CODEX

4000 Series LAN Administrator's Guide

CODEX

4000 Series LAN Administrator's Guide

Notice

Proprietary Material

Information and descriptions contained herein are the property of Codex Corporation. Such information and descriptions may not be copied or reproduced by any means, or disseminated or distributed without the express prior written permission of Codex Corporation, Mansfield, Massachusetts 02048.

This document is for information purposes only and is subject to change without notice.

User/Reader Comments

Your attention is directed to the last page of this document which is a Codex-addressed reply form to be used for evaluating the effectiveness of this manual. All comments are beneficial to both you, the product consumer, and to Codex, the supplier.

Writer: Dorie Hitchen
Editor: Lisa A. Maregni

Part No. 08148
Publication Code LP
First Printing May 1985

Contents

Contents (continued)

Contents (continued)

Contents (continued)

Contents (continued)

Contents (continued)

Preface

This guide is intended for network administrators of the Codex 4000 Series LAN. The information presented here introduces you to network operations, and shows you how to install, configure, and manage the network software.

Since this manual includes only a very brief overview of Codex LAN hardware components, you may wish to refer to the 4000 Series LAN Hardware Reference Guide for more detailed information.

The material presented in this guide assumes that you are familiar with the communication characteristics of the devices you are connecting to your network. If you do not have this familiarity, ensure that someone is available to answer questions or contact your local Codex representative. Chapters 1 through 9 introduce you to the Configure program and show you how to configure your network for Codex LAN virtual circuits. A list of the Configure Program menus in Chapters 5 and 6 appears later in this Preface. You can use this list for quick reference when you are running the Configure program. Chapters 10, 11, and 12 are more highly technical in nature and are intended for users who will provide their own programming interface to the services offered by Codex LAN.

Conventions Used In This Guide

The following conventions have been used in instructional text throughout this guide.

[CR] This keystroke notation refers to the Enter key on the IBM PC or PC XT keyboard. It also refers to the Return, Enter, or Execute key found on the keyboards of other types of terminals.

⟨DEL⟩ This notation refers to ASCII nonprinting characters. Chapters 3 and 5 discuss these characters and show you how to find your terminal's keyboard equivalent.

Ctrl-T	This keystroke notation requires you to press the Ctrl key and simultaneously press another key. The example keystroke can be interpreted this way: hold the Ctrl key down and then type T. Your terminal's Ctrl key may have a different spelling.

Most of the command lines included in this guide are shown in uppercase characters for instructional purposes. You can enter command lines, filenames, or command options in upper- or lowercase characters, except where noted in the text.

Table of Configure Program Menus

The following table lists the titles of menus which appear in Chapters 5 and 6. The menu titles are listed in the order in which they are discussed in the text. When you are running the Configure program, you can use this table for quick reference to sections which describe the menu section you are working with. The actual menu title is listed in the left column. The manual topic column lists the type of device, communication service, or I/O module the menu applies to. The pages on which the menus appear are listed on the right.

Menu Title	*Manual Topic*	*Pages*
RS-232 Line Signals	Asynchronous Devices	5-6
Circuit Deactivation	Asynchronous Devices	5-6
Signal Behavior	Asynchronous Devices	5-10, 5-13, 5-14
Extended RS-232	Asynchronous Device, Serial/Parallel I/O	5-12
Parallel Device Dependent Characteristics	Serial/Parallel I/O	5-15
Handshaking	Parallel Device, Serial/Parallel I/O	5-15
Device Dependent Characteristics	Buffers and Thresholds	5-17, 5-18
Command Interpreter Characteristics	Asynchronous Devices	5-19, 5-21
Asynchronous Device	Buffers and Thresholds	5-23
SCS I Device	Buffers and Thresholds	5-24
SCS II, III Devices	Buffers and Thresholds	5-24
Board Function	Board Configuration	6-3
Asynchronous Driver	Board Configuration	6-4
Asynchronous Driver	Port Data Rates, Serial/Parallel I/O	6-4
Serial Asynchronous Port Parallel Port	Port Data Rates, Serial/Parallel I/O	6-5
SCS I Devices	Port Data Rates, Serial/Parallel I/O	6-5

Menu Title	*Manual Topic*	*Pages*
SCS II, SCS III Devices	Port Data Rates, Serial/Parallel I/O	6-5
Serial Asynchronous Port	Port Data Rates, Six Serial I/O	6-6
Network Originated Circuit Data Rate	Automatic Baud Rate Detection, Six Serial I/O	6-7
Add Port Names	Asynchronous Devices	6-8
Delete Port Names	Asynchronous Devices	6-8
Circuit Initiation	Asynchronous Devices	6-9
Bound Circuits	Asynchronous Devices	6-10
Minimum Number of Circuits	Asynchronous Devices	6-11
Multiple Virtual Circuits	Asynchronous Devices	6-11
Command Interpreter	Asynchronous Devices	6-11
Number of Circuits	Asynchronous Devices	6-12
Circuit Deactivation	Asynchronous Devices	6-13
Circuit Initiation, SCS II, SCS III	Synchronous Devices	6-16
Circuit Initiation, SCS I	BSC Devices, Session-Oriented Circuits	6-16
Command Interpreter Characteristics	BSC Devices	6-17
Circuit Deactivation	Synchronous Devices	6-18
Network ID	Network Configuration	6-19
Local Bridge	Bridge Configuration	6-21
Remote Bridge	Bridge Configuration	6-23

Chapter 1
General Description

Contents

OVERVIEW

The Codex 4000 Series Local Area Network (LAN) is a general-purpose local area communications network which interconnects data processing and word processing equipment in office buildings, computing centers, laboratories, factories, and campuses. From an initial installation of a few nodes, a LAN can be expanded to a large-scale communication system consisting of thousands of different devices. Equipment from different manufacturers and devices with dissimilar communication characteristics can communicate with each other over a shared transmission medium: a baseband or broadband cable plant.

