DOS Client Driver

for

Digi's DataFire™ and PCIMAC™

Adapters

Installation and Configuration Guide

WWW: http://www.dgii.com

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document "as is", without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in hsi manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Table of Contents

About This Manual

Chapter 1: Introduction

Chapter 2: Driver Installation and Configuration Instructions

Chapter 3: Planning and Preparation for the ODI Driver

Chapter 4: Planning and Preparation for the NDIS Driver

Chapter **B: Troubleshooting and Tech Tips**

Index

About This Manual

In this chapter

This chapter describes the purpose and intent of this manual. It discusses the following topics:

Audience

The audience for this manual is a system administrator with knowledge about the applications (such as NetWare® and Windows®) that will use the adapter through a DOS connection, and who has knowledge about the system in which the adapter will be installed.

Scope

This manual describes how to use the DOS-based installation and configuration utility for installing and configuring the driver, describes how to setup calling profiles, how to make connections, and provides some troubleshooting tips for ensuring proper connections.

Conventions Used In This Manual

Certain conventions are used in this manual with respect to keyboard entry and typefaces:

Keyboard entry

- Keystrokes are enclosed in angle brackets (for example, <Enter>, <Esc>, and so on.) Keys which are pressed while holding down another key are shown enclosed together in angle brackets. For example, <Ctrl-A> means hold down the <Ctrl> key while pressing the <A> key. Similarly, <Alt-A> means hold down the <Alt> key while pressing the <A> key.
- “Arrow keys” refers to the up, down, right and left arrow keys.

Typefaces

- Examples of output to your computer screen are shown in sans-serif characters.
- Commands and data that you are to enter via your keyboard are shown in **bold** sans-serif characters.
- Variable information is shown in italics. For example, if you are asked to provide *file_name*, you would provide the actual name of the file to be acted upon.

Related Documents

You may find the following documents useful as you install and use the DOS Client driver for Digi’s ISDN adapter:

- Hardware manual for the adapter
- Documentation for your network

chapter **1**

Introduction

In this chapter

This chapter introduces the DOS client driver for Digi's ISDN adapters. It discusses the following topics:

Product Description

The Digi DataFire and PC-IMAC ISDN adapters can be thought of as "ISDN Ethernet cards." The network software sees them as Ethernet adapters on a LAN. Instead of Ethernet wiring, Digi's ISDN adapters use an ISDN connection, which can be set up as always connected or only connected when there is network information to transfer.

Installing the device driver software for the Digi adapters is much the same as installing an Ethernet card; there are network accounts, logins and other considerations that you must address according to your own situation. Your own network documentation will be required.

System Requirements

- 80386 microprocessor or better
- MS-DOS or PC-DOS version 3.3 or higher

chapter 2

Driver Installation and Configuration Instructions

In this chapter

This chapter provides step-by-step instructions to help you install the DOS client driver on your system. It discusses the following topics:

Starting Point

- At this point you must have contacted your Service Provider and obtained the following information to be used during installation:
  - Phone number(s)
  - Switch type (software used by switch)
  - SPID(s) (if required by your service provider)
- You must also have installed your Digi ISDN Adapter

  Follow the instructions in the hardware installation guide that came with your adapter and install the Digi ISDN adapter into your system. You will need to know the I/O Base Address of a DataFire adapter during driver configuration, and the I/O Base and Memory Address if you are installing a PC-IMAC adapter.

About the Install Program

To use the Install program effectively, remember these few things:

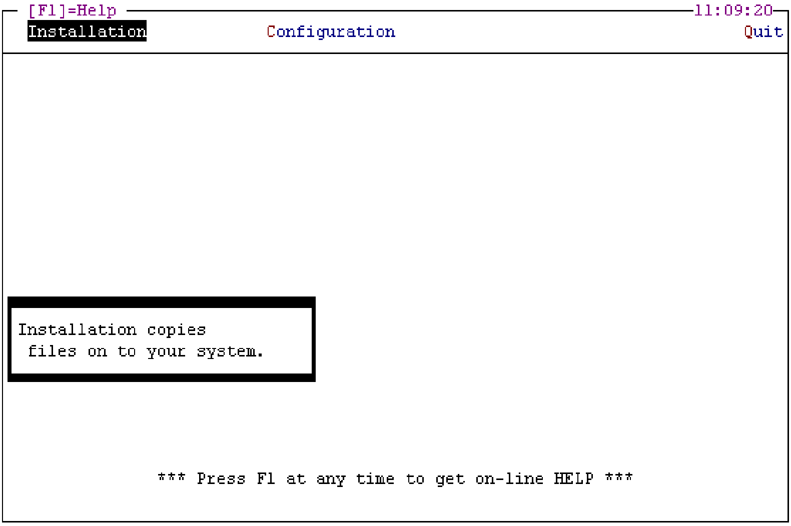
- To save a selection you must press the <F10> key.
- If you are using a monochrome monitor and have trouble reading the screen, press <Alt-Y> to switch to a monochrome screen.
- Press <F1> to view on-line help.
- The default destination directory name is DIGIISDN.

Driver Installation

1. Place the DOS client driver disk in either drive A or B, and switch to that drive from the DOS prompt:

   C:\ **A:** or C:\ **B:**

2. Type INSTALL and press <Enter>. When the program runs, you will see the program's main menu, which will look similar to the figure below:



```
Installation copies
 files on to your system.
```

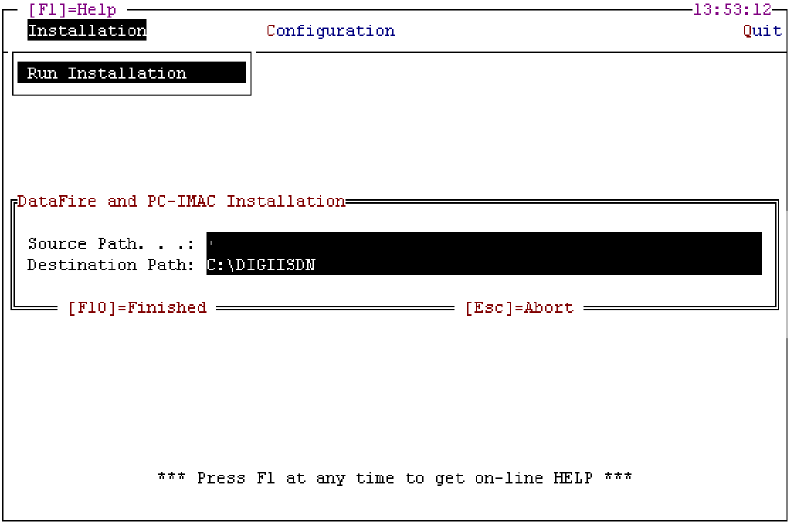
```
*** Press F1 at any time to get on-line HELP ***
```

Figure 2-1. Install Program Main Menu

3. Use the mouse (or the down arrow key) to highlight Installation and then press the <Enter> key. Press <Enter> again to select Run Installation.

4. The figure on page 2-4 illustrates the screen that you will see.The letter of the drive with the Digi installation disk (A or B) will automatically display for the Source Path. The default directory for the Destination Path is C:\DIGIISDN; you may change this if you wish.

   All files necessary to use the DOS client driver will be copied to the directory you name.



```
Source Path. . .:
Destination Path: C:\DIGIISDN
```

```
*** Press F1 at any time to get on-line HELP ***
```

Figure 2-2. Choosing a Destination Path

5. Press <F10> to copy the files to the directory that you selected as the Destination Path.
6. The installation program will ask to modify the PATH statement in your AUTOEXEC.BAT file to include the directory with Digi's driver and utilities.

   If you do not choose to let the program automatically modify the PATH statement, you can modify it yourself. For example:

   PATH= C:\dos;C:\digiisdn

Driver Configuration

Configuration of the driver requires you to select operational parameters for the Digi adapter and also select characteristics of the ISDN line installed in the adapter.

To view the Configuration menu, use the mouse (or arrow keys) to highlight Configuration on the menu bar, then press <Enter>. From the menu, you will perform these general configuration steps:

1. Select an adapter model
2. Select adapter parameters
3. Select line parameters

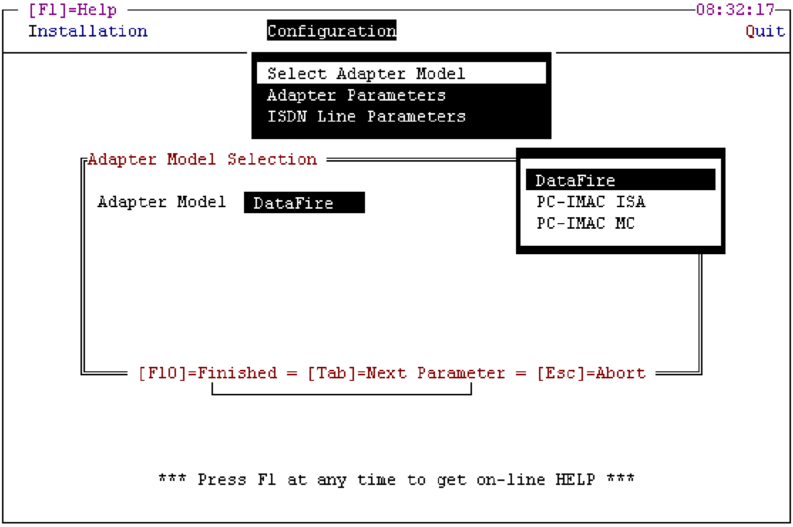
Select Adapter Model

There are three types of Digi ISDN adapters that can be used with this driver, and each has slightly different configuration parameters:

- DataFire
- PC-IMAC (ISA version)
- PC-IMAC (Micro Channel version)

You must select the *type* of adapter you have in your system before you can select parameters that configure it for use with the driver. To select an adapter model:

a. Highlight Select Adapter Model and press <Enter>. You will see a selection box display:



```
Adapter Model Selection

  Adapter Model  DataFire
```

```
*** Press F1 at any time to get on-line HELP ***
```

Figure 2-3. Select Adapter Model

b. Using the arrow keys or your mouse, highlight the Digi adapter model that you have installed in your system.

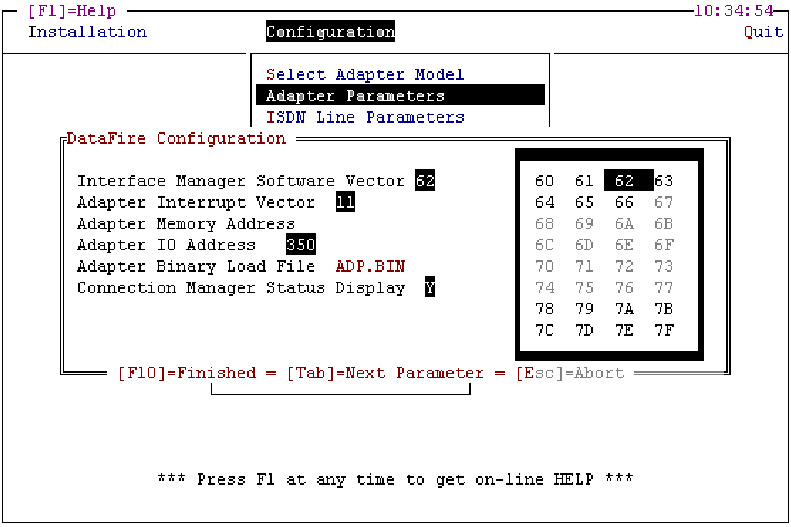
c. Press the <F10> key. A box will display asking if the parameters are correct.

d. Select Yes to save the adapter type.

Select Adapter Parameters

Use the Adapter Parameters menu to setup your Digi ISDN adapter for your system. To setup your adapter:

a. Highlight Adapter Parameters and press <Enter>. You will see a selection box display:



```
Interface Manager Software Vector 62
Adapter Interrupt Vector  11
Adapter Memory Address
Adapter IO Address    350
Adapter Binary Load File  ADP.BIN
Connection Manager Status Display  Y
```

```
60  61  62  63
64  65  66  67
68  69  6A  6B
6C  6D  6E  6F
70  71  72  73
74  75  76  77
78  79  7A  7B
7C  7D  7E  7F
```

Figure 2-4. Adapter Parameters Screen for the DataFire adapter

b. Using the arrow keys or your mouse, highlight the parameters that you want to change.

c. Press the <F10> key. A box will display asking if the parameters are correct.

d. Select *Yes* to save the adapter parameters.

Depending on the type of adapter you chose, you will be able to select slightly different parameters. Each of the parameters is discussed next.

Note: In the descriptions that follow, default parameters are indicated with **bold** type.

Interface Manager Software Vector

For all adapters.

Select an interrupt vector for the several pieces of the driver-set to use during run time. This is a software interrupt—it requires no Dipswitch or Jumper modifications to the adapter. This vector must not be in use by any other software. Usually the default value need not be changed.

Valid values for *all* adapters:

60, 61, **62**, 63, 64, 65, 66, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F

Adapter Interrupt Vector

For DataFire and PC-IMAC ISA adapters only.

Select a bus interrupt vector for the adapter to use. The address you choose must not be used by any other device installed in your system.

Valid Interrupt Vector values for DataFire and PC-IMAC ISA adapters:

3, 5, 7, 10, **11**, 12, 15

Adapter Memory Address

For PC-IMAC ISA adapters only.

The PC-IMAC requires a 16K block of memory that is unused by any other device. Since other devices also require memory space, you may need to try several addresses before you find one that works.

C0000, C4000, C8000, CC000, **D0000**, D4000, D8000, DC000, E0000, E4000, E8000, EC000

Note: CMOS Shadowing may need to be disabled in your BIOS for the memory range you select.

I/O Address

For DataFire and PC-IMAC ISA adapters only.

Select the I/O address that matches the DIP switch setting on your adapter. The address must not be used by any other device in your system. Refer to your hardware manual for details about switch settings (or press <F1> for help).

Valid values for the PC-IMAC ISA adapter:

100, 110, 120, 200, 220, 300, **320**

Valid values for the DataFire adapter:

110, 140, 150, 300, 310, 340, **350**

Note: Remember that the I/O Port setting you choose must match the address to which the DIP switches are set.

Adapter Binary Load File

For all adapters.

This is the name of the binary file downloaded to the adapter during each driver initialization. During installation, the file is copied to the directory you named as the Destination Path.

Binary file name for PC-IMAC adapters (Micro Channel and ISA models):

IDP_XFS.BIN

Binary file name for DataFire adapters:

ADP.BIN

Binary file for Australia switch (PC-IMAC only):

AUSIDPX.BIN

Connection Manager Status Display

For all adapters.

