jan  8 17:25:31.879:ISDN Se2/0:15 Q921f:PBXa TX -> 0x4603130830
Jan  8 17:25:31.879:ISDN Se2/0:15 Q921:PBXa TX -> UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:31.883:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:31.899:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:31.899:ISDN Se2/0:15 Q921f:PBXa RX <- 0x460313
Jan  8 17:25:31.899:ISDN Se2/0:15 Q921:PBXa RX <- UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:31.899:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:31.899:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:31.903:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:31.903:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x3
       chan 1
Jan  8 17:25:32.063:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:32.063:ISDN Se2/1:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:32.063:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:32.063:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:32.063:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x300 event x241
       v_bit 1 chan 1 out_pkt x63EFC5AC
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x6367175C
Jan  8 17:25:32.067:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:32.067:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:32.067:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:32.075:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:32.075:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x6367175C
Jan  8 17:25:32.075:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:32.075:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:32.079:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:32.079:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
Jan  8 17:25:32.083:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:25:32.083:ISDN Se2/1:15 Q921f:PBXa RX <- 0x4403130830
Jan  8 17:25:32.083:ISDN Se2/1:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=1
       i=0x0830
Jan  8 17:25:32.083:ISDN Se2/1:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:25:32.083:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:25:32.087:ISDN Se2/1:15 Q921d:dpnss_l2_main:source_id x200 event
       x141 v_bit x0 chan x0
Jan  8 17:25:32.087:ISDN Se2/1:15 Q921d:s_dpnss_information_transfer:event x2
       chan 1
Jan  8 17:25:32.087:ISDN Se2/1:15 Q921d:dpnss_l2_mail:dest x200 event x3
       v_bit 1 chan 1 out_pkt x6367175C
Jan  8 17:25:32.087:ISDN  Q921d:isdn_l2d_srq_process:QUEUE_EVENT
Jan  8 17:25:32.087:ISDN Se2/1:15 Q921f:PBXa TX -> 0x440313
Jan  8 17:25:32.087:ISDN Se2/1:15 Q921:PBXa TX -> UI(R) dlci=1 cntl=UI nbit=1
Jan  8 17:25:32.087:ISDN  Q921d:isdn_l2d_srq_process:event_count 1
```

The following output shows details of the debugging events above.

The first two octets (0x4403) form the address field, while the third octet (0x03) is the control field. All the octets starting from the fourth constitute DPNSS L3 information, which needs to be backhauled to the Cisco PGW2200.

```
Jan  8 17:24:43.495:ISDN  Q921d:isdn_from_driver_process:QUEUE_EVENT
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921f:PBXa RX <- 0x44030300102A34232A35302A33333330
Jan  8 17:24:43.495:   303030312334343430303031
```

All of the octets following "i=" constitute DPNSS L3 information received from the peer:

```
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921:PBXa RX <- UI(C) dlci=1 cntl=UI nbit=0
        i=0x00102A34232A35302A33333330303030312334343430303031
```

In the INFORMATION TRANSFER state, DLC 1 received a UI(C) frame (event x2) from the peer carrying DPNSS L3 information:

```
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921d:process_rxdata:Frame sent to L2
Jan  8 17:24:43.495:ISDN  Q921d:isdn_from_driver_process:event_count 1
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921d:dpnss_l2_main:source_id x200 event
        x141 v_bit x0 chan x0
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921d:s_dpnss_information_transfer:event x2 chan 1
```

For DLC 1, event information is sent to L3 (IUA BACKHAUL, indicated by dest x300). In this case, DL_DATA_IND (event x241) indicates that some L3 information has been received from the peer.

```
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x300 event x241
        v_bit 1 chan 1 out_pkt x6367175C
```

Information is sent to the driver (dest x200), which is then sent to the peer): An Unnumbered Information—Response [UI(R)] (event x3) acknowledges the received Unnumbered Information—Command [UI(C)].

```
Jan  8 17:24:43.495:ISDN Se2/0:15 Q921d:dpnss_l2_mail:dest x200 event x3
        v_bit 1 chan 1 out_pkt x63F183D4
```

The following is sample output from the **debug isdn q921** command for an outgoing call:

```
Router# debug isdn q921

Jan  3 14:52:24.475: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 5  nr = 2
                              i = 0x08010705040288901801837006803631383835
Jan  3 14:52:24.503: ISDN BR0: RX <-  RRr sapi = 0  tei = 64  nr = 6
Jan  3 14:52:24.527: ISDN BR0: RX <-  INFOc sapi = 0  tei = 64  ns = 2  nr = 6
                              i = 0x08018702180189
Jan  3 14:52:24.535: ISDN BR0: TX ->  RRr sapi = 0  tei = 64  nr = 3
Jan  3 14:52:24.643: ISDN BR0: RX <-  INFOc sapi = 0  tei = 64  ns = 3  nr = 6
                              i = 0x08018707
Jan  3 14:52:24.655: ISDN BR0: TX ->  RRr sapi = 0  tei = 64  nr = 4
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
Jan  3 14:52:24.683: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 6  nr = 4
                              i = 0x0801070F
Jan  3 14:52:24.699: ISDN BR0: RX <-  RRr sapi = 0  tei = 64  nr = 7
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
%ISDN-6-CONNECT: Interface BRI0:1 is now connected to 61885 goodie
Jan  3 14:52:34.415: ISDN BR0: RX <-  RRp sapi = 0  tei = 64 nr = 7
Jan  3 14:52:34.419: ISDN BR0: TX ->  RRf sapi = 0  tei = 64  nr = 4
```

In the following lines, the seventh and eighth most significant hexadecimal numbers indicate the type of message. 0x05 indicates a Call Setup message, 0x02 indicates a Call Proceeding message, 0x07 indicates a Call Connect message, and 0x0F indicates a Connect Ack message.