The LAN provides a number of networking services—virtual circuits, datagrams, file transfer, and a naming facility for connected devices. It resolves equipment incompatibility through programmable interfaces called Entryways, discussed in the following section.

Figure 1-1 illustrates a typical Codex LAN. Most of the terminals, printers, and host computers pictured are not directly connected to the LAN. They are connected to Codex Entryways which, in turn, connect to the coaxial cable.

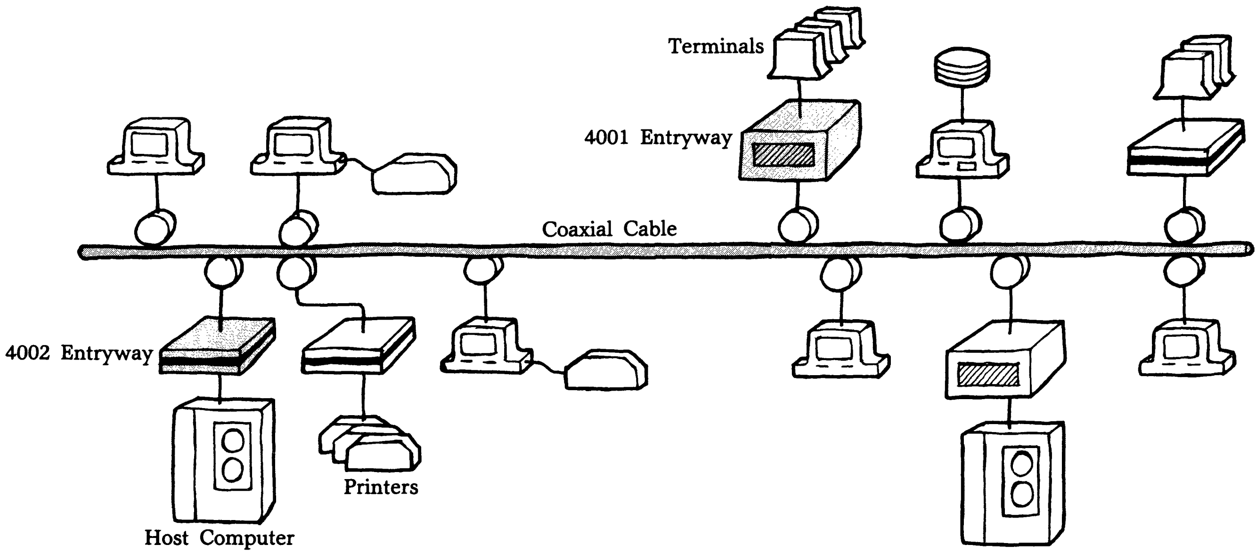


Figure 1-1. User Devices on the Codex LAN

ENTRYWAYS

An Entryway serves as an interface between a Codex LAN and the user devices which it serves. Each Entryway includes from one to four application processor boards. Attached to each board is an I/O module which can contain as many as six ports or connectors to which user devices plug in. Each board has its own microprocessor which provides the intelligence necessary to control communication between the devices connected to its ports and other devices in the network.

An Entryway performs four major tasks:

- Encodes the data sent from its devices into packets which can be transmitted over the network
- Handles the speed conversions necessary to allow incompatible devices to communicate
- Routes data from its devices to the correct destination
- Converts packet data received from the network into the data format of its devices

The 4001 Entryway contains up to four application processor boards, each with an attached I/O module. Codex supplies five different I/O modules, described in the 4000 Series LAN Hardware Reference Guide. Each processor board has a label which identifies it by letter as Board A, B, C, or D. Figure 1-2 shows a rear view of a 4001 Entryway and indicates the location of each of these boards. The 4002 Entryway contains one processor board, always labeled "A".

CODEX PERSONAL COMPUTER NETWORK

Codex manufactures two specially designed personal computer interfaces to the LAN. These are the 4010 and 4011 Entryway adapter cards, designed as single circuit boards which are inserted into an expansion slot in an IBM PC XT* or PC computer. Both of these cards allow a PC user to connect to a

*IBM PC XT, PC, and PC DOS are registered trademarks of IBM Corporation.

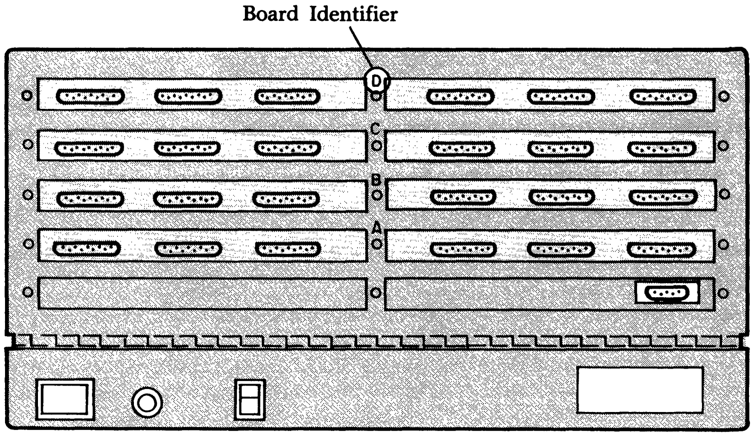


Figure 1-2. Application Processor Boards in a 4001 Entryway

baseband or broadband LAN. When a PC is connected to a broadband LAN, an RF modem card is inserted into a second expansion slot.

The Codex Personal Computer Network (see Figure 1-3) allows PCs to share disk and printer resources with other PCs. Codex Disktalk/Printtalk software is installed on an IBM PC XT with a 10-Mbyte fixed Winchester disk. This PC is called a Network Server. It can share its fixed disk and any attached printers with other PC users in the network. The software manages this function transparently. The PC XT user can run PC DOS programs while the Disktalk/Printtalk software independently manages the sharing of resources.

Users of the Personal Computer Network can not only share hard disk and printer resources with PCs, but they can also connect to other LAN resources.