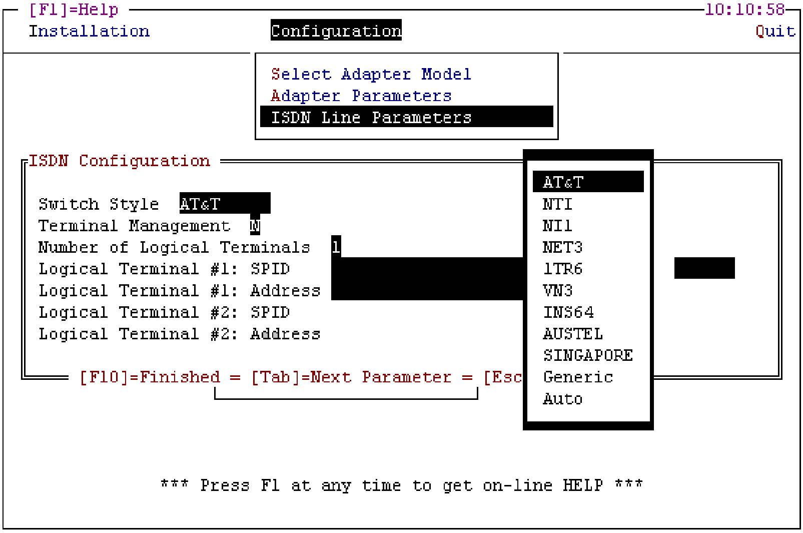
The Connection Manager monitors line status and will display current line status on the console if enabled. The Connection Manager is loaded as part of the driver set, but this option lets you choose whether the Connection Manager will be visible in the upper right corner of your console when you load the driver-set.

You can use the driver shell program to turn the display on and off. See page A-13 for more information about using the *set status* shell command.

This option defaults to **Yes** (Connection Manager status will display).

LINE Configuration

Use the mouse or the <Tab> key and arrow keys to highlight Configuration, then select ISDN Line Parameters; you will see a screen similar to the following:



```
[F1]=Help                                                          10:10:58
Installation               Configuration                               Quit
                          Select Adapter Model
                          Adapter Parameters
                          ISDN Line Parameters

ISDN Configuration                            AT&T
                                              NTI
 Switch Style  AT&T                           NI1
 Terminal Management  N                       NET3
 Number of Logical Terminals 1                1TR6
 Logical Terminal #1: SPID                    VN3
 Logical Terminal #1: Address                 INS64
 Logical Terminal #2: SPID                    AUSTEL
 Logical Terminal #2: Address                 SINGAPORE
                                              Generic
    [F10]=Finished = [Tab]=Next Parameter = [Esc  Auto

             *** Press F1 at any time to get on-line HELP ***
```

Figure 2-5. Line Parameters

Fill in the parameters with the information you have obtained from your ISDN service provider.

Switch Style

This refers to the type of software used with the ISDN switch at your ISDN service provider's central office. The installation program allows you to choose one of the following options:

AT&T	custom software for AT&T 5ESS, Generic 5E4.2 and up switches
NTI	custom software for Northern Telecom DMS-100, Functional or BCS-29 and up switches
NI-1	software that is compatible with the National ISDN standard—can be used with many types of switches
NET3	European Community
1TR6	Germany
VN3	France
INS64	Japan

Generic	Generic CCITT/ANSI switch type
AUSTEL	Australian compatible switch
SINGAPORE	Singapore compatible switch
Auto	for Auto Detect—select only if switch software is unknown

Note: Be certain that your selection reflects the *software* used with the switch, not the type of hardware. For example, your central office can use AT&T hardware, but run it with NI-1 software *or* AT&T custom software. If the NI-1 software was in use by the central office, you would choose NI-1 from the list above.

Terminal Management

This Yes or No selection applies only when you choose the AT&T custom software from the list of switch styles. Terminal Management is needed if your line is Terminal Type B, C, or D, to issue and receive voice calls. Your service provider will be able to tell you your Terminal Type.

Number of Logical Terminals

If you receive two phone numbers from your service provider, you must enter 2 here. The number of Logical Terminals required for your service is determined by the type of switch your provider is using and the type of service you order.

Logical Terminal (#1, #2) SPID

The SPID (Service Profile Identifier) uniquely identifies each terminal on a line, to the local switch. Some switches do not need a SPID (for example, most international switches do not use SPIDs). If you need a SPID, it will be supplied by your service provider and you must enter it exactly. If you chose "2" for the Number of Logical Terminals option, you must enter two different SPIDs.

Logical Terminal (#1, #2) Address

This is the telephone number of the Logical Terminal. If you chose "2" for the Number of Logical Terminals option, you must enter two different phone numbers. Each phone number identifies a terminal to be used for incoming calls.

Note: Enter only the 7-digit phone number(s). The phone number(s) must consist of only numerals in the range of 0 to 9; there must be no spaces or other keystrokes (like a dash) in the number.

What Next

Now that all the driver components have been copied and you have configured the adapter hardware for use in your system and for use with the ISDN line installed, you can load the driver files and test that the configuration works. Choose one of the driver sets below:

Installing ODI drivers

- For planning and preparation instructions for an ODI environment, go to page 3-1.

Installing NDIS drivers

- For planning and preparation instructions for an NDIS environment, go to page 4-1.

Other Information

- For information on using the Status Bar go to page 9-2.
- For information about driver shell command line switches, go to page A-1.

chapter **3**

Planning and Preparation for the ODI Driver

In this chapter

This chapter gives you important information about the Digi driver set you will need for your ODI environment and the preparation you can perform to ensure that all components required to make them work in your system are available. It discusses the following topics:

Planning and Preparation

Before you can implement the ODI driver you must do the following:

- Decide whether you will use Digi’s “Classic” encapsulation or PPP encapsulation
- Load ODI components on your system
- Prepare a Batch file to load all components in the correct order

Digi “Classic” vs. PPP Encapsulation

Connecting using “Classic” Encapsulation

If you plan to call a system with another Digi product that uses a “Classic” driver, much of the setup details will be handled by the built-in compatibility of the products.

Note: The drivers that use a Digi proprietary type of encapsulation to talk to another Digi device are the “Classic” versions. When you see the letters CL in the name of a batch file it refers to this version.

Connecting to a PPP device

Since PPP provides flexibility by allowing several types of connections, you must know what the device expects when it calls or receives a call, in these two areas:

- type of PPP authentication
- type of PPP protocol

Type of PPP authentication

Your choices here will be PAP or CHAP, or NONE. You will need to know this as you set up profiles that pass setup parameters to the driver when you place a call. A whole section of this manual is dedicated to these “Connection Profiles”. For more information see page 6-1

Type of PPP protocol

The type of PPP control protocol used by the device that you will call can be: IP Control Protocol, IPX Control Protocol, or Bridging Control Protocol.

Digi provides a PPP module (PPP.EXE) to handle negotiations for your PPP connection. Depending on the type of PPP control protocol required by the device to which you wish to connect, you must provide the correct arguments(s) when this executable is called. These are the valid arguments:

ppp	with no argument, the default is the same as using /bcp
ppp /bcp	use bridging control protocol; use bridge encapsulation for all protocols
ppp /ipcp	use IP control protocol; all other protocols blocked
ppp /ipxcp	use IPX control protocol; all other protocols blocked
ppp /bcp /ipcp	use IP protocol over PPP, use bridge encapsulation for all other protocols
ppp /bcp /ipxcp	use IPX protocol over PPP, use bridge encapsulation for all other protocols

Note: If you choose /ipcp or /ipxcp, you must specify an Ethernet_II Frame Type in NET.CFG as is discussed in the next section.

Preparing an ODI Environment

If you will use the ODI stack protocol, you must install the software that you will use. Most often, in the NetWare environment, this is the NetWare Client software. Loading the application software ensures that the files you need in order to use ODI with your Digi product will be available to load when you need them. The NetWare files you must have are:

- NET.CFG
- LSL.COM
- IPXODI.EXE

In addition to the files above, to make a connection, you must have the following file:

- VLM.EXE

If possible, obtain the latest versions of these files from Novell.

Location of the files

To load the files using a Digi batch file you must know the name of the directory in which they reside, or move them to the directory into which you load your Digi drivers. These three files are usually loaded into a directory called /NWCLIENT.

Load order of the files

There is a certain order in which the ODI drivers listed above and the Digi drivers must be loaded. This is described on page 3-8.

IP Addressing in NET.CFG for “Classic” Driver

If you plan to use the “Classic” ODI driver, you can still use IP addressing by editing the NET.CFG file to add the IP addresses. Add these entries in the same manner as you would use with a NIC card.

When using the PPP driver, IP addressing is handled in your connection profile which is described in more detail on page 6-11.

Frame type selection in NET.CFG

The default Ethernet frame type is 802.2. To use other frame types, you must add statements similar to the following into your NET.CFG file:

```
Link Driver IDPODI or PPPODI
        Frame Ethernet_802.3
        Frame Ethernet_II
        Protocol IPX 0 Ethernet_802.3
        Protocol IP 0800 Ethernet_II
```

TIP: You *must* TAB-indent these lines for the file to be correctly processed!

Note: A sample NET.CFG file is included with your driver. Look for the DIGINET.CFG file in the driver directory.

To cause IPXODI to choose the Ethernet_802.3 frame type, you must also include a statement like the following in NET.CFG:

```
Protocol IPXODI
Bind #1
```

It is important that your application, your driver, and your destination Frame Type information be compatible, as is explained next.

Selecting a Frame Type when using the "Classic" driver

To make a connection using the "Classic" driver, the Frame Type used by the Digi ISDN adapters in remote and local systems must be compatible. The Frame Type default for NetWare Client software is 802.2, specified in the NET.CFG file. Figure illustrates the following about "Classic" driver Frame Type:

- The Frame Type used between Digi adapters across an ISDN connection *must* match.
- The Frame Type used on a LAN (attached to the server in which a Digi adapter is installed) can be *different* than the Frame Type used across the ISDN connection between Digi adapters. (Client to Server 1)
- If the Frame Type used by the Digi adapter in the client matches the Frame Type on the LAN attached to a server, but does not match the Frame Type used by the Digi adapter in the server, the connection will not work. (Client to Server 2)

Given these facts, you may need to make a change to either your server Connection Profile, or change the Frame Type specified in the NET.CFG file for your Client, to ensure that there is a Frame Type match.

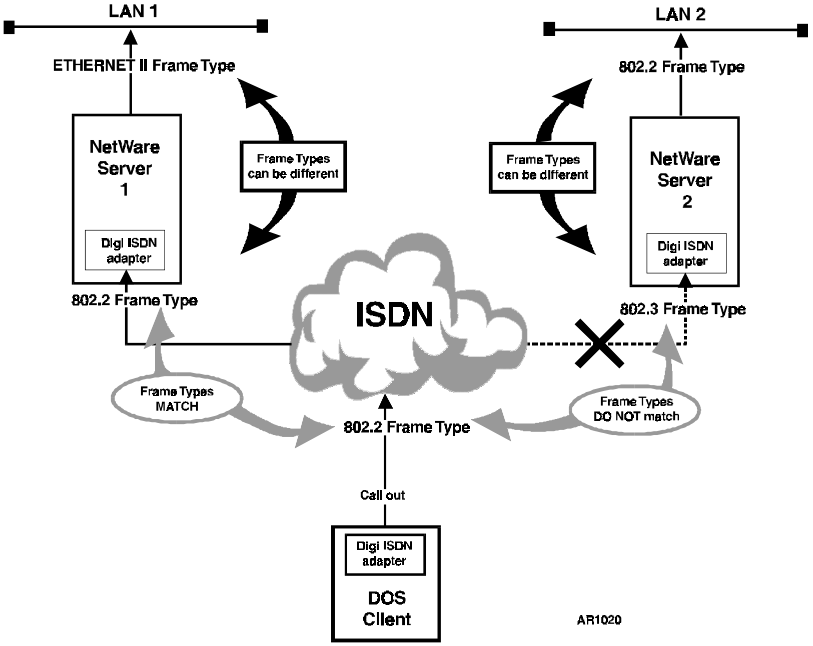


Figure 3-1. Frame Types and "Classic" driver

Selecting a Frame Type when using the PPP driver

Selecting a Frame Type to use when making a connection using the PPP driver depends on the Control Protocol that you intend to use: IP or IPX Control Protocol, or Bridging Control Protocol. The Control Protocol you use, in turn, depends on the remote device to which you will connect:

- If you plan on connecting to a PPP device using the IP or IPX protocol, and specify either /ipcp or /ipxcp as an argument to PPP.EXE, you *must* change your NET.CFG file to specify and bind Ethernet II. See page B-9 for an example NET.CFG file.
- If you choose the /bcp argument to PPP.EXE, your NET.CFG can specify and bind any of the Frame Types, but you must be sure that it is the same Frame Type as is used by the destination LAN.

Figure illustrates the following about PPP driver Frame Type:

- The Frame Type used between a Client and a PPP device using /ipcp or /ipxcp across an ISDN connection can be *different*, but Digi's PPP driver requires a Frame Type of Ethernet II. (Client 1 to LAN 1)
- To communicate to a LAN through a Bridge device using /bcp, the Frame Type used by the Client must be *the same* as the Frame Type used by the LAN attached to the Bridge. (Client 2 to LAN 2)

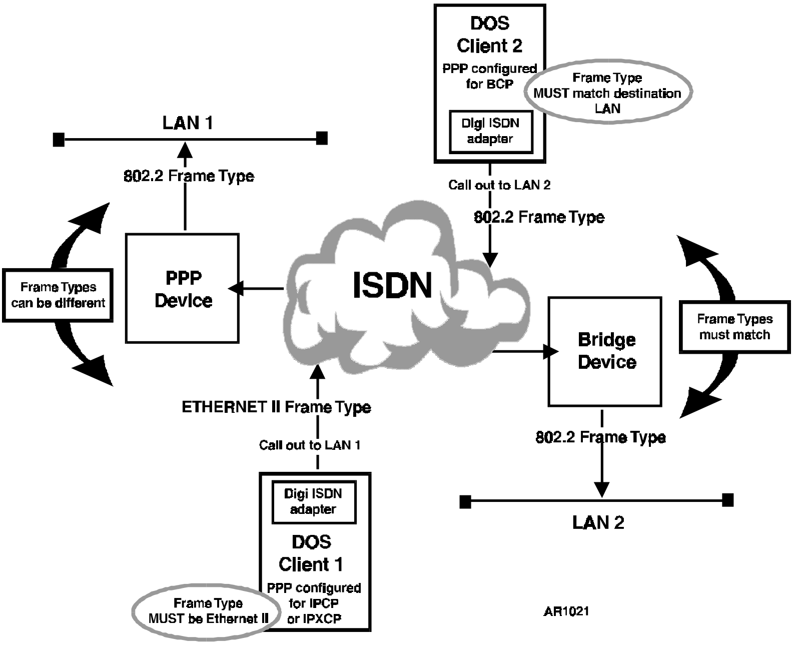


Figure 3-2. Frame Types and PPP driver

Using Batch Files to Load Driver Components

There are multiple executable programs that must operate together for your ISDN connection to be successful. To be sure that all components necessary for a connection are properly loaded (each time you want to make the same type of connection), you should create a batch file.