```
Jan  3 14:52:24.475: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 5  nr = 2
                              i = 0x08010705040288901801837006803631383835
```

```
Jan  3 14:52:24.527: ISDN BR0: RX <-  INFOc sapi = 0  tei = 64  ns = 2  nr = 6
                              i = 0x08018702180189
Jan  3 14:52:24.643: ISDN BR0: RX <-  INFOc sapi = 0  tei = 64  ns = 3  nr = 6
                              i = 0x08018707
Jan  3 14:52:24.683: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 6  nr = 4
                              i = 0x0801070F
```

The following is sample output from the **debug isdn q921** command for a startup message on a DMS-100 switch:

```
Router# debug isdn q921

Jan  3 14:47:28.455: ISDN BR0: RX <-  IDCKRQ  ri = 0  ai = 127 0
Jan  3 14:47:30.171: ISDN BR0: TX ->  IDREQ  ri = 31815  ai = 127
Jan  3 14:47:30.219: ISDN BR0: RX <-  IDASSN  ri = 31815  ai = 64
Jan  3 14:47:30.223: ISDN BR0: TX ->  SABMEp sapi = 0  tei = 64
Jan  3 14:47:30.227: ISDN BR0: RX <-  IDCKRQ  ri = 0  ai = 127
Jan  3 14:47:30.235: ISDN BR0: TX ->  IDCKRP  ri = 16568  ai = 64
Jan  3 14:47:30.239: ISDN BR0: RX <-  UAf sapi = 0  tei = 64
Jan  3 14:47:30.247: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 0  nr = 0
                              i = 0x08007B3A03313233
Jan  3 14:47:30.267: ISDN BR0: RX <-  RRr sapi = 0  tei = 64  nr = 1
Jan  3 14:47:34.243: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 1  nr = 0
                              i = 0x08007B3A03313233
Jan  3 14:47:34.267: ISDN BR0: RX <-  RRr sapi = 0  tei = 64  nr = 2
Jan  3 14:47:43.815: ISDN BR0: RX <-  RRp sapi = 0  tei = 64 nr = 2
Jan  3 14:47:43.819: ISDN BR0: TX ->  RRf sapi = 0  tei = 64  nr = 0
Jan  3 14:47:53.819: ISDN BR0: TX ->  RRp sapi = 0  tei = 64 nr = 0
```

The first seven lines of this example indicate a Layer 2 link establishment.

The following lines indicate the message exchanges between the data link layer entity on the local router (user side) and the assignment source point (ASP) on the network side during the TEI assignment procedure. This assumes that the link is down and no TEI currently exists.

```
Jan  3 14:47:30.171: ISDN BR0: TX ->  IDREQ  ri = 31815  ai = 127
Jan  3 14:47:30.219: ISDN BR0: RX <-  IDASSN  ri = 31815  ai = 64
```

At 14:47:30.171, the local router data link layer entity sent an Identity Request message to the network data link layer entity to request a TEI value that can be used in subsequent communication between the peer data link layer entities. The request includes a randomly generated reference number (31815) to differentiate among user devices that request automatic TEI assignment and an action indicator of 127 to indicate that the ASP can assign any TEI value available. The ISDN user interface on the router uses automatic TEI assignment.

At 14:47:30.219, the network data link entity responds to the Identity Request message with an Identity Assigned message. The response includes the reference number (31815) previously sent in the request and TEI value (64) assigned by the ASP.

The following lines indicate the message exchanges between the layer management entity on the network and the layer management entity on the local router (user side) during the TEI check procedure:

```
Jan  3 14:47:30.227: ISDN BR0: RX <-  IDCKRQ  ri = 0  ai = 127
Jan  3 14:47:30.235: ISDN BR0: TX ->  IDCKRP  ri = 16568  ai = 64
```

At 14:47:30.227, the layer management entity on the network sends the Identity Check Request message to the layer management entity on the local router to check whether a TEI is in use. The message includes a reference number that is always 0 and the TEI value to check. In this case, an ai value of 127 indicates that all TEI values should be checked. At 14:47:30.227, the layer management entity on the local router responds with an Identity Check Response message indicating that TEI value 64 is currently in use.

The following lines indicate the messages exchanged between the data link layer entity on the local router (user side) and the data link layer on the network side to place the network side into modulo 128 multiple frame acknowledged operation. Note that the data link layer entity on the network side also can initiate the exchange.

```
Jan  3 14:47:30.223: ISDN BR0: TX ->  SABMEp sapi = 0  tei = 64
Jan  3 14:47:30.239: ISDN BR0: RX <-  UAf sapi = 0  tei = 64
```

At 14:47:30.223, the data link layer entity on the local router sends the SABME command with a SAPI of 0 (call control procedure) for TEI 64. At 14:47:30.239, the first opportunity, the data link layer entity on the network responds with a UA response. This response indicates acceptance of the command. The data link layer entity sending the SABME command may need to send it more than once before receiving a UA response.

The following lines indicate the status of the data link layer entities. Both are ready to receive I frames.

```
Jan  3 14:47:43.815: ISDN BR0: RX <-  RRp sapi = 0  tei = 64 nr = 2
Jan  3 14:47:43.819: ISDN BR0: TX ->  RRf sapi = 0  tei = 64  nr = 0
```

These I-frames are typically exchanged every 10 seconds (T203 timer).

The following is sample output from the **debug isdn q921** command for an incoming call. It is an incoming SETUP message that assumes that the Layer 2 link is already established to the other side.