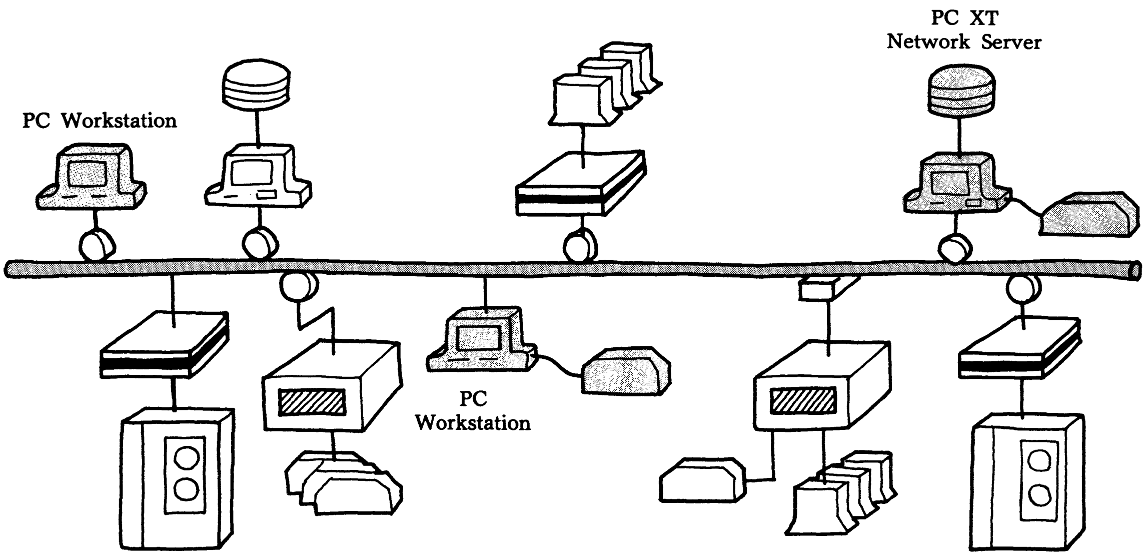


Figure 1-3. Codex Personal Computer Network

BRIDGED NETWORKS

One Codex LAN can be linked to another with specially configured 4000 Series bridges. A Remote Bridge connects two widely separated LANs via a high speed data communications link, such as a phone line. One 4002 Entryway with special Remote Bridge software connects to each network segment. The joined networks together form an Internet.

Two networks that are not geographically separate can be linked into an Internet by a Local Bridge. A Local Bridge is a specially equipped 4001 Entryway (see Figure 1-4).

The two networks can be linked by either a baseband or broadband cable plant. Bridges are further described in the 4000 Series LAN Hardware Reference Guide.

CODEX 4810 PC XT LAN MANAGER

The Codex 4810 PC XT LAN Manager is an IBM PC XT-based Network Management facility. The Codex 4810 has a 10-Mbyte fixed disk and a floppy disk drive. It operates under Release 2.0 (or later) of PC DOS. The Network Operating System Software (NOSS) is installed on its fixed disk. The 4810 can run all the programs any IBM PC XT can. It can be used as a Disk-talk/Printtalk Network Server, and can also run network management programs (described in Chapter 2).

The LAN Manager performs these network management tasks:

- Stores the Network Operating System Software (NOSS) on its fixed disk
- Creates configuration files which define which NOSS software modules should be loaded onto the application processor boards in individual Entryways
- Downloads the software listed in these files to Entryways on the LAN when it is requested to do so

There must be at least one LAN Manager for each network. If you have multiple bridged networks, you must have one LAN Manager for each network segment.

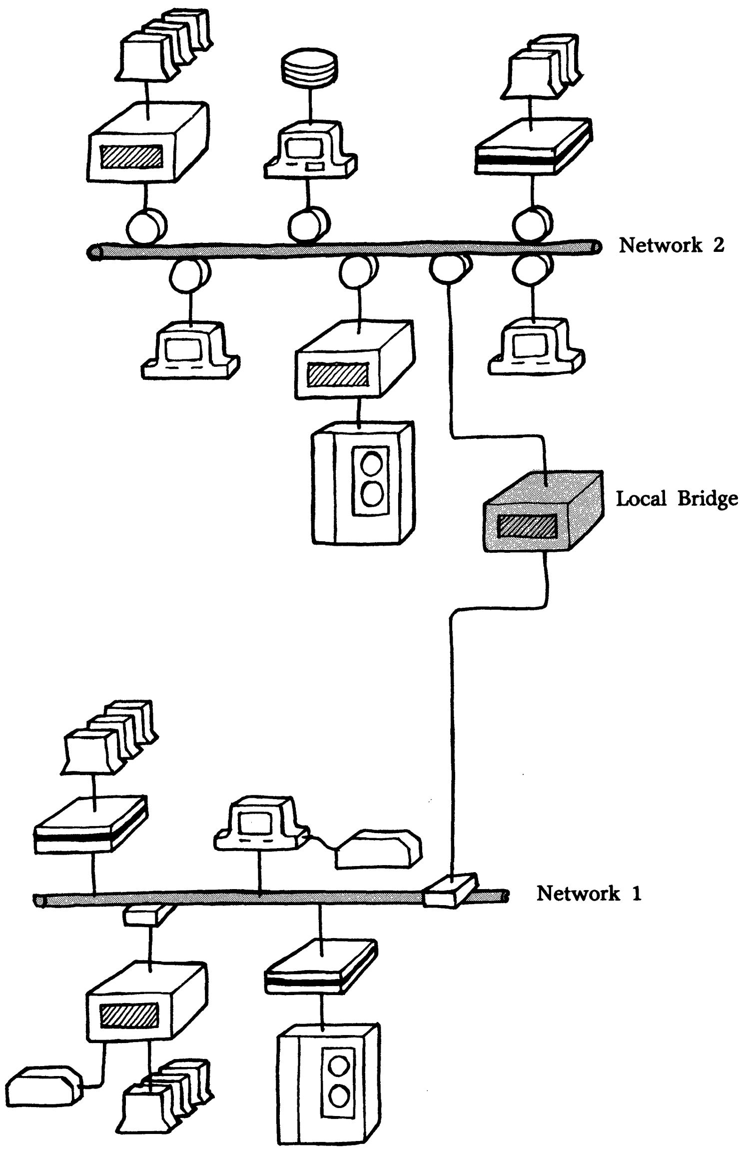


Figure 1-4. Locally Bridged Networks

NETWORK OPERATING SYSTEM SOFTWARE—NOSS

Codex 4000 Series LAN Release 2.0 software includes NOSS and network management programs run from the LAN Manager. NOSS is the distributed Network Operating System Software which runs on application processors within LAN Entryways. This software manages the Entryway's networking activities which provide data transmission and delivery, communications protocol processing, communications drivers, and LAN services.