The simplest batch files merely list DOS commands or executable filenames in the proper order. More sophisticated batch files can check for error conditions or look for command line modifiers. Digi has included a batch file for each type of driver-set you may wish to use. Although the Digi batch files are easy to use, you do not need to use them; you can create your own files.

Digi ODI Batch Files

Use one of the following batch files, depending on the type of device to which you wish to connect:

- CLODI.BAT Loads the ODI “Classic” drivers, those that use Digi’s proprietary encapsulation method
- PODI.BAT Loads the PPP drivers for ODI; edit for the type of PPP protocol you want

Each of these ODI batch files look for a load-line argument that specifies the location of the ODI components needed to run the ODI stack. If you supply no argument, the batch file assumes the files are in the current directory.

Example LOAD line command with directory name as an argument:

```
LOAD CLODI \nwclient
```

What Digi’s batch files do

When you execute one of Digi’s batch files, the first thing it does is scan your system for the type of Digi adapter you have. Then it looks to see if you specified a directory name on the command line. Last, it begins loading the components required.

Editing the Digi Batch Files

Edit a Digi batch file in order to do the following automatically:

- **Use the proper PPP protocol.** Edit a PPP batch file so that the correct argument is used to modify the PPP core module. Do this by deleting the REM statement in front of the line you want to use. See page 3-3 for more information about the arguments.

- **Call a remote site.** Edit the section of the batch file that provides an example connection to actually connect to the remote site you wish to call. Do this by removing the REM statements that precede the lines.
- **Run networking software.** Edit the section of the batch file that provides example lines that load networking software from the batch file. Do this by removing the REM statements that precede the lines, or add new lines that call the software that you need loaded.
- **Add a login.** Edit the section of the batch file that provides example lines to execute LOGIN.EXE. Do this by removing the REM statements that precede the lines.

Creating your own ODI batch file

If you create your own batch file to use, either initially or long-term, you must place ODI components and Digi components in the proper order for loading. This order is shown in the table below:

Using the Digi “Classic” Driver		Using the PPP Driver	
with a DataFire	with a PC-IMAC	with a DataFire	with a PC-IMAC
ADPIM.EXE	IDPIM.EXE	ADPIM.EXE	IDPIM.EXE
PCICM.EXE		IDPCM.EXE	
* LSL.COM		* LSL.COM	
		† PPP.EXE	
IDPODI.COM		PPPODI.COM	
* ‡ IPXODI.EXE		* ‡ IPXODI.EXE	

* Files are to be acquired from Novell.

‡ File necessary only if you want to use IPX protocol for connection.

† Without arguments, defaults to Bridge Control Protocol. See page 3-3 for information about changing the default

Example Batch Files

You can create a simple batch file by merely placing the load lines in the correct sequence. Several example files are shown next:

Example 1:

```
ADPIM
IDPCM
LSL
PPP \ipcp
PPPODI
```

This batch file will load all the drivers required for a DataFire adapter to connect to a remote site with an ODI stack using the IP control protocol.

Note: Since IPX will not be used, the file IPXODI.EXE is not included.

Example 2:

```
ADPIM
PCICM
LSL
IDPODI
IPXODI
```

This batch file will load the “Classic” drivers required for a DataFire adapter to connect to another Digi ISDN adapter using the “Classic” driver at a remote site with an IPX protocol.

What Next

- Once you have the NetWare files that you need, understand the order in which you must load the NetWare and Digi files, and have a batch file that will load them properly, go to page 5-1 for instructions on how to load and test the drivers.

chapter 4

Planning and Preparation for the NDIS Driver

In this chapter

This chapter gives you important information about the Digi driver set you will need for your NDIS environment and the preparation you can perform to ensure that all components required to make them work in your system are available. It discusses the following topics:

Planning and Preparation

Before you can implement the NDIS driver you must do the following:

- Decide whether you will use Digi's “Classic” encapsulation or PPP encapsulation
- Load NDIS components on your system
- Prepare a Batch file to load all components in the correct order

Digi “Classic” vs. PPP Encapsulation

Connecting using “Classic” Encapsulation

If you plan to call a system with another Digi product that uses a “Classic” driver, much of the setup details will be handled by the built-in compatibility of the products.

Note: The drivers that use a Digi proprietary type of encapsulation to talk to another Digi device are the “Classic” versions. When you see the letters CL in the name of a batch file it refers to this version.

Connecting to a PPP device

Since PPP provides flexibility by allowing several types of connections, you must know what the device expects when it calls or receives a call, in these two areas:

- type of PPP authentication
- type of PPP protocol

Type of PPP authentication

Your choices here will be PAP or CHAP, or NONE. You will need to know this as you set up profiles that pass setup parameters to the driver when you place a call. A whole section of this manual is dedicated to these “Connection Profiles”. For more information see page 6-1.

Type of PPP protocol

The type of PPP protocol used by the device that you will call can be: IP, IPX, or Bridging Protocol.

Digi provides a PPP module (PPP.EXE) to handle negotiations for your PPP connection. Depending on the type of PPP protocol required by the device to which you wish to connect, you must provide the correct argument(s) when this executable is called. These are the valid arguments:

ppp	
ppp /bcp	negotiate bridging control protocol; use bridge encapsulation for all protocols
ppp /ipcp	negotiate IP control protocol; all other protocols blocked
ppp /ipxcp	negotiate IPX control protocol; all other protocols blocked
ppp /bcp /ipcp	use IP protocol over PPP, use bridge encapsulation for all other protocols
ppp /bcp /ipxcp	use IPX protocol over PPP, use bridge encapsulation for all other protocols

The PPP.EXE module must be loaded with the proper argument, in the proper order, with the other driver modules from Digi. The best way to ensure that they are loaded correctly is to use a batch file.

Preparing an NDIS Environment

You can use Digi's DOS NDIS 2.0 driver in a Windows for Workgroups or Windows 3.1 environment, and also with DOS applications that work within the NDIS 2.0 environment.

Windows for Workgroups

If you have this version of Windows loaded on your local system, you can follow these steps to create a working environment for your Digi Adapter:

1. Run *Network Setup*, select *Drivers* and *Add Adapter.*
2. In the *Add Adapter* screen, select *Unlisted or Updated Network Adapte*r.

   You will see the *Install Driver* screen. Insert the Digi driver diskette and choose the diskette drive letter.
3. Select the driver to use, and add or remove the protocols you want.
4. Press *Close* when you are done.

You must reboot your system for the driver to be loaded, but first you should check your AUTOEXEC.BAT file to be sure that the drivers will load in the proper order. This order is described more completely on page 4-8.

Note: NET INIT or NET INITIALIZE must be executed in your AUTOEXEC.BAT file *before* the lines loading your Digi executables, and NET START must be executed *after* the Digi executables are loaded.

Windows 3.1 or DOS Applications

If you are using Windows 3.1 or a DOS Application (such as an Ethernet driver) that uses the NDIS stack, you will need to be sure that all the files required to initialize and use NDIS are 1.) present and 2.) properly set up.

These are the files that you typically need to have:

- PROTMAN.DOS or PROTMAN.SYS
- PROTOCOL.INI
- NETBEUI.DOS

A good way to get the files listed above is to install the driver software that comes with an Ethernet card. You could also find these files on another system and copy them. Either way, you will need to edit several files as described in the following steps:

1. Edit the PROTOCOL.INI file to include information about your Digi product. (This file provides configuration information and parameters to NDIS and MAC modules, telling them how they connect together.)

a. Add the following lines to the MODULE section of the file if you will use the "Classic" driver:

```
[DIGI]
        drivername = DIGIMAC$
```

Note: If you want to use the PPP driver, substitute PPPMAC$ on the first line (for DIGIMAC$).

b. Add the following in the MODULE section of the PROTOCOL.INI file:

```
[NETBEUI]
        drivername = NETBEUI$
        bindings = DIGI
```

Note: The binding statement uses the name of the module you created in step a. If you change the name of the module, you must change the name in the BINDINGS statement.

2. Ensure that your CONFIG.SYS file includes two things:

- *A line to load the NDIS Protocol Manager driver* (PROTMAN.DOS or PROTMAN.SYS).

  ```
  DEVICE=C:\DIR_NAME1\PROTMAN.DOS /i C: DIR_NAME2
  ```

  where
  DIR_NAME1
  is the directory in which PROTMAN.DOS is located and
  DIR_NAME2
  is the directory in which PROTOCOL.INI is located.
  (The /i argument to PROTMAN.DOS lets you specify a path.)

  The NDIS Protocol Manger reads the NDIS configuration information in PROTOCOL.INI.

- *A line that loads the NDIS protocol driver.* NETBEUI.DOS is a good example of this type of driver. Be sure that the entire path is specified:

  ```
  DEVICE=C:\DIR_NAME3\NETBEUI.DOS
  ```

Example:

```
DEVICE=C:\LANMAN.DOS\DRIVERS\PROTMAN\PROTMAN.DOS /i C:LANMAN.DOS
DEVICE=C:\LANMAN.DOS\DRIVERS\PROTOCOL\NETBEUI.DOS
```

Note: If you want to load a TSR version of NETBEUI, you can do that by placing a statement in your AUTOEXEC.BAT. In that case, do not include a statement in your CONFIG.SYS file.

3. Ensure that the statements in your AUTOEXEC.BAT file are correct.

- The PATH statement in your AUTOEXEC.BAT file must list the directories in which PROTOCOL.INI, PROTMAN.SYS, and NETBEUI.DOS reside:

```
PATH C:\DIR_NAME1;DIR_NAME2;DIR_NAME3
```

- To load the NDIS protocol as a TSR, include a LOAD statement:

```
LOAD NETBEUI
```

Note: If you load NETBEUI in AUTOEXEC.BAT, do not load it in CONFIG.SYS

- Include a statement that executes NETBIND:

```
NETBIND
```

Using Batch Files to Load Driver Components

There are multiple executable programs that must operate together for your ISDN connection to be successful. To be sure that all components necessary for a connection are properly loaded (each time you want to make the same type of connection), you should create a batch file.

The simplest batch files merely list DOS commands or executable filenames in the proper order. More sophisticated batch files can check for error conditions or look for command line modifiers. Digi has included a batch file for each type of driver-set you may wish to use. Although the Digi batch files are easy to use, you do not need to use them; you can create your own files.

Digi NDIS Batch Files

Use one of the following batch files, depending on the type of device to which you wish to connect:

- CLNDS.BAT Loads the “Classic” NDIS drivers, those that talk to another Digi product
- PNDIS.BAT Loads the PPP drivers

What Digi’s batch files do

When you execute one of Digi’s NDIS batch files, the first thing it does is scan your system for the type of Digi adapter you have. Then it begins loading the components required.

Editing the Digi Batch Files

Edit a Digi batch file in order to do the following automatically:

- **Use the proper PPP protocol.** Edit a PPP batch file so that the correct argument is used to modify the PPP core module. Do this by deleting the REM statement in front of the line you want to use. See page 4-3 for more information about the arguments.
- **Call a remote site.** Edit the section of the batch file that provides an example connection, using the driver shell program, to actually connect to the remote site you wish to call. Do this by removing the REM statements that precede the lines.
- **Run networking software.** Edit the section of the batch file that provides an example line that will call NETBIND from the batch file. Do this by removing the REM statement that precedes the line.
- **Add a login.** Edit the section of the batch file that provides example lines to execute LOGIN.EXE. Do this by removing the REM statements that precede the lines.

Creating your own NDIS batch file

If you create your own batch file, either to use initially or for long-term use, you must place driver components in the proper load order. This order is shown in the table below.

Using the Digi “Classic” Driver		Using the Digi PPP Driver	
with a DataFire	with a PC-MAC	with a DataFire	with a PC-MAC
ADPIM.EXE	IDPIM.EXE	ADPIM.EXE	IDPIM.EXE
PCICM.EXE		IDPCM.EXE	
		† PPP.EXE	
DIGIMAC.EXE		PPPMAC.EXE	

† Without arguments, defaults to Bridge Control Protocol. See page 4-3 for information about changing the default

Example Batch File

An example file, NDS_LOAD.BAT is shown below:

```
ADPIM
IDPCM
PPP /bcp /ipxcp
```

This batch file will load all the drivers required for a DataFire adapter to connect to a remote site with an NDIS stack using the IPX control protocol over PPP and bridge encapsulation for everything else.

Using AUTOEXEC.BAT

If you want to load the drivers from AUTOEXEC.BAT, you can add load lines for each of the files listed in the table above, or a single load line to execute a batch file (that in turn loads the drivers).

Example LOAD statement:

```
LOAD NDS_LOAD
```

What Next

- Once you have the NDIS files you need, understand the order in which you must load the NDIS and Digi files, and have a batch file that will load them properly, go to page 5-1 for instructions on how to load and test the drivers.

chapter **5**

Implementation and Testing

In this chapter

This chapter gives you step-by-step instruction about loading drivers for your Digi adapter and testing them to be sure they work in your system. It discusses the following topics:

Starting Point

Before you follow the steps described in this chapter, be sure that the following has been done:

- Digi ISDN adapter, ISDN line, and driver components are installed.
- The application(s) you will use with your Digi product are installed.
- A known-good remote phone number is available to call.

Overview of Steps to Load and Test Drivers

Although the components of the drivers you will use with your Digi ISDN adapter are loaded on your system, you must follow the instructions in this chapter to utilize them effectively for your situation. This chapter describes the driver components you will need to use and also provides instruction to help you test your installation to be sure that it is working properly.

These are the basic steps you will follow in this chapter:

1. Execute batch file to load driver components
2. Test that Layer 3 is available
3. Connect to a remote site

1: Loading Driver Components

Starting Point

The adapter and ISDN line must be installed, and driver files listed on page 3-7 (for ODI) or page 4-7 (for NDIS) must be available.

Execute the batch file

Whether you created your own or are using the Digi batch file, execute the file so that the driver components load.

- You should see several start-up messages that indicate the driver-set has loaded successfully.
- You should be able to toggle the Status Bar ON and OFF by simultaneously pressing the <Alt-Left Shift-D> keys.
- If you have a line plugged into your DataFire adapter, the Connection Manager Status Display (in the upper corner of your terminal display) should (in most cases) read:

  DIGIISDN Status: 4-0-0 Rx:00.. Tx:00..

  or

  DIGIISDN Status: 4-4-0 Rx:00.. Tx:00..