```
Router# debug isdn q921

Jan  3 14:49:22.507: ISDN BR0: TX ->  RRp sapi = 0  tei = 64 nr = 0
Jan  3 14:49:22.523: ISDN BR0: RX <-  RRf sapi = 0  tei = 64  nr = 2
Jan  3 14:49:32.527: ISDN BR0: TX ->  RRp sapi = 0  tei = 64 nr = 0
Jan  3 14:49:32.543: ISDN BR0: RX <-  RRf sapi = 0  tei = 64  nr = 2
Jan  3 14:49:42.067: ISDN BR0: RX <-  RRp sapi = 0  tei = 64 nr = 2
Jan  3 14:49:42.071: ISDN BR0: TX ->  RRf sapi = 0  tei = 64  nr = 0
Jan  3 14:49:47.307: ISDN BR0: RX <-  UI sapi = 0  tei = 127
                              i = 0x08011F0504028890180189700 6C13631383836
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
Jan  3 14:49:47.347: ISDN BR0: TX ->  INFOc sapi = 0  tei = 64  ns = 2  nr = 0
                              i = 0x08019F07180189
Jan  3 14:49:47.367: ISDN BR0: RX <-  RRr sapi = 0  tei = 64  nr = 3
Jan  3 14:49:47.383: ISDN BR0: RX <-  INFOc sapi = 0  tei = 64  ns = 0  nr = 3
                              i = 0x08011F0F180189
Jan  3 14:49:47.391: ISDN BR0: TX ->  RRr sapi = 0  tei = 64  nr = 1
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
```

Table 1 describes the significant fields shown in the display.

Table 1 debug isdn q921 Field Descriptions

Field	Description
Jan 3 14:49:47.391	Indicates the date and time at which the frame was sent from or received by the data link layer entity on the router. The time is maintained by an internal clock.
TX	Indicates that this frame is being sent from the ISDN interface on the local router (user side).
RX	Indicates that this frame is being received by the ISDN interface on the local router from the peer (network side).
IDREQ	Indicates the Identity Request message type sent from the local router to the network (ASP) during the automatic TEI assignment procedure. This message is sent in a UI command frame. The SAPI value for this message type is always 63 (indicating that it is a Layer 2 management procedure) but it is not displayed. The TEI value for this message type is 127 (indicating that it is a broadcast operation).
ri = 31815	Indicates the Reference number used to differentiate between user devices requesting TEI assignment. This value is a randomly generated number from 0 to 65535. The same ri value sent in the IDREQ message should be returned in the corresponding IDASSN message. Note that a Reference number of 0 indicates that the message is sent from the network side management layer entity and a reference number has not been generated.
ai = 127	Indicates the Action indicator used to request that the ASP assign any TEI value. It is always 127 for the broadcast TEI. Note that in some message types, such as IDREM, a specific TEI value is indicated.
IDREM	Indicates the Identity Remove message type sent from the ASP to the user side layer management entity during the TEI removal procedure. This message is sent in a UI command frame. The message includes a reference number that is always 0, because it is not responding to a request from the local router. The ASP sends the Identity Remove message twice to avoid message loss.
IDASSN	Indicates the Identity Assigned message type sent from the ISDN service provider on the network to the local router during the automatic TEI assignment procedure. This message is sent in a UI command frame. The SAPI value for this message type is always 63 (indicating that it is a Layer 2 management procedure). The TEI value for this message type is 127 (indicating it is a broadcast operation).
ai = 64	Indicates the TEI value automatically assigned by the ASP. This TEI value is used by data link layer entities on the local router in subsequent communication with the network. The valid values are in the range from 64 to 126.

Table 1 debug isdn q921 Field Descriptions (continued)

Field	Description
SABME	Indicates the set asynchronous balanced mode extended command. This command places the recipient into modulo 128 multiple frame acknowledged operation. This command also indicates that all exception conditions have been cleared. The SABME command is sent once a second for N200 times (typically three times) until its acceptance is confirmed with a UA response. For a list and brief description of other commands and responses that can be exchanged between the data link layer entities on the local router and the network, see ITU-T Recommendation Q.921.
sapi = 0	Identifies the service access point at which the data link layer entity provides services to Layer 3 or to the management layer. A SAPI with the value 0 indicates it is a call control procedure. Note that the Layer 2 management procedures such as TEI assignment, TEI removal, and TEI checking, which are tracked with the **debug isdn q921** command, do not display the corresponding SAPI value; it is implicit. If the SAPI value were displayed, it would be 63.
tei = 64	Indicates the TEI value automatically assigned by the ASP. This TEI value will be used by data link layer entities on the local router in subsequent communication with the network. The valid values are in the range from 64 to 126.
IDCKRQ	Indicates the Identity Check Request message type sent from the ISDN service provider on the network to the local router during the TEI check procedure. This message is sent in a UI command frame. The ri field is always 0. The ai field for this message contains either a specific TEI value for the local router to check or 127, which indicates that the local router should check all TEI values. For a list and brief description of other message types that can be exchanged between the local router and the ISDN service provider on the network, see Appendix B, “ISDN Switch Types, Codes, and Values.”
IDCKRP	Indicates the Identity Check Response message type sent from the local router to the ISDN service provider on the network during the TEI check procedure. This message is sent in a UI command frame in response to the IDCKRQ message. The ri field is a randomly generated number from 0 to 65535. The ai field for this message contains the specific TEI value that has been checked.
UAf	Confirms that the network side has accepted the SABME command previously sent by the local router. The final bit is set to 1.
INFOc	Indicates that this is an Information command. It is used to transfer sequentially numbered frames containing information fields that are provided by Layer 3. The information is transferred across a data-link connection.
INFORMATION pd = 8 callref = (null)	Indicates the information fields provided by Layer 3. The information is sent one frame at a time. If multiple frames need to be sent, several Information commands are sent. The pd value is the protocol discriminator. The value 8 indicates it is call control information. The call reference number is always null for SPID information.

Table 1 debug isdn q921 Field Descriptions (continued)

Field	Description
SPID information i = 0x343135393033383336363031	Indicates the SPID. The local router sends this information to the ISDN switch to indicate the services to which it subscribes. SPIDs are assigned by the service provider and are usually 10-digit telephone numbers followed by optional numbers. Currently, only the DMS-100 switch supports SPIDs, one for each B channel. If SPID information is sent to a switch type other than DMS-100, an error may be displayed in the debug information.
ns = 0	Indicates the send sequence number of sent I frames.
nr = 0	Indicates the expected send sequence number of the next received I frame. At time of transmission, this value should be equal to the value of ns. The value of nr is used to determine whether frames need to be re-sent for recovery.
RRr	Indicates the Receive Ready response for unacknowledged information transfer. The RRr is a response to an INFOc.
RRp	Indicates the Receive Ready command with the poll bit set. The data link layer entity on the user side uses the poll bit in the frame to solicit a response from the peer on the network side.
RRf	Indicates the Receive Ready response with the final bit set. The data link layer entity on the network side uses the final bit in the frame to indicate a response to the poll.
sapi	Indicates the service access point identifier. The SAPI is the point at which data link services are provided to a network layer or management entity. Currently, this field can have the value 0 (for call control procedure) or 63 (for Layer 2 management procedures).
tei	Indicates the terminal endpoint identifier (TEI) that has been assigned automatically by the assignment source point (ASP) (also called the layer management entity on the network side). The valid range is from 64 to 126. The value 127 indicates a broadcast.

Related Commands

Command	Description
service timestamps debug datetime msec	Includes the time with each debug message.
debug isdn event	Displays ISDN events occurring on the user side (on the router) of the ISDN interface.
debug isdn q931	Displays information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network.

isdn bind-l3 iua-backhaul

To specify ISDN backhaul using Stream Control Transmission Protocol (SCTP) for an interface or to bind Layer 3 to DUA for DPNSS backhaul, use the **isdn bind-l3 iua-backhaul** command in interface configuration mode. To disable the backhaul capability, use the **no** form of this command.

isdn bind-l3 [*application-server-name*]

no isdn bind-l3

Syntax Description

application-server-name	(Optional) Name of the application server (AS) to use for backhauling the interface.

Defaults

No default behavior or values

Command Modes

Interface configuration

Command History

Release	Modification
12.1(1)T	This command was introduced on the Cisco AS5300.
12.2(4)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco MC3810.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 7200 series. Support for the Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco IAD2420 series. The Cisco AS5850 is not included in this release.
12.2(11)T	This command was implemented on the following platforms: Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(15)ZJ	The capability to bind Layer 3 to DUA for DPNSS backhaul was added.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

DPNSS is not configured for backhaul and is not associated with a session set.

Examples

The following example configures DUA for DPNSS backhaul using an AS called “as1:”