Codex offers two NOSS software packages for use with the LAN. One is a "starter set" which includes general-purpose networking software, the Virtual Circuit Service software, and the Codex ANSI Terminal Emulator (CATE), the software which allows PCs on the network to initiate virtual circuits. A "full function" software set includes the general-purpose networking software, the Virtual Circuit, Datagram, and Ethernet Data Link services, CATE, Bisynchronous (SCS-I), SDLC/HDLC (SCS-II), and DDCMP (SCS-III) protocol support, and Local or Remote Bridge software. Each application processor board provides one service and one communications protocol for the devices connected to its ports.

Release 2.0 also includes two network management programs run under PC DOS from the LAN Manager: Configure and the Download Server.

Configure Program

Configure is a menu-driven program which asks the Network Administrator to describe the communication characteristics of the devices connected to Entryways, the I/O modules attached to application processor boards, the services running on each board, and the connection names of resources in the network. The answers to these questions are listed in Entryway configuration files (see Chapter 4) stored on the LAN Manager fixed disk. These files determine which NOSS software modules will be downloaded into Entryway application processors.

Download Server

When an Entryway on the LAN is reset or powered on, it performs internal diagnostics, and then requests a software load. The Download Server listens for these requests and responds by transmitting NOSS software modules over the LAN to the requesting Entryway. Each board within an Entryway receives a separate software load. A network processor on Board A supervises the transfer of this software from the LAN to Board A and then, via an internal databus, to each application processor board in turn.

The Download Server does not download software to PCs on the network (see Figure 1-5). You install the Codex ANSI Terminal Emulator (CATE) software onto the PC fixed disk or onto a bootable floppy diskette. When the PC is powered on, routines are invoked which load networking software from the disk or diskette onto the Entryway card. Since the networking software is already resident on the PC, there is no need to create PC configuration files.

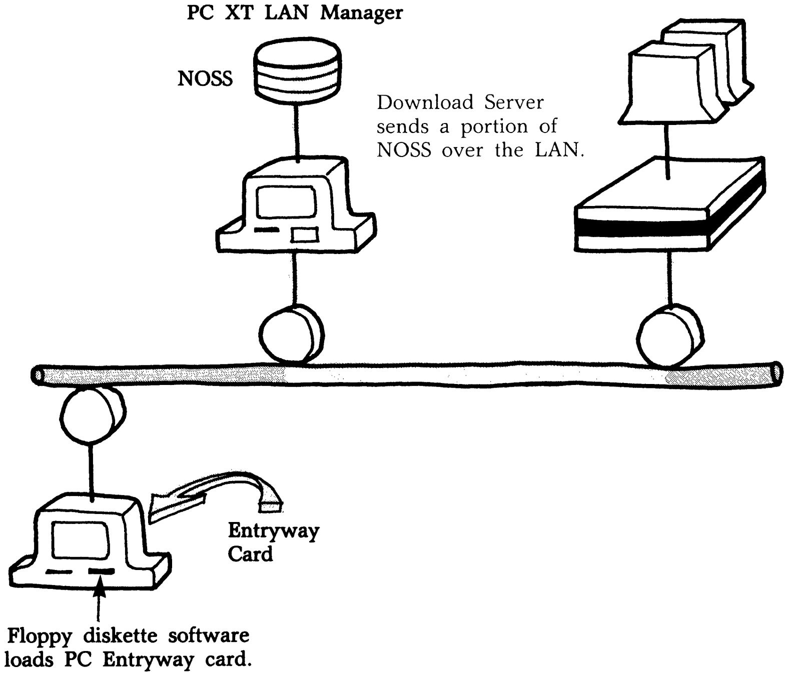


Figure 1-5. Software Downloading

Chapter 2
Personal Computer Guide to Operations

Contents

CODEX ANSI TERMINAL EMULATOR (CATE)

The Codex ANSI Terminal Emulator (CATE) software allows an IBM PC or PC XT to communicate with many resources on the LAN as if the PC were a standard ANSI terminal connected to an Entryway. Using the Terminal Emulator program, a PC user can establish virtual circuits to network resources such as asynchronous terminals or host computers. PCs, however, can only connect to other PCs through the Disktalk/Printtalk network. Table 2-1 lists the commands available to PC users through the Terminal Emulator program. The commands listed can be typed in upper or lower case unless otherwise specified. You will learn how to use each of these commands in Chapter 3.

Before you can connect a PC to the network, a Codex 4010 or 4011 Entryway card must be installed in an expansion slot in the PC or PC XT. Refer to the 4000 Series LAN Hardware Reference Guide for information on these Entryways.

To install the CATE software, which is on a master floppy diskette, the Network Administrator can copy the software and create boot diskettes for PCs or set up the software on the fixed disk of a PC XT. ■ **Note:** The CATE software for the 4810 PC XT LAN Manager is included in the NOSS network management software. Installation of NOSS (described later in this chapter) implements the Terminal Emulator.

If you are installing CATE onto a Network Server with the Disktalk/Printtalk software, refer to the *4000 Series LAN Disktalk/Printtalk Administrator's Guide* for installation instructions.