  For more information about the Connection Manager Status Display, see page 9-2.

2: Testing for Layer 3

Starting Point

The adapter and ISDN line must be installed, and all the necessary drivers must be loaded for a test for Layer 3 to be successful.

Bring up the correct shell program

Use the appropriate driver shell program (*clsh* if you are using a “classic” driver, or *pppsh* if you are using a PPP driver) to look at Layer 3 status. First bring up the shell program by typing its name on the command line:

```
C:\> pppsh <Enter>
```

You will see a command line prompt like the following:

```
C:\ ppp>
```

Examine Layer 3 Status

To examine layer 3, type the following:

```
C:\ppp> show idp /l3
```

This instructs the driver to display layer 3 status. A summary of the status will display, providing information about the connection. If you see any of the following status displayed next to the word *State*, layer 3 connection was probably successful:

- Initialized
- Active
- Established, Awaiting Restart

If you see any other status, layer 3 is not up. See the *Troubleshooting and Tech Tips* information starting on page B-1 for help.

3: Manual Connect to Remote

Starting Point

The adapter and ISDN line must be installed, and all the necessary drivers must be loaded for a connection to be made to a remote site.

For this step, the correct driver shell program must also be loaded (see page 5-4).

Use the Connect command

You can connect to a remote site by typing a command, using the following syntax, on the shell command line:

C:\DIGIISDN\ ppp> **conn** *phone_number*

In order to verify that all the pieces are working, use a phone number that you know is available for a connection.

A successful connection

If you connect to the remote site, you will see a message reporting that the connection was successful.

If you do not see this type of message, check the phone number you used on the command line:

- do not use spaces or dashes in the number
- if long distance, be sure all prefixes are correct
- if pbx or centrix, use a prefix of 9

For more information, check the *Troubleshooting and Tech Tips* information starting on page B-1.

What Next

- At this point you know that you can connect to a remote site. This means that both hardware and drivers are correctly setup. The next step is to create one or more connection profiles that you can use to easily make the connections you need.

  Connection profiles are explained in detail starting on page 6-1.

chapter 6

Connection Profiles

In this chapter

This chapter gives you step-by-step instruction for creating and managing connection profiles used to access remote sites. It discusses the following topics:

Starting Point

Before you begin creating profiles, you should understand the types of connections you will be making. You should plan the connections you will make with remote sites and be sure to have phone numbers and passwords available if you need them.

About the Profiles

What they will do

You can use profiles to automate the connections that you want to make. Once you have setup an application, you can have separate profiles that initialize your Digi ISDN adapter for use with that application.

For example, if you need to call a certain server periodically, you can create a *calling* profile that sets up your Digi product the same way each time. To setup your system to receive a particular call, you can create a profile that will *listen* for it.

Shell programs

Profiles are created and called using one of the driver shell programs:

• *clsh*	Shell program available with the “classic” drivers
• *pppsh*	Shell program available with the PPP drivers

There are numerous commands available for use with the shell programs. These are documented in the *Shell Command Reference* beginning on page A-1.

Default profiles

The default profiles are your starting point for creating new profiles. There are two available:

- default — A profile that defaults to calling. If you use the connect command without specifying a profile name, this profile is used to provide connection parameters required by the driver. (See page 7-3 for more information.)
- listen — A profile that defaults to listening

Although it is not recommended to directly edit either of the default profiles listed above (the best course is to copy to a new profile and edit that profile), it is possible to do so. A third profile exists, called factory_defaults that cannot be changed. Use it restore the default values to the default profile.

Using a profile

Once created, you can use a profile in the following ways:

- *From the DOS Command line*. To execute a profile from the command line, use the following syntax:

  clsh *profile_name* or pppsh *profile_name*

  where *profile_name* is the name of the profile you want to execute.
- *Executed from a DOS batch file.* To execute a profile from within a DOS batch file include a command line to execute the profile using the syntax shown above.

Table 6-1: Profile Parameters

Parameter and values	“listen” profile	“default” profile
name=*profile name*	“listen” /memory	“default” /memory
access=nailed_up|demand	demand	demand
channel=any|b1|b2	any	any
number_of_channels=0|1|2	2	1
persistent=yes|no	no	no
type_of_channel=d64|d54|voice	d64	d64
address=*phone number*	“ ”	“ ”
activation=manual|frame	manual	manual
idle_timer=*timeout value*	0	0
listen=yes|no	yes	no
fallback=yes|no	yes	yes
† authentication=none|chap|pap		
† user_name=anonymous|*name*		
† password= “ ”		
remote_name=*	*	*
† remote_password= “ ”		
connection_name=*conn%d*	“conn%d”	“conn%d”
debug_level=*level*	2	2

† Indicates the parameter is available only when running the *pppsh* shell.

For complete descriptions of these parameters, see the *Shell Command Reference* appendix on page A-1.

Creating Profiles

Basic steps for creating a new profile

These are the general steps you must take to create a profile:

1. Copy an existing profile.
2. Set the parameters that you want to change.
3. Save the new profile.
4. Test (Connect) the profile.

Instructions for creating profiles

The following pages describe how to create several types of profiles:

- For information about *Creating a Calling Profile,* go to page 6-5
- For information about *Creating a Listening Profile*, go to page 6-7
- For information about *Using 2-Channel Calling*, go to page 6-9
- For information about *Using IP Addressing*, go to page 6-11
- For information about *Using Frame-activation*, go to page 6-13

Creating a Calling Profile

Follow these steps to create a profile that calls a remote site. Bring up the appropriate driver shell before beginning.

Copy an existing profile

At the shell prompt, type the following command, using a meaningful name in place of the term *profile_name*:

```
> copy profile default profile_name
```

Example:

```
> copy profile default boston
```

The name of the new profile is boston.

Set profile parameters

Next, type in a series of profile parameters that tailor this profile for the type of connection you want to make. The syntax to use is:

```
> set profile profile_name /param1= /param2= /param3= and so on
```

Example:

```
> set profile boston /address=5551212 /type_of_channel=d56
```

The new profile will make a call to phone number 555-1212 using a d56 rate. Keep the following in mind as you assign parameters:

- Do not use any spaces or dashes within the phone number.
- Always place a space between each parameter.
- Use only numbers 0-9 for phone numbers.

Save the profile

If you do not specifically save the profile you have created, it will be lost when you quit the shell program. To save your profile, use the following command:

```
> save profile profile_name
```

Example:

```
> save profile boston
```

The profile called boston is now stored as a file on your system hard disk.

Test the profile

To ensure that the parameters are properly setup, test the profile using the following command syntax:

```
> connect profile_name
```

Example:

```
> connect boston
```

If the connection is successful, you will see connection message display. If your connection does not work, see *Troubleshooting and Tech Tips* on page B-1.

Creating a Listening Profile

Follow these steps to create a profile that listens for a call from a remote site. Bring up the appropriate driver shell before beginning.

Copy an existing profile

At the shell prompt, type the following command, using a meaningful name in place of the term *profile_name*:

```
> copy profile listen profile_name
```

Example:

```
> copy profile listen corporate
```

The name of the new profile is corporate.

Set profile parameters

Next, type in a series of profile parameters that tailor this profile for the type of device that will call. The syntax to use is:

```
> set profile profile_name /param1= /param2= /param3= and so on
```

Example:

```
> set profile corporate /channel=1 /activation=frame /remote_name=hq
```

The new profile will make one channel available for a call from the “hq” server only. Since “hq” is set up for frame activation, so must this client be set up for frame activation.

Save the profile

If you do not specifically save the profile you have created, it will be lost when you quit the shell program. To save your profile, use the following command:

```
> save profile profile_name
```

Example:

```
> save profile corporate
```

The profile called corporate is now stored as a file on your system hard disk.

Test the profile

To ensure that the parameters are properly setup you should test the profile. For a listening profile, do this by initiating the listening profile, and then attempting a call from the expected remote site. To set up the listening connection, use the following command syntax:

```
> connect profile_name
```

Example:

```
> connect corporate
```

If the connection is successful, you will see a connection message display. If your connection does not work, see *Troubleshooting and Tech Tips* on page B-1.

Using 2-Channel Calling

Follow these steps to create a profile that uses 2 channels to call a remote site. Bring up the appropriate driver shell before beginning.

Create a new profile or edit an existing profile

To create a new profile, at the shell prompt, type the following command, using a meaningful name in place of the term *profile_name*:

```
> copy profile default profile_name
```

Example:

```
> copy profile default 2channel
```

The name of the new profile is 2channel.

To edit an existing profile, at the shell prompt, type the following command, using the name of the profile you want to edit in place of the term *profile_name*:

```
> restore profile profile_name
```

Set profile parameters

Next, define the profile parameters for a two-channel connection using the following syntax:

```
> set profile profile_name /address="address1:address2" /num=2
/channel=b1:b2
```

Use the */channel* parameter to select **local** B channel options as follows:

If you have two local phone numbers, choose /channel=b1:b2a
If you have one local phone number, choose /channel=b1:b2

Use the */address* parameter to select **remote** B channel options as follows:

If you have two remote phone numbers: /address=*“address1:address2”*
If you have one remote phone number: /address=*“address1:address1”*

Note: You *must* enclose the phone numbers in quotes.

Example:

```
> set profile 2channel /address=“5551212:5551010” /num=2
/channel=b1:b2a
```

This profile will make a 2-channel call to phone numbers 555-1212 and 555-1010.

Keep the following in mind as you assign parameters:

- Do not use any spaces or dashes within the phone number, and enclose the phone numbers in quotes as shown.
- Always place a space between each parameter.
- Use only numbers 0-9 for phone numbers.

Save the profile

If you do not specifically save the profile you have created, it will be lost when you quit the shell program. To save your profile, use the following command:

> save profile *profile_name*

Example:

```
> save profile 2channel
```

The profile called 2channel is now stored as a file on your system hard disk.

Test the profile

To ensure that the parameters are properly setup, test the profile using the following command syntax:

> connect *profile_name*

Example:

```
> connect 2channel
```

If the connection is successful, you will see a connection message display. If your connection does not work, see *Troubleshooting and Tech Tips* on page B-1.

Using IP Addressing

With the “Classic” ODI driver

It is possible to use IP addressing with the “Classic” ODI driver by editing the NET.CFG file. See page 3-4. In that case, IP addressing is not handled in the connection profile, and the instructions in this section do not apply.

With the PPP driver

To use IP, you must load the PPP.EXE program with the /ipcp switch. This switch is described on page 3-3 (ODI) or on page 4-3 (NDIS).

Follow these steps to create a profile that sets up the IP addresses used by an IP connection with PPP. Bring up the appropriate driver shell before beginning.

Create a new profile or edit an existing profile

To create a new profile, at the shell prompt, type the following command, using a meaningful name in place of the term *profile_name*:

> \> copy profile default *profile_name*

To edit an existing profile, at the shell prompt, type the following command, using the name of the profile you want to edit in place of the term *profile_name*:

> \> restore profile *profile_name*

Set IP parameters

Next, define the profile parameters for an IP connection using the following syntax:

> \> set profile /local_ip=*“ip_address”* remote_ip=*“ip_address”*

Use the */local_ip* parameter to specify the local IP address as follows:

- Use the exact address if you want to announce that address to the remote peer
- Use 0.0.0.0 in place of the address if you want to request an IP address from the remote peer
- Use 255.255.255.255 if you want no address negotiation to take place

Use the */remote_ip* parameter to select remote IP address options as follows:

- Use the exact address to be supplied if requested by the remote side
- Use 255.255.255.255 if you want no address negotiation to take place

Example:

```
> set profile diane_call /local_ip=0.0.0.0
```

This profile requests that the remote side supply an IP address to the local side of the connection.

Save the profile

If you do not specifically save the profile you have created or edited, all data will be lost when you quit the shell program. To save your profile, use the following command:

> save profile *profile_name*

Example:

```
> save profile diane_call
```

The profile called diane_call is now stored as a file on your system hard disk.

Test the profile

To ensure that the parameters are properly setup, test the profile using the following command syntax:

> connect *profile_name*

Example:

```
> connect diane_call
```

If the connection is successful, you will see a connection message display. If your connection does not work, see *Troubleshooting and Tech Tips* on page B-1.

Using Frame-activation

Follow these steps to create a profile that uses Frame-activation to control the connection. With Frame-activation, a connection does not stay up when there is no data to transmit, but reconnects automatically when data is ready to be sent.

Important: Although it is a convenient feature and can be useful to ensure that connections do not stay up unnecessarily, you must monitor connections made using the Frame-activation feature to ensure that the setup does not cause excessive automatic connections to be made, and thereby increase the line charges that will accrue.

Bring up the appropriate driver shell before beginning.

Create a new profile or edit an existing profile

To create a new profile, at the shell prompt, type the following command, using a meaningful name in place of the term *profile_name*:

```
> copy profile default profile_name
```

To edit an existing profile, at the shell prompt, type the following command, using the name of the profile you want to edit in place of the term *profile_name*:

```
> restore profile profile_name
```

Set Frame-activation parameters

Next, define the profile parameters for a Frame-activated connection using the following syntax:

```
> set profile profile_name /persistence=yes /activation=frame
/idle=seconds
```

Use the */persistence=yes* parameter to keep the connection viable even when it is disconnected after a time-out.

Use the */activation=frame* parameter to make the connection automatically depend on whether there is data to transmit.

Use the */idle* parameter to specify the length in seconds that the connection should remain “up” with no data available to transmit, waiting for additional data.

Note: The idle timer begins to count down from the timer value when there is no data being transferred from the remote or local side; if data is begins to be passed it resets to full value until the transfer is complete and then begins to count down again. If you want the timer to only reset to full value when a data transfer from the local side is initiated, use a negative value in the

/idle argument. *Example*: /idle=-40 will cause the timer to wait 40 seconds from the time the last transfer inititiated by the local side was completed before disconnecting.

Example:

```
> set profile corporate /persistence=yes /activation=frame /idle=30
```

This profile will make a call when the first data frame is ready to transmit, subsequently disconnect if there is no traffic for 30 seconds, and then switch to a Listening mode to await an incoming call from the remote device when it has data to pass.

Save the profile

If you do not specifically save the profile, your setup work will be lost when you quit the shell program. To save your profile, use the following command:

> save profile *profile_name*

Example:

```
> save profile corporate
```

The profile is now saved on your system hard disk.

Test the profile

To ensure that the parameters are properly setup, test the profile using the following command syntax:

> connect *profile_name*

Example:

```
> connect corporate
```

Since this connection is Frame-activated, a connection will not be initiated until there is data to send. To test the connection, attempt to send data. At that point, you should see a connection message display. If your connection does not work, see *Troubleshooting and Tech Tips* on page B-1.