```
Router(config-if)# isdn bind-l3 iua-backhaul as1
```

The following example configures T1 signaling channel serial 0:23 for signaling backhaul and associates the D channel with the session set named “set1”:

```
Router(config)# interface s0:23
Router(config-if)# isdn bind-l3 set1
```

The following example configures E1 signaling channel serial 0:15 for signaling backhaul and associates the D channel with the session set named “set3”:

```
Router(config)# interface s0:15
Router(config-if)# isdn bind-l3 set3
```

Related Commands

Command	Description
as	Defines an AS for backhaul.
asp	Defines an ASP for backhaul.

isdn dpnss

To indicate whether ISDN DPNSS is to act as PBX A or PBX B, or revert to Layer 2, use the **isdn dpnss** command in interface configuration mode. To reset to the default, use the **no** form of this command.

isdn dpnss [**pbxA** | **layer 2** [**retry max-count** *range*] [**timers** [**Tretry** *timer-value*] [**Ttest** *timer-value*]] [**test frame**]]

no isdn dpnss [**pbxA** | **layer 2** [**retry max-count** *range*] [**timers** [**Tretry** *timer-value*] [**Ttest** *timer-value*]] [**test frame**]]

Syntax Description

pbxA	(Optional) Enables DPNSS to act as PBX A.
layer 2	(Optional) Reverts to Layer 2.
retry max-count *range*	(Optional) Selects the number of times a frame will be retried if unacknowledged. The max-count value can be any number from 0 to 64. Default is 4
timers	(Optional) Selects DPNSS timers, which can be **Tretry** or **Ttest**.
Tretry *timer-value*	(Optional) Sets the Tretry timer in ms and seconds. Valid retry time values range from 5 ms to 10 seconds. Default is 500 ms.
Ttest *timer-value*	(Optional) Sets the Ttest timer in minutes. When the Ttest timer expires, frames are sent on all the DLCs. Valid test time values range from 1 to 60. Default is 5.
test frame	(Optional) Allows test frames to be sent periodically.

Defaults

PBX B

Command Modes

Interface configuration

Command History

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

Examples

The following example sets ISDN DPNSS to act as PBX A:

```
Router(config-if)# isdn dpnss pbxA
```

The following example sets the Tretry and Ttest timers:

```
Router(config-if)# isdn dpnss layer2 timers Tretry 500 Ttest 5
```

The following example selects the number of times a frame will be retried if unacknowledged:

```
Router(config-if)# isdn dpnss layer2 retry max-count 4
```

The following example allows test frames to be sent periodically:

```
Router(config-if)# isdn dpnss layer2 test frame
```

Related Commands

Command	Description
isdn bind-l3 iua-backhaul	Binds Layer 3 for DPNSS to DUA.
isdn switch-type (PRI)	Specifies the central office switch type on the ISDN interface.

isdn switch-type (PRI)

To specify the central office switch type on the ISDN interface, or to configure the Cisco MC3810 PRI interface to support Q Signaling (QSIG), use the **isdn switch-type** command in global or interface configuration mode. To disable the switch or QSIG signaling on the ISDN interface, use the **no** form of this command.

isdn switch-type *switch-type*

no isdn switch-type

Syntax Description

switch-type	Service provider switch type; see Table 2 for a list of supported switches and the associated keywords.

Defaults

No switch type is defined (the switch is disabled on the ISDN interface)

Command Modes

Global configuration
Interface configuration

Command History

Release	Modification
9.21	This command was introduced as a global configuration mode command.
11.3 T	This command was introduced as an interface configuration mode command.
12.0(2)T	The **primary-qsig-slave** and **primary-qsig master** switch type options were added to support PRI QSIG signaling.
12.2(15)ZJ	The **primary-dpnss** switch type option was added for Digital Private Network Signaling System (DPNSS) backhaul.
12.3(4)T	Support for the **primary-dpnss** switch type option was added to Cisco IOS Release 12.3(4)T.

Usage Guidelines

You have a choice of configuring the **isdn-switch-type** command to support QSIG at either the global configuration level or at the interface configuration level. For example, if you have a QSIG connection on one line and on the BRI port, you can configure the ISDN switch type in one of the following combinations:

- Set the global **isdn-switch-type** command to support QSIG, and set the interface **isdn-switch-type** command for the **interface bri 0** command to a BRI setting such as 5ess.
- Set the global **isdn-switch-type** command to support BRI 5ess, and set the interface **isdn-switch-type** command for the **interface serial 1:23** command to support QSIG.
- Configure the global **isdn-switch-type** command to another setting (such as switch type VN3), and then set the interface **isdn-switch-type** command for the **interface bri 0** command to a BRI setting, and set the interface **isdn-switch-type** command for the **interface serial 1:23** command to support QSIG.

Note This command can be entered in either global configuration mode or in interface configuration mode. When entered in global configuration mode, the setting applies to the entire Cisco MC3810. When entered in interface configuration mode, the setting applies only to the T1/E1 interface specified. The interface configuration mode setting overrides the global configuration setting.

Issuing the **no isdn switch-type** command in interface configuration mode results in the configured global switch-type being applied to the specified interface. If no global switch-type is defined, issuing the **no isdn switch-type** command at the interface level disables the switch.

Issuing the **no isdn switch-type** command in global configuration mode prevents the configuration of ISDN PRI and results in the following warning message:

```
Router(config)# no isdn switch-type

Warning:No ISDN switch-type defined.  No calls possible, unless switchtype defined
Globally and/or per interface. This change will take full effect upon reload.
```

Note The **codec** (dial-peer) command must be configured before any calls can be placed over the connection to the private integrated services network exchange (PINX). The default codec type is G729a.

Note If you are using the Multiple ISDN Switch Types feature to apply the ISDN switch types to different interfaces, refer to the chapter “Setting Up Basic ISDN Service” in the *Cisco IOS Dial Technologies Configuration Guide* for additional details.

For more information about DPNSS backhaul, refer to the *Digital Private Network Signaling System Backhaul* feature document on Cisco.com.

Table 2 lists the *switch-type* keywords for supported PRI switch types by geographic area.

Table 2 ISDN Service Provider PRI Switch Types

Keywords by Area	Switch Type
Voice/PBX Systems	
primary-qsig-slave	Supports QSIG signaling per Q.931. Network-side functionality is assigned with the **isdn-protocol emulate** command. Specifies the Cisco MC3810 or the interface to act as the primary QSIG slave when the PINX is the primary QSIG master.
primary-qsig-master	Supports QSIG signaling per Q.931. Network-side functionality is assigned with the **isdn-protocol emulate** command. Specifies the Cisco MC3810 or the interface to act as the primary QSIG master when the PINX is the primary QSIG slave.
primary-dpnss	Sets the ISDN switch type to DPNSS.
Australia and Europe	
primary-net5	Specifies NET5 ISDN PRI switch types for Asia, Australia, and New Zealand; European Telecommunications Standards Institute (ETSI)-compliant switches for Euro-ISDN E-DSS1 signaling system.

Table 2 ISDN Service Provider PRI Switch Types

Japan	
primary-ntt	Specifies Japanese ISDN PRI switch.
North America	
primary-4ess	Specifies AT&T 4ESS switch type for the United States.
primary-5ess	Specifies AT&T 5ESS switch type for the United States.
primary-dms100	Specifies NT DMS-100 switch type for the United States.
primary-ni	Specifies National ISDN switch type.

Examples

The following example demonstrates the Multiple ISDN Switch Types feature. The global ISDN switch type setting is basic-net3. The PRI interface (channelized T1 controller) is configured with the **isdn switch-type (PRI)** command using the **primary-net5** keyword, and BRI interface 0 is configured with the **isdn switch-type (BRI)** command using the **basic-ni** keyword (formerly the **basic-ni1** keyword).

```
isdn switch-type basic-net3
!
interface serial0:23
 isdn switch-type primary-net5
 ip address 172.21.24.85 255.255.255.0
!
interface BRI0
 isdn switch-type basic-ni
```

The following example configures T1 interface 23 on the Cisco MC3810 to act as the QSIG master:

```
interface serial 1:23
 isdn switch-type primary-qsig-master
```

The following example sets the switch type to DPNSS:

```
isdn switch-type primary-dpnss
```

Related Commands

Command	Description
codec	Specifies the voice coder rate of speech for a dial peer.
interface	Configures an interface type and enters interface configuration mode.
interface bri	Configures a BRI interface and enters interface configuration mode.
isdn protocol-emulate (dial)	Configures the Layer 2 and Layer 3 port protocol of a BRI voice port or a PRI interface to emulate NT (network) or TE (user) functionality.
isdn switch-type (BRI)	Specifies the central office switch type on the ISDN interface.
pri-group nec-fusion	Configures the NEC PBX to support FCCS.
show cdapi	Displays the CDAPI.
show rawmsg	Displays the raw messages owned by the required component.

show isdn

To display information about memory, Layer 2 and Layer 3 timers, and the status of PRI channels, use the **show isdn** command in privileged EXEC mode.

show isdn {**active** [*dsl* | *serial-number*] | **history** [*dsl* | *serial-number*] | **memory** | **service** [*dsl* | *serial-number*] | **status** [*dsl* | *serial-number*] | **timers** [*dsl* | *serial-number*]}

Syntax Description

active [*dsl* | *serial-number*]	Displays current call information of all ISDN interfaces or, optionally, a specific digital subscriber line (DSL) or a specific ISDN PRI interface (created and configured as a serial interface). Values of *dsl* range from 0 to 15. Information displayed includes the called number, the remote node name, the seconds of connect time, the seconds of connect time remaining, the seconds idle, and Advice of Charge (AOC) charging time units used during the call.
history [*dsl* | *serial-number*]	Displays historic and current call information of all ISDN interfaces or, optionally, a specific DSL or a specific ISDN PRI interface (created and configured as a serial interface). Values of *dsl* range from 0 to 15. Information displayed includes the called number, the remote node name, the seconds of connect time, the seconds of connect time remaining, the seconds idle, and AOC charging time units used during the call.
memory	Displays ISDN memory pool statistics. This keyword is for use by technical development staff only.
service [*dsl* | *serial-number*]	Displays the service status of all ISDN interfaces or, optionally, a specific DSL or a specific ISDN PRI interface (created and configured as a serial interface). Values of *dsl* range from 0 to 15.
status [*dsl* | *serial-number*]	Displays the following status: • All ISDN interfaces or, optionally, a specific DSL or a specific ISDN PRI interface (created and configured as a serial interface). Values of *dsl* range from 0 to 15. • Displays information about 60 Digital Private Network Signaling Systems (DPNSS) channels (30 real plus 30 virtual).
timers [*dsl* | *serial-number*]	Displays the values of Layer 2 and Layer 3 timers for all ISDN interfaces or, optionally, a specific DSL or a specific ISDN PRI interface (created and configured as a serial interface). Values of *dsl* range from 0 to 15.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1	This command was introduced.
12.2(15)ZJ	This command was modified to include information about DPNSS channels when the **status** keyword is used.

Release	Modification
12.2(15)ZJ and 12.3(4)T	This enhanced command was integrated into Cisco IOS Release 12.2(15)T, and implemented on the Cisco 2420, Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers, and Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco AS5850 network access server (NAS) platforms.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

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

```
Router# show isdn active

-------------------------------------------------------------------------------
                              ISDN ACTIVE CALLS
-------------------------------------------------------------------------------
History Table MaxLength = 100 entries
History Retain Timer = 15 Minutes
-------------------------------------------------------------------------------
Call Calling and Called  Remote Node  Seconds  Seconds  Seconds Recorded Charges
Type Phone Number        Name         Used     Left     Idle    Units/Currency
-------------------------------------------------------------------------------
In +---Not Available----    aerocore   684802  +499598     401
In +---Not Available----         pmg   363578  +499503     496
In +---Not Available----      solpro   253232  +499325     674
In +---Not Available----               194047  +499965      34
In +---Not Available----       taber   189165  +499841     158
In +---Not Available----        newt   110342                0
In +---Not Available----                 2603  +499997       2
In +---Not Available----                 1310  +499798     201

-------------------------------------------------------------------------------
```

The following is sample output from the **show isdn history** command:

```
Router# show isdn history

-------------------------------------------------------------------------------
                              ISDN CALL HISTORY
-------------------------------------------------------------------------------
History Table MaxLength = 100 entries
History Retain Timer = 15 Minutes
-------------------------------------------------------------------------------
Call Calling and Called  Remote Node  Seconds  Seconds  Seconds Recorded Charges
Type Phone Number        Name         Used     Left     Idle    Units/Currency
-------------------------------------------------------------------------------
In +---Not Available----    aerocore   684818  +499583     416
In +---Not Available----         pmg   363593  +499488     511
In +---Not Available----      solpro   253248  +499310     689
In +---Not Available----               194062  +499950      49
In +---Not Available----       taber   189180  +499826     173
In +---Not Available----        newt   110357                0
In +---Not Available          a45968     5244
In +---Not Available----                 2619  +499997       0
In +---Not Available----       zetta     1432
In +---Not Available----                 1325  +499783     216
In +---Not Available----         trf      161

-------------------------------------------------------------------------------
```

Table 3 describes the fields in the **show isdn active** and **show isdn history** output displays.

Table 3 show isdn active and show isdn history Field Descriptions

Field	Description
History Table MaxLength	Maximum number of entries that can be retained in the Call History table.
History Retain Timer	Maximum amount of time any entry can be retained in the Call History table.
Call Type	Type of call: In for incoming, Out for outgoing, or -- when direction of call cannot be determined.
Calling and Called Phone Number	For incoming calls, the number from which the call was received. For outgoing calls, the number to which the call was placed, or +---Not Available---- when a phone number is not available. The phone number display is limited to 20 digits. (+---Not Available---- is the truncated version of ----Not Available----. The + in the field means more data is available than can be displayed. The low-order data is displayed, and the overflowing data is replaced by a +.)
Remote Node Name	Name of the host placing the call or the host called. The name display is limited to ten characters.
Seconds Used	Six digits (999999) of seconds showing connect time used, or Failed when the connection attempt fails.
Seconds Left	Six digits (999999) of seconds of connect time remaining (when configured through the **dialer idle-timeout** command. The + in the field means more data is available than can be displayed. The low-order data is displayed and the overflowing data is replaced by a +.)
Seconds Idle	Six digits (999999) of seconds since the last interesting packet.
Recorded Charges Units/Currency	For outgoing calls, number of ISDN AOC charging units used or the currency cost of the call. Currency information display is limited to ten characters.

The following is sample output from the **show isdn service** command when PRI is configured on a T1 controller:

```
Router# show isdn service

PRI Channel Statistics:
ISDN Se0:15, Channel (1-31)
  Activated dsl 8
  State (0=Idle 1=Propose 2=Busy 3=Reserved 4=Restart 5=Maint)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Channel (1-31) Service (0=Inservice 1=Maint 2=Outofservice)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Table 4 describes the significant fields shown in the **show isdn service** display.