Table 2-1.
CATE Commands

Command	*Action*
CATE [CR]	Invokes the terminal emulator.
Ctrl-Wquit [CR]	Quits the terminal emulator. The command word "quit" must be typed in lowercase characters.
CONNECT [CR]	Connects to another network resource.
Ctrl-T	Disconnects a virtual circuit.
Ctrl-P	Data Link Escape character.

Installing CATE on an IBM PC

Use the following procedure to create a CATE boot diskette for the PCs in your network.

1) Insert the IBM PC DOS diskette into Drive A of the PC.

2) Power on the PC. After diagnostics have been completed, enter the date and time if prompted.

3) Insert a blank double-sided/double-density diskette into Drive B. The diskette should be write-enabled.

4) Format the diskette by entering:

    FORMAT /S B: [CR]

    The /S option transfers necessary PC system information onto the blank diskette.

5) After the format is completed, copy the ANSI.SYS file from the PC DOS diskette onto the newly formatted diskette by entering:

    COPY A:ANSI.SYS B: [CR]

6) When the ANSI.SYS file has been copied, remove the PC DOS diskette from Drive A.

7) Insert the diskette labeled CATE into Drive A. Enter:

    COPY A:*.* B: [CR]

8) When all of the files have been copied, ensure that the boot diskette contains both a CONFIG.SYS and an AUTOEXEC.BAT file. If it does not, you will have to create these files. Refer to the *IBM Disk Operating System (DOS) Reference Guide* for instructions on how to do so.

9) Type the following entries into the CONFIG.SYS file if they do not already exist:

```
DEVICE=ANSI.SYS
DEVICE=WSNETI.DRV
DEVICE=NETCI.DRV
```

10) Type the following entries into the AUTOEXEC.BAT file if they do not already exist:

```
LOAD4010 -R -D KER186 4010
MEPPAS
```

11) Remove the diskette from Drive B and label it CATE Boot Diskette.

12) Insert the boot diskette into Drive A. Reboot the PC by holding the Ctrl and Alt keys and at the same time pressing the Del key.

You are now ready to invoke the Terminal Emulator.

Installing CATE on an IBM PC XT

Use the following procedure to install the Terminal Emulator software onto the fixed disk of an IBM PC XT. ■ **Note:** Installing NOSS on a PC XT also installs CATE. Refer to the section titled Installing NOSS on a PC XT later in this chapter.

1) Power on the IBM PC XT. After diagnostics have been completed, enter the date and time if prompted.

2) If it has not already been done, partition and format the fixed disk, and copy the files from the PC DOS diskette onto it. Refer to the *IBM Disk Operating System (DOS) Reference Guide* for instructions on how to do so.

3) Insert the IBM PC DOS diskette into Drive A (the diskette drive). Enter:

```
COPY A:ANSI.SYS [CR]
```

This step is necessary because normal PC DOS installation does not copy the ANSI.SYS file onto the fixed disk.

4) When the copy is complete, remove the PC DOS diskette.

5) Insert the diskette labelled CATE into Drive A (the diskette drive). Enter:

   COPY A:*.* [CR]

6) When all of the files from the CATE master diskette have been copied, remove it and store it in a safe place.

7) Ensure that the fixed disk contains a CONFIG.SYS and AUTOEXEC.BAT file. If it does not, you will have to create these files. Refer to the *IBM Disk Operating System (DOS) Reference Guide* for instructions on how to do so.

8) Enter the following into the CONFIG.SYS file:

```
DEVICE=ANSI.SYS
DEVICE=WSNETI.DRV
DEVICE=NETCI.DRV
```

9) Enter the following into the AUTOEXEC.BAT file:

```
LOAD4010 -R -D KER186 4010
MEPPAS
```

10) Reboot the PC XT by holding the Ctrl and Alt keys and at the same time pressing the Del key.

You are now ready to invoke the Terminal Emulator.

INSTALLING NOSS ON THE LAN MANAGER

Before you begin, you should have completed the following steps:

1) Installed a Codex LAN 4010 Entryway into the Codex 4810 PC XT LAN Manager (also referred to as the LAN Manager). If you have not done so, refer to the *4000 Series LAN Hardware Reference Guide* for installation instructions.

2) Installed PC DOS onto your fixed disk. If you have not done so, you should partition and format your fixed disk and then copy PC DOS onto it. Refer to the section titled Preparing Your Fixed Disk in the *IBM Disk Operating System (DOS) Reference Guide* shipped with your PC XT.

3) If you are installing NOSS on a LAN Manager that will also be a Disktalk/Printtalk Network Server, the Disktalk/Printtalk software must be installed first, then PC DOS, and finally NOSS. ■ **Note:** Refer to the *4000 Series LAN Disktalk/Printtalk Administrator's Guide* for installation instructions. Do *not* follow the instructions included here.

To install the NOSS software:

1) Power on the IBM PC XT. After diagnostics have been completed, enter the date and time if prompted.

2) Change your default drive to C, if you are not already reading from the fixed disk drive.

3) Insert the diskette labeled NOSS DISK 1 into Drive A (the diskette drive).

4) Enter the following:

   A:NINSTALL [CR]

   A system prompt asks you to confirm that you wish to create a directory named "\codex" for the NOSS software and the network configuration files.

5) Enter Y to create the directory "\ codex". Press any key when you are ready to proceed.

6) When all of the files from NOSS DISK 1 have been copied, remove and archive it.

7) Insert NOSS DISK 2 into Drive A. Repeat Step 4, above.

8) After all of its files have been copied, remove and archive the master diskette.

If you purchased the Codex software "starter set," all of the NOSS files are now copied onto the fixed disk. Proceed to Step 10 below. If you purchased the "full function set," you received a third diskette labeled NOSS EXTEND. Perform Steps 9 and 10 below.

9) Insert NOSS EXTEND into Drive A. Repeat Step 4, above. When all of the files have been copied from the master diskette, remove and archive it.

10) Reboot the PC XT using the Ctrl, Alt, and Del keys. You are now ready to invoke Configure, the Download Server, or the Terminal Emulator (CATE).