Editing an Existing Profile

Follow these steps to edit a profile that already exists on disk. Bring up the appropriate driver shell before beginning.

Load the profile into memory

At the shell prompt, type the following command, using the name of the profile you want to edit in place of the term *profile_name*:

```
> restore profile profile_name
```

Example:

```
> restore profile corporate
```

The profile corporate will be loaded into memory.

Display the profile parameters

Before you change the parameters of a profile, it is helpful to know to what value the current parameters are set. To see the current parameters, use the following syntax:

```
> show profile profile_name
```

Example:

```
> show profile corporate
```

The parameters for the profile corporate will be displayed on the terminal screen.

Set profile parameters

Next, type in a series of profile parameters that tailor this profile for the type of device that will call. The syntax to use is:

```
> set profile profile_name /param1= /param2= /param3= and so on
```

Example:

```
> set profile corporate /channel=1 /remote_name=*
```

The profile will now allow calls from all servers.

Save the profile

If you do not specifically save the profile you have edited, the changes you made will be lost when you quit the shell program. To save your profile, use the following command:

```
> save profile profile_name
```

Example:

```
> save profile corporate
```

The changed profile called corporate is now stored on your system hard disk.

Test the profile

To ensure that the parameters are properly setup, test the profile using the following command syntax:

```
> connect profile_name
```

Example:

```
> connect corporate
```

Listing Available Profiles

This section describes how to do the following:

- Find out which profiles are currently loaded into memory
- Find out the names of all profiles saved on disk

To find which profiles are currently loaded in memory

You can load multiple connection profiles into memory at once. To find out which profiles are loaded using this command:

```
> list profiles
```

To list the names of all profiles saved on disk

You can a list of all the profiles that are saved to disk using this command:

```
> list profiles /file
```

Displaying Profile Parameters

It is a good idea to name a profile in a manner that makes it easy to tell what the profile does, like:

```
call_hq
listen_bost
```

If, however, you need to examine the parameters used by a profile, do one of the following:

If profile is currently loaded in memory

Once a profile is loaded using the restore profile command, you can display its parameters by using the following syntax:

```
> show profile profile_name
```

Example:

```
> show profile call_hq
```

A list of the profile parameters will display.

Reading from a disk file

You can see the parameters of profiles that are not currently in memory by using this syntax:

```
> show profile /file profile_name
```

Example:

```
> show profile /file call_hq
```

The file of the profile will be read and displayed.

Removing a Profile

This section describes how to do the following:

- Remove a profile from memory, but keep the copy on disk
- Delete a profile from disk

Removing a profile from memory

When you are no longer using a profile, you can unload it from memory by issuing a shell command with the following syntax:

```
> delete profile profile_name
```

Example:

```
> delete profile corporate
```

The profile corporate is no longer loaded in memory. Use the restore profile command to return it to memory

Deleting a profile from disk

You can change a profile at any time by following the instructions under *Editing an Existing Profile* on page 6-15. If, however, you no longer want a profile, you can delete it by following these steps:

```
> delete profile /file profile_name
```

Example:

```
> delete profile /file corporate
```

The profile called corporate will be deleted from the disk and no longer available for use.

What next

- To find out more about making connections, go to the *Connections* chapter beginning on page 6-1.
- To find out more about any of the commands referenced in this chapter, go to the *Shell Command Reference* beginning on page A-1.

chapter **7**

Connections

In this chapter

This chapter helps you learn about using the shell commands to connect to other sites. It discusses the following topics:

Starting Point

Before you can use the instructions in this chapter, the correct drivers must be loaded.

About Making Connections

- Use the instructions in this chapter to make a connection to a remote site. Once you make a connection to another site, you can use an application program to transfer data over the connection.
- Connections are made using the shell commands, from within the shell program. Use the pppsh or clsh commands to bring up the appropriate shell program.
- Although you can have multiple connection profiles, actual connections can be made one at a time. When you wish to access a different remote site than the one to which you are currently connected, you must disconnect and then connect to the other site.
- You should monitor your connections carefully to be sure that connections that are unnecessary do not remain up. Unexpected phone bills can result from leaving a connection up for long periods of time.

Connecting to a Remote Site

When you want to connect to a remote site, use the connect command in one of these ways:

- Using a connection profile name as an argument to the connect command.
- Using a phone number as argument to the connect command, (the default profile is used to supply the other connection parameters).
- Change specific parameters of either type of connection listed above by using qualifiers to the connect command, on the command line.

These connection methods are described next.

Connecting using a profile

You can create a connection profile that specifies all of the parameters for the connection that you want to make. It becomes very simple then to issue this command from the shell command line:

```
> connect profile_name
```

Example:

```
> connect diane_call
```

Since diane_call is a calling profile that does not use frame-activation, a call to the address specified in the profile will occur immediately.

The command would look like this in a DOS batch file:

```
clsh connect profile_name
```

Example:

```
clsh connect diane_call
```

Instructions on how to create connection profiles are provided in the chapter called *Connection Profiles* starting on page 6-1.

Connecting using a phone number

If you have not created a connection profile for a site that you wish to call, you can issue a connect command from the shell command line using the phone number of the remote site.

The default profile will be used for the other connection parameters required for a call. See page 6-2 to find out the parameter settings for the default profile.

```
> connect address
```

Example:

```
> connect 916125551212
```

A call will immediately be made to the phone number specified on the command line.

Changing connection parameters on the command line

There are a variety of reasons to change the parameters used for a connection. Depending on what you need to do, you can choose one of the following methods to make a connection using parameters that are not saved:

1. *Over-ride parameters set in a profile:*

   > connect *profile_name /param1 /param2 /param3 etc.*

   Example:

   ```
   > connect diane_call /local_ip=255.255.255.255
   ```

   On page 6-11, the profile diane_call was set up to request an IP address from the remote side. Using the command above, no negotiation is requested.

2. *Over-ride parameters in the* default *profile:*

   > connect *address /param1= /param2= /param3= etc.*

   Example:

   ```
   > connect 16125551212 /activation=frame /persistence=yes
   /idle_timer=30
   ```

   A connection using this command will not be made until there is actually data to send because frame-activation is specified on the command line.

The following are valid parameters to use on the command line:

/access=nailed_up|demand
/activation=manual|frame
/address= **“ ”**
/authentication=none|pap|chap
/channel=any|B1|B2
/connection_name=conn%d|name
/debug_level=0|2|9
/fallback=yes|no
/idle_timer=0|value_in_seconds
/listen=yes|no
/local_ip= “ip_address”
/name=profile_name
/number_of_channels=1|2
/password=
/persistence=yes|no|permanent
/remote_ip= “ip_address”
*/remote_name=******|*name*
/remote password=password_string
/type_of_channel=d64|d56|voice
/user_name=name_string

These arguments are also used by the set profile command. For a complete description of their use, see page A-8.

Disconnecting from a Remote Site

There are two ways that a disconnect from a remote site can occur:

- An automatic disconnect after a time-out value has been reached.
- A manual disconnect from the command line of the shell program

Using the time-out to disconnect

When you use a profile that specifies a time-out value, your connection will be disconnected after the specified length of time has passed without data to transfer. Using this feature can help keep connection costs down. See page A-10 in the *Shell Command Reference* appendix for information about the timeout parameter, or page 6-13 to find out more about setting up a profile that uses frame-activation, in which timeout plays a role.

Using the "disconnect" command

If you don't specify a timeout value in the profile that you use to connect to a remote site (and don't specify a timeout value on the command line), you must manually issue a disconnect once you are done transacting with the remote site. Use this command syntax to disconnect:

> disconnect *connection_name*

Example:

```
> disconnect conn0
```

The connection named conn0 will be terminated.

Note that you must use the *connection name* in this command. You can determine the connection name by following the instructions on the next page.

Displaying Connection Information

There are two commands that you can use to find out information about a current connection:

- Use the list connections command to find out the connection name, the state, and the name of the profile used to make the connection.
- Use the show connection command to get detailed information such as channel assignment and state. You must know the connection name (default is conn0 and conn1) to use this command.

Each of these commands is described next.

Using the "list connections" command

Use this command whenever you need to know the connection name (as with the disconnect command), or to get a quick view of the connection state. This is the command syntax:

```
> list connections
```

The connection name, the profile name, and the state (Active, Awaiting Establishment, etc.) will display.

Using the "show connection" command

Use this command to find out more detailed information about a connection. To use this command, you must know the connection name. Use list connections to find out the connection name.

> show connection *connection_name*

Example:

```
> show connection conn0
```

Detailed information about the connection named conn0 will be displayed.

If you also wish to see the parameters of the profile used to make the connection, use the command as shown:

> show connection *connection_name* /profile

Example:

```
> show connection conn0 /profile
```

Profile information will display along with the detailed information about the connection named conn0.

What Next

At this point you may wish to do the following:

- Go to the *Shell Command Reference* starting on page A-1 to find out more information about the commands described in this chapter.
- Create a Custom Profile using the instructions starting on page 6-1.

chapter **8**

Using the Trace Utility

In this chapter

This chapter provides instructions to help you perform a trace using the trace utility. It discusses the following topics:

Starting Point

To use the trace utility, you must bring up the shell program.

Trace Message Format

If you wish to have complete information concerning an interaction between the Digi ISDN board and the ISDN central office switch. The Digi ISDN software has a trace function that allows you to see all the D channel messages that pass between the adapter board and the ISDN switch.

Starting and Stopping a Trace

The trace function starts collecting information when the adapter board is powered up, and continues until a stop trace command is issued. After stopping, the trace can be started again by the start trace command. The adapter buffers as many lines as memory allows. The software will continue collecting information until it is stopped by using the stop trace command.

Trace Elements

The trace that the software prints to your screen may appear somewhat cryptic. However, each line represents an interaction between the board and the ISDN switch. Both low-level and high level messages may be transmitted on the D channel.

Here is an element-by-element breakdown of a frame obtained using the software's trace capability, with the trace decode level set at Layer 3. The first column is at the left of the screen as indicated below (Column numbers do not appear in the actual trace):

Example 1:

```
   1  2,3   4     5   6     7     8        9          10
00007 0N 00-064-R I PF=0 Ns=000 Nr=001 Crv=000(T) INFORMATION
       [EndPointID] 3B (02) 81 c0
```

Column	Description for Layer 3 Non-Information Frame	Description for a Layer 3 Information Frame
1	A sequential number for each line, starting with 0000, which identifies each message that passed between the adapter board and the switch. The number wraps around to 0000 after 32,767 lines.	← same
2	A number that represents the IDP module involved in the message. For your equipment, the number should always be 0.	← same

3	Either a `T` or an `N`, indicating the source of the message. A `T` message is sent by the adapter board, while `N` represents a message sent by the ISDN network.	⟵ same
4	Address field, consisting of three elements separated by dashes. The first two elements in the field identify the terminal adapter that the message is addressed to or sent from. The first element is the SAPI (Service Access Point Identifier), the second is the TEI (Terminal Endpoint Identifier), and the last element indicates Command or Response.	⟵ same
5	Frame type. There are various frames in ISDN signaling, including UI, UA, etc., and the frame type is indicated here. If the frame type is `I` (information) or `UI` (unnumbered information frame), the contents of columns 7 through 10 will be affected.	Information frame - If this column contains an I or UI the frame is an Information Frame and columns 7 through 10 must be interpreted using the descriptions in this column.
6	Poll Final field. D channel signaling uses a Poll Final bit to indicate whether it is answering a message, or that it requires a message.	⟵ same
7	Not Valid	Next send control I frame - indicates the number of the message that should be sent in the next message.
8	Not Valid	Next receive control I frame - indicates the number of the message that should be received in the next message.

9	Message meaning. An English translation of the meaning of the message is contained in this column.	Call reference value. A letter in parentheses follows the Call reference value number. If the letter is a T, it indicates a terminating (received) call - if it is an O, it means an originating (transmitted) call.
10	Not Valid	Message decode - meaning of the message in English. The unnumbered lines following an Information line is the raw information in hexadecimal numbers, along with some decoded messages. If the codes are unknown, the software indicates [Unknown] for the message.

Following some layer 3 messages, an information element line will be displayed as shown in Example 1. These elements have the following meaning:

Element	**Description**
First element	(from left) The PCIMAC software presents this element enclosed in square brackets. It is a decode of the "type" field (next element to the right in the line), to the extent the software can understand the code. If the software does not know the "type," it prints "<unknown>" in this element. Should you want to know the element decode in these cases, you can look it up in your ISDN switch reference manual.
Second element	Type field - the actual hexadecimal value for the field.
Third element	Field length - the length of the field in hexadecimal numbers.
Fourth element	Information (information or data) represented in hexadecimal numbers.

Example Negotiation

In the following example, a Layer 3 trace is depicted, step-by-step, as the ISDN adapter board negotiates with the ISDN switch for an automatic TEI:

Example 2:

```
  1  2,3   4     5  6   7   8      9        10
00000 0T 63-127-R UI PF=0          Id Req Ri:54358 Ai:127
```

A layer 2 message. The full meaning of the message is deciphered as follows:

Column 1: The line number is `0000`.

Column 2: The IDP module number is `0`.

Column 3: The “`T`” indicates that the message originated from the adapter board.

Column 4: SAPI is `63` indicating a management message, TEI is `127` indicating a broadcast for any TEI on the line, and the message is a Response (`R`) message.

Column 5: Frame type is `UI`

Column 6: Poll Final bit is set to `0`

Column 7: Blank

Column 8: Blank

Column 9: Message meaning is an ID request

Column 10: Message random number (`Ri`), and Action Indicator (`Ai`) set to `127` (broadcast TEI)

```
00001 0T 63-127-R UI  PF=0       Id Req   Ri:12345 Ai:127
```

A layer 2 message. Same as previous, for second TEI.

```
00002 0N 63-127-R UI PF=0     Id Assign Ri:12345 Ai:064
00003 0N 63-127-R UI PF=0     Id Assign Ri:54358 Ai:065
```

The ISDN network sends ID assignments for the new TEIs.

```
00004 0N 00-064-C UA    PF=1         Not Q931
```

The ISDN network has sent a layer 2 message to the board with a Poll Final bit set to `1`. Since the software's decode level is set to layer `3`, this layer 2 message is not decoded. Actually, this is a SABME transaction. At this point, layer 2 transactions are complete.

```
00005 0T 00-064-R I PF=0 Ns=000 Nr=000 Crv=000(T) INFORMATION
     [ServProfID] 3A (0A) 36 30 39 35 30 30 30 30 30 30
```

The board responded to the switch with an Information packet indicating its SPID. Note this is an `I` frame containing a layer 3 message.

```
00006 0T 00-065-R I PF=0 Ns=000 Nr=000 Crv=000(T) INFORMATION
```

```
    [ServProfID] 3A (0A) 36 31 39 35 30 30 30 30 30 30
```

The board next sent an Information packet indicating the SPID on the other B channel. Note that the TEI (column 4, element 2) is different from the previous message (`00005`), showing that each TEI has a unique SPID.

```
00007 0N 00-064-R I PF=0 Ns=000 Nr=001 Crv=000(T) INFORMATION

       [EndPointID] 3B (02) 81 c0

00008 0N 00-065-R I PF=0 Ns=000 Nr=001 Crv=000(T) INFORMATION

       [EndPointID] 3B (02) 82 c0
```

The network sent two messages to the board to assign USID/TIDs to the two channels.