Table 4 show isdn service Field Descriptions

Field	Description
ISDN Se1/0:15	ISDN PRI interface corresponding to serial interface 1/0:15.
Channel (1-31)	Channel range "1-31" is a standard format for both T1 and E1 outputs, but the state value shown identifies whether the channel is used.
Activated dsl 8	The DSL value.
State (0=Idle 1=Propose 2=Busy 3=Reserved 4=Restart 5=Maint)	Current state of each channel. Channels 24 through 31 are marked as reserved when the output is from T1.
Channel (1-31) Service (0=Inservice 1=Maint 2=Outofservice)	Service state assigned to each channel. Channel 24 is marked as out of service.[1]

1. If channel 24 (marked as out of service) is configured as the NFAS primary D channel, NFAS will roll over to the backup D channel if one is configured. If channel 24 is a B channel, it will not accept calls.

The following is sample output from the **show isdn status** command. Each DLC channel state is indicated. There are five states in which a DLC can be present: OUT OF SERVICE, IDLE, RESET ATTEMPTED, RESET COMPLETE, INFORMATION TRANSFER. These are standard DLC states.

```
Router# show isdn status

Global ISDN Switchtype = basic-5ess
        dsl 1, interface ISDN Switchtype = primary-dpnss
        L2 Protocol = Q.921  L3 Protocol(s) = IUA BACKHAUL
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        Channel 1 :INFORMATION TRANSFER,
        Channel 2 :RESET COMPLETE,
        Channel 3 :INFORMATION TRANSFER,
        Channel 4 :RESET COMPLETE,
        Channel 5 :INFORMATION TRANSFER,
        Channel 6 :RESET COMPLETE,
        Channel 7 :INFORMATION TRANSFER,
        Channel 8 :RESET COMPLETE,
        Channel 9 :INFORMATION TRANSFER,
        Channel 10:RESET COMPLETE,
        Channel 11:INFORMATION TRANSFER,
        Channel 12:RESET COMPLETE,
        Channel 13:INFORMATION TRANSFER,
        Channel 14:RESET COMPLETE,
        Channel 15:INFORMATION TRANSFER,
        Channel 17:INFORMATION TRANSFER,
        Channel 18:RESET COMPLETE,
        Channel 19:INFORMATION TRANSFER,
        Channel 20:RESET COMPLETE,
        Channel 21:INFORMATION TRANSFER,
        Channel 22:RESET COMPLETE,
        Channel 23:INFORMATION TRANSFER,
        Channel 24:RESET COMPLETE,
        Channel 25:INFORMATION TRANSFER,
        Channel 26:RESET COMPLETE,
        Channel 27:INFORMATION TRANSFER,
        Channel 28:RESET COMPLETE,
        Channel 29:INFORMATION TRANSFER,
```

```
        Channel 30:RESET COMPLETE,
        Channel 31:INFORMATION TRANSFER,
        Channel 33:INFORMATION TRANSFER,
        Channel 34:RESET COMPLETE,
        Channel 35:INFORMATION TRANSFER,
        Channel 36:RESET COMPLETE,
        Channel 37:INFORMATION TRANSFER,
        Channel 38:RESET COMPLETE,
        Channel 39:INFORMATION TRANSFER,
        Channel 40:RESET COMPLETE,
        Channel 41:INFORMATION TRANSFER,
        Channel 42:RESET COMPLETE,
        Channel 43:INFORMATION TRANSFER,
        Channel 44:RESET COMPLETE,
        Channel 45:INFORMATION TRANSFER,
        Channel 46:RESET COMPLETE,
        Channel 47:INFORMATION TRANSFER,
        Channel 49:INFORMATION TRANSFER,
        Channel 50:RESET COMPLETE,
        Channel 51:INFORMATION TRANSFER,
        Channel 52:RESET COMPLETE,
        Channel 53:INFORMATION TRANSFER,
        Channel 54:RESET COMPLETE,
        Channel 55:INFORMATION TRANSFER,
        Channel 56:RESET COMPLETE,
        Channel 57:INFORMATION TRANSFER,
        Channel 58:RESET COMPLETE,
        Channel 59:INFORMATION TRANSFER,
        Channel 60:RESET COMPLETE,
        Channel 61:INFORMATION TRANSFER,
        Channel 62:RESET COMPLETE,
        Channel 63:INFORMATION TRANSFER,
```

The following is sample output from the **show isdn status** command with one active call:

```
Router# show isdn status

The current ISDN Switchtype = ntt
ISDN BRI0 interface
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 64, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status:
        1 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 1
        CCB:callid=8003, callref=0, sapi=0, ces=1, B-chan=1
    Number of active calls = 1
    Number of available B-channels = 1
    Total Allocated ISDN CCBs = 1
```

Table 5 describes the significant fields shown in the **show isdn status** display.

Table 5 show isdn status Field Descriptions

Field	Description
Layer 1 Status	
ACTIVE	Status of ISDN Layer 1.
Layer 2 Status	

Table 5 show isdn status Field Descriptions (continued)

Field	Description
TEI = 64, State = MULTIPLE_FRAME_ESTABLISHED	Status of ISDN Layer 2. Terminal endpoint identifier number and multiframe structure state.
Layer 3 Status	
1 Active Layer 3 Call(s)	Number of active calls.
Activated dsl 0 CCBs =	Number of the DSL activated. Number of call control blocks in use.
CCB:callid=8003, callref=0, sapi=0, ces=1, B-chan=1	Information about the active call.
Number of active calls =	Number of active calls.
Number of available B-channels =	Number of B channels that are not being used.
Total Allocated ISDN CCBs =	Number of ISDN call control blocks that are allocated.