Successful software installation creates the CONFIG.SYS and AUTOEXEC.BAT files for you. Entries are then made in the AUTOEXEC.BAT file which set the default (initial) directory to "\ codex" and sets the prompt to the drive and directory name. The DOS prompt changes and looks like this:

C:\ codex

LAN MANAGER MODES OF OPERATION

The LAN Manager can operate in one of two modes: the Download Server mode or Console mode. When the LAN Manager is in Download Server mode, it listens for and responds to download requests over the LAN. It cannot use PC DOS commands, connect to other workstations except through the Download function, or run the Configure program.

When the LAN Manager is in Console mode, it cannot listen for or respond to download requests. It can, however, perform the following functions:

- Operate as a standalone workstation, using PC DOS commands
- Operate as a network configuration facility using the Configure command.
- Emulate an ANSI terminal on the LAN using the CATE command.

Refer to the section on the Codex ANSI Terminal Emulator at the beginning of this chapter for a description of the commands available to the LAN Manager when it invokes the Terminal Emulator. The use of these commands is discussed in Chapter 3.

Using Network Management Commands

The two network management commands, Configure and the Download Server, are invoked from Console mode. The Configure program is used after NOSS is installed on the PC XT fixed disk. Before the network can operate, configuration files must be created and resident in the Codex directory.

To invoke Configure, enter at the Codex DOS prompt:

CONFIGURE [CR]

The following prompt appears:

CONFIGURE
Type ? for HELP

Chapters 4, 5, and 6 describe the questions Configure asks you when it builds these configuration files. Chapter 8 shows you how to move through the Configure menus.

Each time you use the Configure program to create or modify Entryway configuration files, you must reset the Entryways whose configurations you created or changed. The Download Server is then invoked to transmit the appropriate software modules to each application processor.

To invoke the Download Server, enter at the Codex DOS prompt:

DLS [CR]

The following message appears:

Codex Download Server
DLS—Awaiting Download Request

After the Entryways have completed diagnostics, the software modules defined by your configuration are downloaded over the LAN on a per-board basis. As each application processor is booted, a message displays showing you the address of the processor and the names of the modules sent. Pressing Ctrl-C at any time aborts the Download Server and returns you to the Codex DOS prompt.

Local and Remote Access

The LAN Manager can be placed in Local or Remote Console mode. When placed in Remote Console mode, the LAN Manager can be accessed from any asynchronous terminal on the LAN. That terminal, once the connection has been established, can invoke Configure and the Download Server. ■ **Note:** Some PC DOS commands can also be invoked from a remote terminal. Any programs, however, which access the PC screen and keyboard directly will not display correctly on the remote terminal, e.g., BASIC.

To place the LAN Manager in Remote Console mode, enter next to the Codex DOS prompt:

REMOTE [CR]

This message displays:

Codex LAN Remote Console Mode
Listen mode has been entered

The LAN Manager can now accept a connection from any remote terminal. Since the Codex DOS prompt leaves the screen, the LAN Manager cannot accept commands or keystrokes from its own keyboard.

To connect a remote terminal to the LAN Manager, type:

C MGR [CR]

MGR is used here as a connection address for the LAN Manager. When the connection has been made, the Codex DOS prompt displays on the remote terminal.

The remote terminal can now enter network management commands. The 4810 terminal cannot enter commands until the remote terminal returns it to Local Console mode. To do so, the remote terminal would enter at the DOS prompt:

LOCAL [CR]

The remote terminal would then enter a disconnect sequence to break the connection. The LAN command prompt would display on the remote terminal and the Codex DOS prompt would display on the 4810.

Table 2-2.
LAN Manager Commands

Command	*Action*
REMOTE [CR]	Places the LAN Manager in Remote Console mode.
LOCAL [CR]	When entered from a remote terminal, returns the LAN Manager to Local Console mode.
CONFIGURE [CR]	Invokes the Configure program.
DLS [CR]	Invokes the Download server.
Ctrl-C	Entered while running Configure or the Download Server, aborts the program. Returns the LAN Manager to the mode it was in when these programs were invoked.
CATE [CR]	Invokes the Terminal Emulator program.
Ctrl-Wquit [CR]	Quits the Terminal Emulator program. The command word "quit" must be typed in lowercase characters.

Table 2-2 summarizes the commands described in this section. Unless otherwise specified, all commands listed can be typed in upper or lowercase characters.

Using Setup Commands

You can program an initial operating mode for the LAN Manager by entering a Setup command at the Codex DOS prompt. Each command specifies a mode of operation. When the LAN Manager is powered on or reset, its default (initial) mode will be the mode you select. Setup commands are described in Table 2-3. All commands listed are entered at the Codex DOS prompt and may be typed in upper or lowercase characters.

Table 2-3.
LAN Manager Setup Commands

Command	*Action*
NSETUP -D [CR]	The Download Server mode is invoked on power-on or reset.
NSETUP -C [CR]	Console mode is invoked.
NSETUP -L [CR]	Local Console mode is invoked.
NSETUP -R [CR]	Remote Console mode is invoked.

USING PC NAMES

After NOSS installation, you can use the PC Names program to establish console names for the LAN Manager. Console names identify the LAN Manager to network users. The console name is entered as a connection address when a user at a remote terminal initiates a connection (see Chapter 3). You do not have to establish console names for other PCs or PC XTs on the network, since PCs cannot accept connections.

After CATE and NOSS installation, you should establish Terminal Emulator names for the LAN Manager and for other PCs and PC XTs on the network. Terminal Emulator names are used by the network to identify the PC. The Examine command, described in Chapter 3, allows a user to see what resources a specific device is connected to. If the device is connected to a PC or PC XT, the Terminal Emulator name you supply here will be the connection name listed. If you do not supply a Terminal Emulator name, the connection name will be left blank.