Once this process is complete, the ISDN adapter board is recognized by the network as a terminal adapter that is capable of handling two B channels, each with its own TEI and SPID.

This sequence is representative of a configuration where the board has been connected to a DMS/100 ISDN switch. If the board is connected to a different switch the sequence of commands might be different.

Example Call Setup

This is an example layer 3 trace in which the ISDN adapter sets up a call:

Example 3:

```
  1    2,3     4       5    6     7      8        9          10
00015 0T 00-064-R  I  PF=0 Ns=007 Nr=010 Crv=001(T) SETUP

          [BearerCap ] 04 (02) 88 90
          [ChanIdent ] 18 (01) 83
          [KeypadFac ] 2C (04) 36 30 39 32
```

This is a call set-up packet sent by the adapter board. Note that the board has indicated the type of B channel `[BearerCap]` call that it wants, the B channel number `[ChanIdent]` (in this case the board requested any available channel), and is using a keypad to place the call `[KeypadFac]`.

```
00016 0N 00-064-R I PF=0 Ns=010 Nr=008 Crv=001(O) CALL PROCEEDING

          [ChanIdent ] 18 (01)  89
```

The switch has indicated to the board which B channel it will use for the call.

```
00017 0N 00-064-R I  PF=0 Ns=011 Nr=008 Crv=001(O) ALERTING
```

The switch is ringing the remote terminal adapter

```
00018 0N 00-064-R I  PF=0 Ns=012 Nr=008 Crv=001(O) CONNECT
```

The switch indicates that the call set up was successful, and the remote bridge answered the call.

```
00019 0T 00-064-R I  PF=0 Ns=008 Nr=013 Crv=001(T) CONNECTACKNOWLEDGE
```

The board signals the switch that it acknowledges that the call set up is completed.

chapter 9

Usage and Implementation Notes

In this chapter

This chapter describes several aspects of the DOS client driver that are not covered in other chapters. It discusses the following topics:

Using the Connection Manager Status Display

The Status Bar is useful when you want to monitor transactions over your ISDN line. From it you can get information about:

- Q.931 protocol states
- Call connection states
- Transmit and Receive Frame counts
- The presence or absence of traffic on both “B” channels

Turning the Status Bar ON/OFF

To toggle the Status Bar between ON/OFF states, simultaneously press the <Alt-Left Shift-D> keys. When it is ON you will see a small bar appear in the upper right-hand corner of your screen.

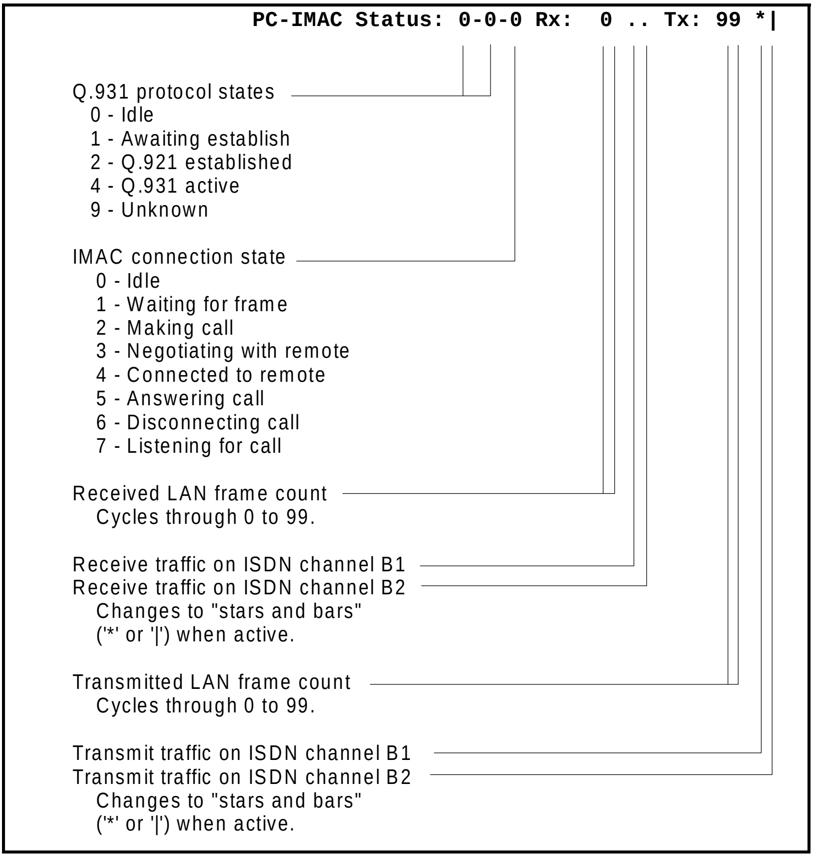
Status Bar Components

A representation of the bar and the meaning of the digits in the display is shown on the next page.

Notes on Status Display Usage

Keep in mind the following information as you work with the Status Bar:

- The Status Bar only works in text mode. If you have trouble seeing the Status Bar, consult the manual for the video card in your computer for information about changing to text mode.
- The Status Bar will not work in a Windows DOS Box or Full screen or Windows 95 DOS box.
- If your Status Bar appears “frozen” this may be because an application that you are running does not allow interrupts. Try unloading all applications.
- You can use the Status Bar for a quick update of line status, but for detailed status you should use the trace utility provided by Digi.



```
PC-IMAC Status: 0-0-0 Rx:  0 .. Tx: 99 *|
```

Q.931 protocol states
0 - Idle
1 - Awaiting establish
2 - Q.921 established
4 - Q.931 active
9 - Unknown

IMAC connection state
0 - Idle
1 - Waiting for frame
2 - Making call
3 - Negotiating with remote
4 - Connected to remote
5 - Answering call
6 - Disconnecting call
7 - Listening for call

Received LAN frame count
Cycles through 0 to 99.

Receive traffic on ISDN channel B1
Receive traffic on ISDN channel B2
Changes to "stars and bars"
('*' or '|') when active.

Transmitted LAN frame count
Cycles through 0 to 99.

Transmit traffic on ISDN channel B1
Transmit traffic on ISDN channel B2
Changes to "stars and bars"
('*' or '|') when active.

You can control the Connection Manager at load time with a statement in the IDP.CFG file:

```
status = on   (default if not present)
status = off
```

Or, you can control the Connection Manager at run time with a command to the *pppsh* or *clsh* program:

```
set status /display=on
set status /display=off
```

About PDUMP and the PPP Implementation

PPP Executables

The programs that implement PPP for this driver are:

PPP.EXE—implements the PPP core dependent on the arguments passed to it on the command line. These argument switches are: /bcp, /ipcp, /ipxcp.

PPPMAC.EXE—implements the PPP NDIS client driver and emulates an Ethernet NDIS driver.

PPPODI.EXE—implements the PPP ODI driver and emulates an Ethernet ODI driver.

PDUMP.EXE—manages and displays the PPP dumps and traces.

PPP switches:

/bcp	Use bridge encapsulation for all protocols.
/ipcp	Use IP protocol over PPP, blocking other protocols.
/ipxcp	Use IPX protocol over PPP, blocking other protocols.
/log	Allocates buffer space for packet trace and PPP logs, buffer size equals 64.
/log32	Allocates buffer space for packet trace and PPP logs, buffer size equals 32.
/log128	Allocates buffer space for packet trace and PPP logs, buffer size equals 128.
/biglog	Allocates buffer space for packet trace and PPP logs, buffer size equals 256.

Combination Options:

/bcp/ipcp-	Use IP protocol over PPP, use bridge encapsulation for all other protocols.
/bcp/ipxcp-	Use IPX protocol over PPP, use bridge encapsulation for all other protocols.

The bridging network protocol is enabled by default, equivalent to specifying /bcp. When /ipcp is used, IP and TCP traffic are diverted to the IP PPP encapsulation. Both /bcp and /ipcp can be specified simultaneously; specifying /ipcp alone turns off the default /bcp option.

When /ipcp is used local and remote IP addresses may appear in the PPP negotiation. Since a protocol stack above the NDIS interface is not aware of any such IP address negotiation, an address obtained from the peer must be reported to the protocol stack via either manual configuration or automatically through a special interface.

These IP addresses are reported as part of the call progress log. They can be displayed by executing the program `GETIP.EXE`.

The local IP address and the IP address of the remote PPP peer can be specified so that the same values configured into a third party TCP/IP implementation will be reported to the remote side if necessary.

Local IP address:

0.0.0.0	- request local address from peer
255.255.255.255	- do not negotiate address
other	- indicate local address to peer

Remote IP address:

255.255.255.255	- do not negotiate address
other	- supply remote address of peer to peer

PPP Multilink

PPP Multilink is automatically negotiated when an ISDN call is made with 2 B channels. If the peer answers only one call, and does not recognize the multilink negotiation, the call will automatically revert to a single channel non-multilink negotiation.

Using PDUMP

The shell program reports many details of PPP negotiations via the call progress log. If compatibility problems with third party products occur, or a failed negotiation can not be diagnosed from the shell log, the PDUMP program can access an event log and a trace log which Digi Technical Support can use to help resolve PPP negotiation failures. The event log records the states, transitions, and actions of the PPP protocols. The trace log records the first 64 bytes of all PPP frames sent or received by the PPP core. The PDUMP program turns packet tracing on and off. Space for these logs is reserved by the PPP /log option. Although the event log is always active, it holds very few entries without the /log option. The packet log is off by default.

Output from PDUMP can be directed to a file for later analysis.

Interface Manager Buffer Settings

Since the Digi driver is a TSR, you may wish to decrease the amount of memory used by the driver to avoid problems with other programs loaded into memory or increase the memory used in order to optimize the operation of the driver.

How to change the number of buffers used

You can change the number of buffers used by the driver by using a command line argument when either the IDPIM.EXE (PC-IMAC) or ADPIM.EXE (DataFire) driver module is loaded.

For example:

```
ADPIM /b30
```

The b/30 flag causes 30 buffers to be used instead of the default.

Buffer values

Valid buffer number values are 12 to 68.

Buffer default value = 24

- If you specify a number smaller than 12, the driver will use the minimum value of 12, and if you specify a number greater than 68, the driver will still use the maximum value of 68.
- Most applications can use the default buffer setting. If you need to change the buffer value, increment or decrement the buffer value by small values; often an increment of just one will suffice.
- If you experience driver load failures or have problems reading the data to be transferred, you can attempt to correct the problem by increasing the number of buffers used by the driver interface manager.

Unloading the Driver from Memory

Requirement

Since Digi's DOS client driver is a TSR (Terminate and Stay Resident) program, when you no longer want to use the Digi driver and wish to free up the memory it is using you must unload all of the driver components *in the exact order in which they were loaded*.

Using the Digi Unload batch file

Digi has supplied a batch file that will unload the DOS client driver components from memory. To use it, type the following on the command line:

```
C:\> digi_rmv
```

Finding out the load order

If you don't want to use the Digi batch file to unload the driver files, reference the batch file you used to load them, and then unload them in the reverse order, in a manner similar to this:

```
ipxodi /u
pppodi /u
idpodi /u
```

etc.

appendix A

Shell Command Reference

In this chapter

In this appendix you will find information that will help you use the driver shell programs. It discusses the following topics:

Using the Shell

Use the driver shell to create calling profiles or to manually connect to a remote site.

Invoking the shell

There are two shells: pppsh and clsh:

- Use *pppsh* with the drivers for a PPP connection.
- Use *clsh* with the “Classic” drivers.

To bring up the shell type the following at your DOS prompt:

C:\> **pppsh** <Enter>

or

C:\> **clsh** <Enter>

You will see a command line prompt like the following:

```
C:\ppp>
```

or

```
C:\cl>
```

Command Syntax

The general syntax of the shell programs is:

command [argument] [*/qualifie*r[=*value*]]

where

The argument often is a profile or connection name or a phone number.

The qualifiers often narrow or direct the action of the command, and the value further modifies or directs the action.

Abbreviations

The commands can be abbreviated to the shortest non-ambiguous string within the context of the command.

Examples:

```
list defines /total
```

can be abbreviated: `list d /t`

```
type trace /from=4 /to=6 /filter=none /decode=none /device=all
```

can be abbreviated: `t tr /fr=4 /t=6 /fi=n /dec=n /dev=a`

Command Line Editor

As you use the shell program, the following keystrokes can be used on the command line:

<Home	Go to beginning of current line
<End>	Go to end of current line
<←>	Move cursor left (backward) one position
<→>	Move cursor right (forward) one position
<Del>	Deletes character under the cursor
<Backspace>	Deletes character to the left of cursor
<Esc>	Delete entire line
<↑>	Recall previous line (last 20 are stored)
<↓>	Recall next line

Shell Commands

Shell commands for both *clsh* and *pppsh* are described in this section. You must run *pppsh* to use commands that are labelled ***PPP only***.

Note: Values in bold indicate a default.

connect

You must provide either a valid profile name or valid phone number to attempt a connection. You may use any of the parameters described under the ***set profile*** command to change a parameter on the command line.

connect *phone_number*

Uses the default connection parameters to call the phone number you provide on the command line. Default values are overridden by command line arguments.

connect *profile_name*

Uses the connection parameters of the profile you name on the command line. Profile values are overridden by command line arguments.

Examples:

```
connect call_boston /user_name=accounting
```

Uses the *profile_name* of call_boston along with the */user_name* qualifier.

```
connect 5551212 /channel=any
```

Uses the phone_number of **555-1212** and the */channel* qualifier.

copy profile

Copies the profile values of one profile into a new profile.

copy profile *profile_name new_profile_name*

/file

Causes a search for the source profile on disk.

Example:

copy profile default server1

Profile to copy is **default**; name of new profile is **server1**.

define

Used in batch programming.

define *name value*

Creates a name/value pair definition. A name/value pair is a statement that equates an arbitrary value to a definition or set of values, parameters, etc.that is used to modify the board's behavior. Definitions are saved on disk. If a definition for *name* already exists, it is replaced with the new definition.

define alias.*short_command "long_command"*

Use to shorten the name of a command string to execute. A command string of up to 79 characters, consisting of a command and its arguments can be given a single word alias of 31 characters or less. Quotes are required around the *long_command* argument.