The following is sample output from the **show isdn timers** command:

```
Router# show isdn timers

ISDN Serial0:23 Timers (dsl 0) Switchtype = primary-5ess
        ISDN Layer 2 values
         K     =   7 outstanding I-frames
         N200  =   3 max number of retransmits
         T200  =   1.000 seconds
         T202  =   2.000 seconds
         T203  =  30.000 seconds
        ISDN Layer 3 values
         T303  =   4.000 seconds
         T304  =  20.000 seconds
         T305  =   4.000 seconds
         T306 =  30.000 seconds
         T307 = 180.000 seconds
         T308  =   4.000 seconds
         T309    Disabled
         T310 =  30.000 seconds
         T313 =   4.000 seconds
         T316  = 120.000 seconds
         T318  =   4.000 seconds
         T319  =   4.000 seconds
         T322  =   4.000 seconds
         T3OOS =   5.000 seconds
         TGUARD=   8.000 seconds, Expiry = REJECT_CALL
ISDN Serial1:23 Timers (dsl 1) Switchtype = primary-5ess
        ISDN Layer 2 values
         K     =   7 outstanding I-frames
         N200  =   3 max number of retransmits
         T200  =   1.000 seconds
         T202  =   2.000 seconds
         T203  =  30.000 seconds
        ISDN Layer 3 values
         T303  =   4.000 seconds
         T304  =  20.000 seconds
         T305  =   4.000 seconds
         T306 =  30.000 seconds
         T307 = 180.000 seconds
         T308  =   4.000 seconds
```

```
          T309    Disabled
          T310 =  30.000 seconds
          T313 =   4.000 seconds
          T316  = 120.000 seconds
          T318  =   4.000 seconds
          T319  =   4.000 seconds
          T322  =   4.000 seconds
          T3OOS =   5.000 seconds
          TGUARD=   8.000 seconds, Expiry = REJECT_CALL
         *** dsl 2 is not configured
         *** dsl 3 is not configured
         *** dsl 4 is not configured
         *** dsl 5 is not configured
         *** dsl 6 is not configured
         *** dsl 7 is not configured
ISDN BRI0 Timers (dsl 0) Switchtype = basic-net3
         ISDN Layer 2 values
          K    = 1   outstanding I-frames
          N200 = 3   max number of retransmits
          N202 = 2   max number of retransmits of TEI ID Request
          T200 = 1   seconds
          T202 = 2   seconds
          T203 = 10  seconds
         ISDN Layer 3 values
          T303 = 4   seconds
          T305 = 30   seconds
          T308 = 4   seconds
          T310 = 40  seconds
          T313 = 4   seconds
          T316 = 0   seconds
          T318 = 4   seconds
          T319 = 4   seconds
```

Table 6 and Table 7 show typical and default values of the timers shown in the **show isdn timers** display. The values of the timers depend on the switch type. The Cisco routers support the following switch type keywords: **basic-ni**, **basic-net3**, **primary-5ess**, and **basic-qsig** and **primary-qsig**. Refer to the Q.921 specifications for detailed technical definitions of the Layer 2 timers; refer to the Q.931 specifications for detailed technical definitions of the Layer 3 timers.

Table 6 show isdn timers Layer 2 Command Output

Timer Number Field	System Parameter (Typical)
K = 7 outstanding I-frames	None
N200 = 3 max number of retransmits	3 seconds
T200 = 1.000 seconds	1 second
T202 = 2.000 seconds	2 seconds
T203 = 30.000 seconds	10 seconds

Table 7 show isdn timers Layer 3 Command Output

Timer Number Field	Network Side ITU Default Value	User Side ITU Default Value
T303 = 4.000 seconds	4 seconds	4 seconds
T304 = 20.000 seconds	20 seconds	30 seconds
T305 = 4.000 seconds	30 seconds	30 seconds

Table 7 show isdn timers Layer 3 Command Output (continued)

Timer Number Field	Network Side ITU Default Value	User Side ITU Default Value
T306 = 30.000 seconds	30 seconds	None
T307 = 180.000 seconds	180 seconds (3 minutes)	None
T308 = 4.000 seconds	4 seconds	4 seconds
T309 Disabled	90 seconds	90 seconds
T310 = 30.000 seconds	10 seconds	30 to 120 seconds
T313 = 4.000 seconds	None	4 seconds
T316 = 120.000 seconds	120 seconds (2 minutes)	120 seconds (2 minutes)
T318 = 4.000 seconds	None	4 seconds
T319 = 4.000 seconds	None	4 seconds
T322 = 4.000 seconds	4 seconds	4 seconds
T3OOS = 5.000 seconds	Time interval after which the software should attempt to recover from a Layer 2 failure. Default is 5 seconds.	Time interval after which the software should attempt to recover from a Layer 2 failure. Default is 5 seconds.
TGUARD = 8.000 seconds, Expiry = REJECT_CALL	Managed timer for authentication requests configured with the **isdn guard-timer** command. Default is 8 seconds.	Managed timer for authentication requests configured with the **isdn guard-timer** command. Default is 8 seconds.

Related Commands

Command	Description
debug isdn q921	Displays Layer 2 access procedures that are taking place at the router on the D channel of its ISDN interface.

Glossary

API—application programming interface.

AS—application server.

ASP—application server process.

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

DLC—data link connection.

DPNSS—Digital Private Network Signaling Systems.

DS0—digital signal level 0.

DUA—DASS User Adaptation.

DUARL—DPNSS User Adaptation Routing Library.

I-frame—information frame (I-frame).

ISDN—Integrated Services Digital Network.

L2—Layer 2.

L3—Layer 3.

LAPD—Link Access Procedure on the D channel. ISDN data link layer protocol for the D channel. LAPD was derived from the LAPB protocol and is designed primarily to satisfy the signaling requirements of ISDN basic access. Defined by ITU-T Recommendations Q.920 and Q.921.

MGCP—Media Gateway Control Protocol.

PBX—Public Branch Exchange.

PDU—protocol data unit.

PGW—PSTN GateWay.

SABMR—Set Asynchronous Balanced Mode Restricted.

SCTP—Stream Control Transmission Protocol.

SRL—Signaling Routing Library.

UA—Unnumbered Acknowledgment.

UI(C)—Unnumbered Information - Command.

UI(R)—Unnumbered Information - Response.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.