The default console name for the LAN Manager is MGR. You can change it, add additional names, or delete existing names. There is no system default Terminal Emulator name. To use the PC Names program, at the Codex DOS prompt, type:

PCNAMES [CR]

Prompts, similar to those below, appear:

The following Console names are currently defined:
a)name
Delete any Console names (Y or *N)?

Add any Console names (Y or *N)?

No Terminal Emulator names are yet defined.
Add any Terminal Emulator names (Y or *N)?

To add or delete a name, type Y, then press [CR].

The program will prompt you to supply a name or select a name you wish to delete. When you reboot the PC or PC XT, the name changes take effect and the following message displays during initialization:

cn Name
Network Console
*****name registered*****
x Name(s) Registered

''Name'' is the name you have assigned, and x is the number of names assigned. The first three lines of the screen message are repeated for each console name registered.

Chapter 3
Administrator's Guide to Operations

Contents

Contents (continued)

VIRTUAL CIRCUITS

Devices on the network communicate with each other through circuits (links) created by software known as the Codex LAN Virtual Circuit Service. These software-created circuits, called virtual circuits, carry data between any two devices in the network, without regard to their physical location. Virtual circuits can also carry data between different networks.

Virtual circuits are implemented by software running on the application processor boards within an Entryway. The devices at each circuit endpoint communicate with an Entryway and not directly with each other. The Entryway then transmits the data over a shared broadcast communications medium: the LAN cable. Although the devices are the ultimate endpoints of the circuit, the endpoints are really software processes within the respective Entryways. Because these software processes control the devices at each end of the circuit, the effect, as far as the user is concerned, is just as though the devices were directly connected by a point-to-point link. This scheme parallels the telephone system: while the communicating endpoints in the phone system appear to be the communicating parties, the circuits are in fact established between two telephones (see Figure 3-1).

Virtual circuits have many advantages over physical circuits. The virtual circuit software can resolve speed mismatch problems between devices and can provide for data retransmission in case of communication errors. The software generates, transmits, and receives packets on a per-circuit basis, keeping the data of the individual circuits separate. Codex LAN ensures that the data on each circuit arrives in the proper order, correctly, and without duplication.

Using software-controlled processes as the endpoints for virtual circuits provides great flexibility in the types of service that can be provided for any particular connection. For example, circuit endpoints might be individually tailored to accept or reject connection attempts so that various types of access control can be placed on a particular device. You implement these special features when you configure your network using the Configure program.

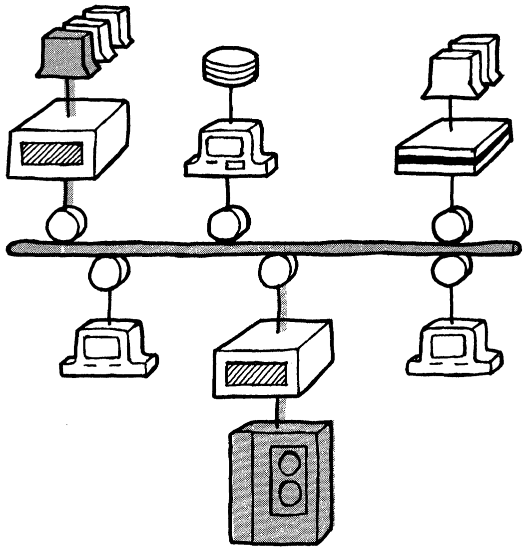


Figure 3-1. Virtual Circuit Endpoints

Finally, since the connection between devices is actually a software-controlled connection, devices can establish circuits to network services running on specially-configured application processor boards.

CIRCUIT INITIATION METHODS

Virtual circuits can be established either between two devices connected to the same processor board in an Entryway or between two devices connected to different processor boards in the same or different Entryways.

Codex LAN provides three types of virtual circuits: session-oriented, bound, and administrative. These circuits are described by the way in which they are initiated; the circuits themselves are not actually different from one another.

Session-Oriented Circuits

A device establishes a session-oriented circuit with another device by issuing the LAN Connect command. Session-oriented circuits are so named because they exist only for the length of the operating session. When a device breaks the connection, the circuit is terminated and can only be recreated by another user Connect command.

Bound Circuits

Bound circuits are created by the Network Administrator at software configuration time, rather than by a direct user command. Entryway configuration files define which circuits should be created every time an Entryway is powered on or reset. Unlike session-oriented circuits, bound circuits are static. The circuits are established every time the Entryway is initialized unless you change the Entryway configuration files which create them.

There are three types of bound circuits: permanent circuits, demanded circuits, and initial circuits.

Permanent Circuits. A permanent circuit is the simplest form of bound circuit. It will look like a hard-wired connection to the device initiating the circuit. The initiating device will not be able to connect to other LAN resources. Even if the circuit endpoint device breaks the circuit and connects to another device, the initiating device's Entryway will persist in the attempt to reestablish the connection. Examples of devices you might configure for this type of circuit would be a computer with a dedicated printer, or a terminal requiring the processing capability of one device, such as a host computer.

Demanded Circuits. A demanded circuit is created whenever a device issues a demanded circuit initiation character, which you will learn how to configure in Chapter 6. Whenever a user device issues this character, the Entryway attempts to create the circuit. If the resource is busy (connected to another resource), the Entryway will persist in the attempt until the connection is established. It will, however, eventually time out. If one of the devices breaks the connection, the Entryway will not attempt to reestablish it except on demand. Each party is then free to connect to other devices on the LAN. You might, for example, configure a demanded circuit for a computer connected to a printer. The computer could issue a print command after the demanded circuit initiation character was entered to establish the connection.

Initial Circuits. An initial circuit is created whenever an Entryway is powered on and remains in effect until one of the connected devices breaks the connection. If the connection resource is busy, the Entryway will persist until the circuit is established. If one of the devices breaks the connection, the Entryway does not attempt to recreate it. An example of a device you might configure for an initial circuit is a terminal which downloads some initialization sequence from a host computer. After initialization is completed, the circuit can be broken, and there is no need to reestablish it. Both parties are then free to connect to other devices on the LAN.