Example:

```
define alias.cls “set term /clear”
```

When you type `cls`, the terminal screen will clear.

define prompt *new_prompt*

Used to create a new prompt in place of the default (`pppsh>` or `clsh>`).

delete profile

Deletes a profile from either memory or disk.

delete profile *profile_name*

Removes the profile from memory, but leaves it on disk.

/file

Causes the profile to be deleted from disk.

disconnect

disconnect *connection_name*

Tears down the connection named as an argument. All channels used by the connection are freed. Use the **list connections** command to find out the connection names.

echo

Used for batch programming.

execute

Used for batch programming.

/from=line_number

/to=line_number

checklayer3

Waits until layer 3 is up.

list

list connections

Lists all currently established connections by name, related profile, and state.

list defines

Lists all definition pairs currently defined in the system.

/total

Displays only a line containing the total number of definition pairs and the amount of memory they use.

/mask=name

Displays only definition pairs whose names match the mask. Wildcard characters (* ?) are recognized.

list profiles

Lists the names of all profiles currently available in system memory.

/file

Lists profiles on disk as well as those in system memory.

pause

Used in shell batch files, stops command execution until a key is pressed.

quit

Causes a graceful exit from the shell.

repeat

repeat *command*

Repeats the execution of a command. If no qualifier is used, the command is repeated until a key is pressed. Command strings may be up to 79 characters.

> */sleep=seconds*
>
> Pause between command repetitions the number of seconds indicated. Using the */sleep* argument (with no = value) defaults to a value of 2 seconds.
>
> */times=count*
>
> Repeat the command the number of times indicated.
>
> */terminal=clear|home|none*
>
> Action to be taken before command execution: clear will clear the console screen with cursor at upper left corner; home places the cursor at upper left corner without clearing screen.

reset trace

Halts any trace in progress and clears the trace buffer.

restore profile

restore profile *profile_name*

Causes the profile to be retrieved from disk and stored in memory. If a copy exists in memory, this command will overwrite it from the disk file.

run

Used in shell batch files.

save profile

save profile *profile_name*

Causes the profile to be saved to a disk file. Use this command to save the profile you are working with before exiting the shell program.

set

set compression

Controls negotiation for IDP compression.

> */negotiation=**on**|off*

Controls whether IDP compression will be attempted on future connections. Both sides of a connection must enable this option for it to work. Does not apply to an existing connection.

set profile *profile_name /param1= /param2= /param3= and so on*

Sets the connection parameters for a particular profile. Valid parameter options are:

*/access=nailed_up|**demand***

Defines whether connection setup procedures are applied to the connection; *nailed_up* defines the connection as dedicated (like a leased line), requiring no setup; *demand* requires normal connection setup.

*/activation=**manual**|frame*

Used for outgoing calls only, selects whether a connection occurs at the time a *connect* command is issued (*manual*), or is delayed until a frame is ready to be transferred (*frame*).

If you choose an activation method of *frame*, and also configure the profile for */persistence=yes* and provide a non-zero value for */idle_timer*, your profile will connect when data is ready to transfer, wait the amount of time specified for *idle_timer*, disconnect, then reconnect when additional data is to be transferred.

See */idle_timer* and */persistence* for further information.

/address= **“ ”** *or /address=address1:address2 (for 2-channel calls)*

Supplies the phone number to use when calling out. Up to 63 characters can be used, including * and #. A valid phone number must be supplied for an outgoing call, but this field should be blank for incoming calls.

Use the second form of syntax when the remote device has two phone numbers, in order to ensure a 2-channel call. If the remote side has only one phone number, you should repeat it.
Example: /address=address1:address1

This parameter must be used along with the */number_of_channels* and */channel* parameters to initiate a 2-channel call. See their descriptions for more information.

If sub-addressing is used, the address is followed by an “*” and the subaddress. *Example*: /address=5551234*234

Indicates a telephone number of 555-1234 with a sub-address of 234.

*/authentication=**none**|pap|chap* ***PPP only***

Type of authentication chosen locally to authenticate a remote site.

Choosing *none* means that no authentication will occur locally.

Choosing *PAP* (Password Authentication Protocol) means that the local end, as the authenticator, will require the remote side to supply the correct *user_name* and *remote_password.*

Choosing *CHAP* (Challenge Handshake Authentication Protocol) means that the local side as authenticator requires the remote side to respond to a repeatedly varying challenge message with a response that is very difficult to compute without knowing a shared secret. This is much stronger than PAP since the secret is never sent, but uses *password* as the secret.

*/channel=**any**|B1|B2 or /channel=B1|B2:B1|B2 (for 2-channel calls)*

Defines which bearer channel to use for the connection; *any* allows the next available channel to be used; use of other possible values force use of that channel for a connection. Other valid values: *b1, b2, b1a, b2a, bx, bxa*. Choose a value based on the available B channels and Logical Terminals at your site, according to the values shown in this table:

Value	B Channel Used	Logical Terminal Used
any	next available B channel	next available logical terminal
bx	next available B channel	logical terminal 1
bxa	next available B channel	logical terminal 2
b1	B channel 1	logical terminal 1
b1a	B channel 1	logical terminal 2
b2	B channel 2	logical terminal 1
b2a	B channel 2	logical terminal 2

If locally you have two phone numbers, you should use a separate B channel and Logical Terminal for each number.
Example: /channel=b1:b2a

If locally you have only one phone number, you must use the same Logical Terminals for both B channels. *Example:* /channel=b1:b2

See the parameter definition for */address* for information about supplying the correct phone numbers for the remote site.

*/connection_name=**conn%d**|name*

Use this option to assign a meaningful name to the connections you make using this profile. For example, naming the connection after a location or a server lets you know immediately who you are connected to when you examine status. If you do not specify a name, the default *conn%d* will be used and the %d replaced by the value *0*.

*/debug_level=0|**2**|9*

Sets the level of messages sent to the console. A setting of *0* sends only command syntax errors; a setting of *2* also includes call processing and explanation messages; a setting of *9* sends all messages to the console.

*/fallback=**yes**|no*

Use this option to cause the adapter to continue to attempt a connection using progressively slower speeds than it is set to until a connection is achieved. Set this option to *no* if you don't want this type of attempt to be made.

*/idle_timer=**0**|value_in_seconds*

Number of seconds of no activity to pass before the connection is timed-out by the Digi adapter. A value of 0 disables the timer. A negative value causes the timer to only count data transfers initiated locally when determining no activity.

Use this option with the */persistence* option, which sets up the behavior of the adapter once a disconnect has occurred after a time-out has been reached, and the */activation* option which sets a condition for reconnect.

/listen=yes|no

Set this option to *yes* to create a listening profile—a profile that waits for incoming calls; set it to *no* for a calling profile—a profile that calls the number specified with the */address* argument.

/local_ip= "ip_address" or "0.0.0.0" or "255.255.255.255" ***PPP only***

Announces the indicated value as the local IP address during negotiations with the remote connection. Use 0.0.0.0 to request a local address from the remote side. Use 255.255.255.255 to keep address negotiation from occurring. Address must be enclosed in quotes.

When you use the /ipcp argument as you invoke the Digi PPP core driver, IP encapsulation will be negotiated over the connection. Digi's PPP core driver supports the provision of IP addresses *to* the remote end, or requesting an IP address *from* the remote end. In this way, the local side can be either the supplier or the recipient of IP address information. The default is to do neither.

Note: If a local IP address is dynamically assigned, or if the remote side announces its IP address, the values must be provided to the IP protocol stack on the local machine. The current dynamic values can be determined using Digi's GETIP.EXE program residing in the driver directory.

/name=profile_name

Assigns the provided name to the current profile. Names may be up to 31 characters in length.

*/number_of_channels=**1**|2*

Defines the number of channels the connection will use; use this parameter in conjunction with the */channel* parameter to initiate 2-channel calls.

/password= ***PPP only***

Password to be supplied to the remote side if the remote side asks for authentication. CHAP authentication uses this as the “secret”.

*/persistence=yes|**no**|permanent*

Defines how the driver responds when a disconnect occurs:

- Setting persistence to *permanent* causes the driver to attempt a reconnect after four seconds.
- If persistence is set to *no,* when a disconnect occurs no further action is taken.
- A listening profile with persistence set to *yes* will revert to listening mode when a timeout occurs.
- Also, if persistence is set to *yes,* and if */activation=frame,* setting persistence to *yes* allows the driver to re-connect when a frame is ready to be transferred. If you use this setting, be sure to monitor your setup to ensure that excessive connections are not made that will result in excessive connect charges from your service provider.

See the descriptions under */activation* and */idle_timer* for further information.

/remote_ip= “ip_address” or “255.255.255.255” ***PPP only***

Supplies the indicated address as the remote IP address if requested by the remote site. Use 255.255.255.255 to keep address negotiation from occurring. Address must be enclosed in quotes.

See the information under */local_ip* for more information about IP addressing.

*/remote_name=*****|name*

If you use the wildcard (*) for this option, any remote device can connect or be connect to your system when you use this profile. You can also use the character wildcard (?) in the name.

This option is analogous to the PAP authentication used with PPP and can be used with the “Classic” drivers to perform much the same function. At the start of a connection, local and remote drivers exchange profile names and look for a wildcard or complete match between the stored profile name and the name offered from the other side. Each side checks for this match and disconnects if not satisfied.

/remote password=password_string ***PPP only***

Password which the remote side must supply if the local side requires authentication. CHAP authentication uses this as the “secret”.

*/type_of_channel=**d64**|d56|voice*

Defines the quality of service available for use in the connection. Select *d64* if 64K bps service is available; select *d56* if 56K bps is available; select *voice* if voice-grade channels are available.

/user_name=name_string ***PPP only***

Name to be supplied to the remote side of a connection if the remote side asks for authentication.

set terminal

Controls terminal parameters.

/clear

Clears the screen.

/home

Homes the cursor to the upper left corner of the screen.

term.clear

Sequence to clear from the cursor position to the end of the screen. Cursor does not move.

term.home

Sequence to home the cursor without clearing the screen.

term.kill

Sequence to delete all characters from the cursor position to the end of the line.

term.left

Sequence to move cursor one position to the left.

term.right

Sequence to move the cursor one position to the right

The values for these definitions are hex strings of the following format: “xx:xx:xx: . . xx:xx”

Example:

```
define term.kill “1b:5b:4b”
```

set trace

Defines the characteristics and behavior of a trace, even if in progress.

*/filter=none|L2|**L3***

Defines the level of filtering to use during a trace:

none	No filtering, all frames saved
L2	Only LAPD/Q.921 frames will be saved
L3	Only LAPD/Q9.31 frames will be saved

*/decode=none|L2|L2_data|L3|L3_data|**L3_elem***

Defines the type of interpretation used when trace data is displayed:

none	Hexadecimal dump
L2	Decodes and displays LAPD/Q.921 frame headers
L2_data	Adds up the user data area in hex
L3	Decodes and displays L2/L3 Q9.31 frame headers, message types and call reference values
L3_data	Adds up the user data area in hex
L3_elem	Adds up the user data area on Q.931 packets, decoded as Q.931 elements

set status

Controls display of the Connection Manager.

*/display=**on**|off*

Sets the Connection Manager on. This setting is overridden by the configuration you choose during installation that is stored in a configuration file.

show

show compression

Shows whether compression negotiation is enabled.

show connection *name*

Show detailed information about a connection including state and channel assignment.

/profile

Also displays the connection parameters for the profile that initiated the connection.

show define *name*

Displays the indicated name/value pair. This command is equivalent to **list defines** */mask=name.*

show idp

Provides information about the interface.

/L1

Shows S/T (I.430) interface status screen.

/L2

Shows LAPD layer (Q.921) status screen.

/L3

Shows Network layer (Q.931) status screen.

Area=10|11|12|21|22

10=Show HDLC channel statistics screen for the D channel
11=Show HDLC channel statistics screen for B channel B1
12=Show HDLC channel statistics screen for B channel B2
21=Show compression statistics screen for B channel B1
22=Show compression statistics screen for B channel B2

show profile *profile_name*

Looks in memory for the indicated profile and lists the parameter settings for the profile.

/file

Looks on disk for the indicated profile.

show trace

Prints out the characteristics of the trace as set by the set trace command, and reports the number of accumulated frames in the buffer along with related information.

sleep

sleep *number*

Suspends command execution for the specified number of seconds, or if a negative value is supplied for *number*, suspends until a key is pressed.

start trace

Causes the trace facility to begin executing according to the characteristics setup with the set trace command.

stop trace

Causes the trace facility to cease data collection.

type trace

Displays the entire contents of the trace buffer to the console.

/entry=line number

Displays only the entry at the line number provided.

/from=line number

Begins displaying entries at the line number provided.

/to=line number

Displays entries up to the line number provided.

/filter=none|L2|L3

Defines which frames to display:

none= no filtering, all frames displayed
L2 = only LAPD/Q.921 frames will be displayed
L3= only LAPD/Q9.31 frames will be displayed

/decode=

Defines the type of interpretation used when trace data is displayed:

none ....Hexadecimal dump
L2...............Decodes and displays LAPD/Q.921 frame headers
L2_data......Adds up the user data area in hex
L3...............Decodes and displays L2/L3 Q9.31 frame headers, message types and call reference values
L3_data......Adds up the user data area in hex
L3_elem .....Adds up the user data area on Q.931 packets, decoded as Q.931 elements

This argument overrides the */decode* parameter used with the **set trace** command.

undefine

undefine *name*

Deletes the indicated definition pair.

Creating Shell Batch Files

Within the shells, you can execute multiple commands with one command by creating a batch file.

About Batch files

- Batch operations are composed of individual command lines, which are executed in sequence.
- Lines in a batch are numbered sequentially from 0 to 32767.
- Commands strings must be surrounded by quotation marks.

Creating or editing batch files

To edit an existing batch file, or to create a new batch file and enter shell commands into it, use the **define** command as follows:

define *batch_name.batch_line_number "shell_command"*

where:

batch_name is the name of the new or existing batch file

batch_line_number is the line of the file with the command you want to change or add

shell_command is the command (with any parameters or arguments) to be executed (up to 70 characters)

Example:

```
define init.0 “reset trace”
```

This line creates a batch file called "init" with a reset trace as the first command. Additional lines can be added to form a complete routine:

```
define init.1 “set trace /filter=L3”
define init.2 “start trace”
define init.3 “conn 5551212”
define init.4 “stop t trace”
```

When executed, this routine will first reset the trace feature, then set the trace filter to Layer 3, start a trace, make a connection to phone number 555-1212, and then stop the trace once a connection is made.

Executing a batch file

To execute a batch file, use the **execute** command as follows:

execute *batch_name*

where:

batch_name is the name of the batch file to execute.

Example:

```
execute init
```

This command causes each line of the batch file to be executed in order.You may use the repeat command to cause a batch file to repeat as many times as you request, and you may use the /to and /from qualifiers for the execute command. These are explained in more detail on page A-6.

Batch commands are not parsed until execution. Any errors in the command will generate error messages at that time. It is best to test batches to ensure that your adapter will respond the way you wish it to.

appendix ***B***

Troubleshooting and Tech Tips

In this chapter

In this appendix you will find information that will help you setup your Digi ISDN adapter and make it work in your environment. It discusses the following topics:

Troubleshooting

Tips for Finding Problems

These are tips that will help you narrow down what might be wrong with your installation:

- Perform tests at each stage of installation as described in the appropriate chapter.
- After installing driver components and configuring the board for the line, check the Connection Manager

Testing Layer Connections

To find out if there is proper communication to your ISDN provider, you can check the status of the 3 ISDN layers:

- Layer 1 shows the physical (S/T) status
- Layer 2 is the LAPD (Q.921) status
- Layer 3 is Network Layer (Q.931) status

It is helpful to understand which layer is experiencing problems in order to help diagnose what the problem is.

To test the layers, you must use either the *clsh* or *pppsh* shell program. Instructions for use of the shell programs and a complete description of all the commands are provided in Appendix A.

Layer 1

This is a test to see if there is proper communication to the physical layer, either directly to the ISDN line, or through an NT1.

Command: sh idp /L1

You will see Layer 1 status similar to the following:

```
Tx_Info_Pattern          INFO 3
Rx_Info_Pattern          INFO 4
Tx_Frames                0
Tx_Collisions            0
Tx_Errors                0
Rx_Frames                0
Rx_Errors                0
```

Layer 1 status is good if:

- **Tx_Info_Pattern= INFO3**
  INFO3 means “activated”
  Other possible values:
  INFO 0 — no line is connected
  INFO 1— “in activation”
- **RX_Info_Pattern = INFO 4**
  INFO 4 means “activated”
  Other possible values:
  INFO 0— no line connected
  INFO 2 — in activation
  Lost Framing — was activated and lost it

If Layer 1 status is good, go on to test Layer 2.

If you see any other status, Layer 1 is not up. You should check the following:

- Ensure that the driver loaded with no errors.
- Check cabling. Use cables that came from Digi or find another cable with which to test. A standard RJ11 phone cable will work. Unplug all cabling, wait a few seconds, plug back in, and retest.
- Check for error lights on your NT1. Try another NT1 if possible.

Layer 2

This is a test to see if the requested TEI (terminal endpoint identifier) assignment from the switch was successful. The TEI is assigned dynamically, and if passes will assign a number from 64-126. If you reboot or lose the ISDN connection to the switch the TEI will be reassigned.

Command: sh idp /L2

You will see Layer 2 status similar to the following:

```
State                   Tei Assigned
Sapi                    0
Tei                     96
Lap_State               Multiframe Established
Lap_V(S)                1
Lap_V(A)                1
Lap_V(R)                46
I_Queue_Frames          0
Ack_Queue_Frames        0
```

Layer 2 passes if:

- **State = Tei Assigned**
  Tei Assigned means the switch assigned a TEI value
  Other possible values:
  Awaiting Tei Assignment —waiting for assignment of Tei
  Released — TEI released, in recovery
- **Tei = Number from 64-126**
  If you see a TEI of 127, it means that a TEI is being requested from the Switch and has not yet been assigned. Without the TEI, calls are not possible.
- **Lap_State = Multiframe Established**
  Multiframe Established means operating mode is normal
  Other possible values:
  Initialized — just started, no TEI yet
  TEI Assigned — LAPD has TEI, just reported to LAP
  Awaiting Multiframe Establishment — waiting for UA
  Awaiting Multiframe Release — DISC received
  Timer Recovery — time-out recovery from link errors

If Layer 2 status is good, go on to test Layer 3.

Causes for waiting TEI assignment:

- Switch busy, needed resources
- Connection not available yet (customer's switch programming not done yet).

Try unplugging the RJ cables from the wall, NT1, and Digi ISDN board, wait a few seconds, plug back in, and retest.

Layer 3

This is a test to see if full service is available from the ISDN switch. It will check to see if the switch type entered during Digi driver configuration is correct. It will also check the SPIDs (typically, only used in the U.S.) against the values setup at the ISDN service provider. These values must match exactly; it is just like a password being approved.

Command: sh idp /L3

You will see Layer 3 status similar to the following:

```
State                 Active
Tei                   96
Style                 National ISDN 1
Spid                  61298858901111
Usid                   0
Tid                   11
Service               Full
Style_Flags            0
Connection_U_State    10
Connection_Crv        1
Connection_Flags      0
```

Note: If you have 2 SPIDs you will see the above fields shown twice, the first time for the 1st SPID, then again for the 2nd SPID. Check both because you can have a problem with 1 B channel.

Layer 3 passes if:

- **State = Active, or Established, Awaiting Restart**
  Active means operational and *Established, Awaiting Restart* means operational with no Restart received
  Other possible values:
  Initialized — just created
  Awaiting Layer 2 Establishment — waiting for LAPD to initialize,
  Restart Pending, Awaiting Ack — Restart sent, waiting for response
- **Service = Full, or Non Initializing**
  Full means all subscribed features are available and *Non-Initializing* means the line is non-initializing
  Other possible values
  Not Determined— no line initialization performed
  Restricted — default service is restricted
  Rejected, Bad or No SPID — terminal reject by switch, no service

If you see any other status, you should check the following:

- Check that you used the correct switch type. The switch type is entered during driver configuration and is meant to describe the *software* used by the switch. Here are some tips to picking the correct switch type.

NI-1- Pick National ISDN 1 if your switch is National ISDN or the protocol running on it is National ISDN.

NTI - Pick Northern Telecom ISDN if your switch is a DMS100 (running their Functional protocol) pick NI-1 if running National ISDN.

AT&T- Pick this if the switch is AT&T running their Custom protocol, but pick NI-1 if switch is running National ISDN.

NET3 - Standard international switch type

EuroISDN - Identical to NET3

VN4 - Identical to NET3

1TR6 - ITR6 (Germany)

VN3 - VN3 (France)

INS64 - INS64 (Japan)

AUSTEL - Australia

SINGAPORE - Singapore

- Check that the SPIDs (typically only used in the U.S.) you entered during Digi driver configuration are correct (no spaces, no dashes, just one long number). You may need to reconfirm the address information with your ISDN service provider.

Tech Tips

These are answers to common questions that the help desk receives. You may find the answer to your problem here. You can also access Digi’s web page at **www.dgii.com** for additional information.

Q: To what value should I set my DataFire I/O port address?

The best I/O address to use is 350H. If this does not work, try other addresses in the 300 range, then the 100 range. Leave those in the 100 range as a last resort because a number of systems and peripherals default to values in that range.

Q: What is an NT1 and why do I need one?

An NT1 (network terminator 1) is a device which provides an interface between the two-wire twisted pairs used by telephone companies in their ISDN Basic Rate (BRI) network and an end-user's four-wire terminal equipment. Any Digi ISDN product with a S/T interface will require an external NT1. Any Digi ISDN product with a U interface will already have a built-in NT1.

In North America, you have to buy and maintain your own NT1 device. The telephone company offers end-users a U interface. In Europe and Japan, the telephone company provides the NT1, owns it, and offers end-users a S/T interface directly. In North America, some ISDN equipment vendors offer devices which connect directly to the U interface (for example, the Datafire U). If you have one of these devices, you don't need to buy a separate NT1. The U interface can't be built in to the device when it's offered for sale in Europe or Japan.

Q: How do I set termination on the DataFire adapter?

This applies only to the S/T models of DataFire adapters that are used with an NT1 (from a third party vendor).The DS2 dip switch sets Line Termination. The factory setting of DS2 on all S/Tmodels of DataFire adapters is down, by default this places 100 ohm terminating resistors across the transmit and receive lines. In the event that another device is installed on the same BRI line, (like a handset) only one device should be terminated. If the other device's termination can not be easily disabled, reposition the DS2 dip switch; up disables line termination.

On a Datafire S/T, Dip Switch DS2 has four switches, two for each line. To remove termination from line 1, place Switches 1 & 2 in the UP position, for line 2, place Switches 3 & 4 in the UP position.

On a Datafire S/T 4 Switches 7 & 8 are used for line 1, Switches 5 & 6 for line 2, Switches 3 & 4 for line 3, Switches 1 & 2 for line 4.

Q: To what values should I set my PC-IMAC I/O port address and memory address?

The best I/O address to use is 320H.If this does not work, try other addresses in the 300 range, then the 200 range. Leave those in the 100 range as a last resort because a number of systems and peripherals default to values in that range.

The best memory address is D0000H. Next try other addresses in the D0000H range, then the E0000H range, then C8000-CFFFFH range.

Q: How do I test to see if the adapter is talking to the switch?

Run diagnostics on layers 1-3 to find out if they are passing correctly.

See page B-2 for information about how to test the layers.

Q: I have performed the tests on the layers, now how do I test my line?

If you need help, Digi's has several ISDN test systems that you can call. For details on how to access these test systems, contact Digi Technical Support.

Q: What should I check if I'm having trouble calling out?

- Treat the ISDN call as if it were a phone call; use no spaces, dashes, or commas.
- If local call, just use seven digits. *Example:* `conn 9886928`
- If long distance, use 1, area code, and then phone number. *Example:* `conn 16129886928`
- If long distance, try another carrier prefix: MCI 10222, Sprint 10333, AT&T 10288. Add prefix number to phone number. *Example:* `conn 1022216129886928` (This will call number using MCI)
- If using centrex or pbx, you may need to put a 9 at beginning for outside call.
- If problems using 64k for call, drop down to 56k. (Not all areas or the routes support 64k clear channel calling)
- If problem is with a 2 channel call, drop down to one channel.

If still having problems, write down error and call Tech Support.

Q: What should I check if I'm having trouble calling in?

Run Digi DOS install.exe program. Go to Configuration -> ISDN Line Parameters -> Check Address field. This should be *your* ISDN phone number, not the calling side. It must be one long number no spaces or dashes. In the U.S. it is usually the SPID minus any prefix or suffix.

If still having problems, write down the error and call Tech Support.

Q: I get an “incomplete address” error when I call out. Why?

Try adding # (pound symbol) to end of phone number and enclose in quotes.

Example: conn “16129886938#”

Q: Windows applications don’t work, but NetWare works great, why?

Make sure you are running the latest DOS client driver. Try changing the Software Interrupt Vector from default. Run install, select Configuration -> Adapter Parameters Change Software Interrupt Vector from 62 to 78 or 79. Save, reboot, and reload drivers.

Q: Why do I get an “interworkin, unspecified” error when I try to call out?

Try adding a 9 prefix to the phone number when calling out.

Q: Why do I get an “protocol error, unspecified” error when I try to call out?

Try adding a 9 prefix to the phone number when calling out

Q: What should *net.cfg* look like?

See example below. This example is the same as the sample file (DIGINET.CFG) provided with your driver:

```
REM -------------------------------------------------------------------
REM Digi International DataFire and PC-IMAC DIGINET.CFG File Version 1.00
REM Copyright (c) 1995-96 by Digi International, Inc. All rights reserved.
REM -------------------------------------------------------------------

REM  This file provides an example of a NET.CFG configuration file used
REM  by Novell's LSL.COM and IPXODI.COM.

REM  Comment out unused Frame types by starting line with ; (semi-colon)
REM  Comment out unused Protocol types by starting line with ; (semi-colon)
REM  Change INT to match your Digi ISDN board's interrupt
REM  Change PORT to match your Digi ISDN board's I/O port address
REM  Change MEM to match your Digi ISDN board's memory address (PC/IMAC only)
REM  Change FIRST NETWORK DRIVE to appropriate driver letter (Usually F)
REM  Change PREFERRED SERVER to match your Netware server name

REM  Remember: For improved performance you should use latest VLM.EXE
REM  Remember: If using VLM.EXE you must have LAST DRIVE=Z in CONFIG.SYS
REM  Remember: NET.CFG should be in same directory as LSL.COM & IPXODI.COM

Link Driver IDPODI or PPPODI
        Frame Ethernet_802.3
        Frame Ethernet_802.2
        Frame Ethernet_II
        Protocol IPX 0 Ethernet_802.3
        Protocol IPX E0 Ethernet_802.2
        Protocol IPX 8137 Ethernet_II
        Protocol IP 0800 Ethernet_II
        INT 12
        PORT 320
        MEM D0000

NetWare DOS Requester
        FIRST NETWORK DRIVE = F
        PREFERRED SERVER=OPUS
```

Q: I have an ISDN connection from the DOS client system to a Novell server but VLM doesn't find the server.

- Make sure the correct *net.cfg* is being used. Duplicate copies can cause problems. When *lsl* loads it will tell you from which directory *net.cfg* is being called.
- Make sure correct frame type is being bound. This can be done by watching to see what gets bound after loading *idpodi.com*. If this does not match what is bound to the Digi ISDN board in the Novell server, you can not attach to the server. If binding to wrong protocol check *net.cfg* file.
- Make sure you are running the latest drivers. This applies to both Digi ISDN DOS client and Novell client (VLMs and associated files). Also make sure all Digi ISDN drivers are in their own directory, and Novell client files are in their own directory, both with proper path statements in *autoexec.bat*.
- Try using the preferred server option for connecting to server. Example - vlm /ps=servername

Q: A basic connection works with my DataFire U, but fails if I try to use a batch file. Why?

After the drivers load with a Datafire U, it takes about 30 seconds for the built-in NT1 to sync up with the ISDN switch. (With a PC/IMAC or Datafire S/T, the NT1 is external and already synced up).

The solution is to add a delay to the batch file. After the drivers are loaded, add *clsh sleep 30* or *pppsh sleep 30*.

Example:

```
adpim.exe
         .
         .
 ipxodi.exe
 clsh sleep 30
 connect "profile-name"
  vlm
```

Q: Using a batch file to load drivers and automatically connect works fine with a PC/IMAC but fails with a Datafire U.

The NT1 on the Datafire U takes 25 seconds after drivers load to properly sync with the ISDN switch. Since the PC/IMAC uses an external NT1 it is already synced up and just waiting for the drivers to be loaded. The fix is to add a 30 second pause before executing the "conn profile" line.

See example above.

Index