Administrative Circuits

Administrative circuits are session-oriented circuits created by a third party and interconnecting two other devices. Either device can break the circuit, or a third party Disconnect command can break it.

A network user can create only session-oriented circuits. As a Network Administrator, however, you can create all of these circuits.

DEVICE FLOW CONTROL

Codex LAN virtual circuit software provides complete device-to-device flow control. Since devices communicate with each other by first communicating with their Entryways, some form of flow control between the user device and the Entryway is required. This ensures that the transmitting device does not overrun the Entryway with data in cases where device bursts can exceed the buffering capacity of the virtual circuit.

The virtual circuit software allows the use of several industry-standard flow control techniques. One example is the XON/XOFF character strategy, in which one party asks the other to suspend its data transmissions until the receiving party is ready to accept more data. Many devices currently employ this kind of simple flow control mechanism to prevent themselves from being overrun by other data-generating devices.

Other types of flow control involve the manipulation of various RS-232 line signals (such as RTS:CTS or DSR:DTR).

DEVICE TYPES

Depending on the function of the devices you connect to your LAN, you may configure them to manage virtual circuits in different ways. The Configure program will ask you if a device can initiate a virtual circuit using the LAN Connect command. It will also ask if a device can serve as the target or endpoint of a connection attempt from another device. To simplify your configuration task, you can think of your devices as falling into three general types: command devices, data devices, or command/data devices.

- ***Command devices*** use the LAN Connect command (discussed later in this chapter) to initiate circuits or connections to other devices, but cannot serve as targets for circuits created elsewhere on the network. An example of a command device is a computer terminal that is solely dedicated to entering data but that cannot accept a connection from another device for display purposes.
- ***Data devices*** cannot generate LAN commands. They cannot, therefore, initiate virtual circuits. They can serve only as targets for connection attempts from other devices. A line printer without a keyboard is a data device. It only receives data; you cannot use it to issue a command to create a circuit. Once the circuit has been established, a data device can exchange data on the circuit just like any other device.
- ***Command/data devices*** can generate commands to create virtual circuits and can serve as targets for other devices' connection attempts. A display terminal with attached keyboard could be designated as a command/data device because you can use it to enter commands (to create circuits), and it can also accept a connection from (be used as a data device by) another user on the network.

DEVICE MODES OF OPERATION

Codex LAN implements three modes of operation for the devices you connect to the network: Command mode, Connected mode, or Idle mode. The following paragraphs explain each mode in greater detail.

Command Mode

When a device is in Command mode it can issue Codex LAN commands, including the Connect command which initiates a virtual circuit. When a command or command/data device first enters Command mode, it displays the following message:

```
You may now enter Codex LAN commands
>
```

The right angle bracket prompt character (>) in the left margin indicates that the Codex LAN command interpreter is waiting for a command. The commands you and your network users can enter at this prompt are discussed later in this chapter. If no command has been entered for a predetermined period of time, the LAN command interpreter times out, and a command/data or data device enters Idle mode (described below). The Codex LAN default timeout period is 288 seconds; however, you may change this figure using the Configure program.

Connected Mode

Two devices enter Connected mode when a virtual circuit has been established from one device to another. Once the connection has been established, both devices are said to be in Connected mode and can therefore exchange data.

When one of the devices breaks the connection by using a special string of characters known as a Disconnect sequence (explained later in this chapter), each device returns to the mode from which it began. If a terminal initiated a connection to a printer, it would first be in Command mode, then in Connected mode, and after the disconnect, would go back into Command mode.

Idle Mode

When a device is in Idle Mode, any other device can connect to it. Command devices are never in Idle Mode and thus can never serve as targets for connection requests. When an Entryway is powered on or reset, command/data and data devices are initially in Idle mode.

Not all devices can operate in all three modes. Only a command/data device can switch between all three modes of operation. A data device can only be in either Idle mode or Connected mode. It can never be in Command mode because it cannot be used to issue LAN commands. A command device can only be in either Command mode or Connected mode. Because it cannot serve as a target for another device's connection attempt, it can never be in Idle mode.

DEVICE AND SERVICE ADDRESSES

Devices and services connected to LAN Entryways have a unique physical address. This physical, or resource, address describes an exact physical location within the network. Physical or resource addresses are used by the Configure program to identify the configuration files it creates for LAN Entryways. These addresses are also used to name the target of a connection request. ■ **Note:** IBM PCs and PC XTs have hexadecimal network node IDs, which are used for addressing by the Disktalk/Printtalk software. The PC XT LAN Manager is given a console name which can be used as a connection address. Since other PCs cannot accept connections, there is no need to give them a console name. Both PCs and PC XTs should be given Terminal Emulator names. Refer to the section titled Using PC Names in Chapter 2 of this guide.

Device Addresses

Device addresses identify the Entryway to which the device is connected and the port (connector) on the Entryway it uses. This address consists of three items: the Entryway number (its serial number), the processor board ID, and the number of the port on the processor board. Sample addresses are shown on the next page and are explained in the paragraphs that follow. Addresses are always entered as shown. Do not leave spaces between the fields of an address.

```
3729D3                          432B1
↑  ↑↑─Port #3                   ↑  ↑↑─Port #1
│  └──Processor Board D         │  └──Processor Board B
└─────Entryway #3729            └─────Entryway #432
```

- ***Entryway Number.*** The serial number of the Entryway to which the device is connected. The serial number is printed on a label on the Entryway's rear panel.
- ***Processor ID.*** A letter, A through D, which corresponds to the processor board to which the device is connected. The board ID is printed on the Entryway's rear panel.
- ***Port Number.*** The specific port on a board to which the device is connected. The number and type of ports on a board depend upon the type of I/O module attached to it. Port identifiers are printed as decimal numbers on the Entryway's rear panel.

Service Addresses

Services are internal software resources, such as the File Transfer Service, that are available to network users. A service resides on a processor board in an Entryway.

A service address is similar to a device address, with a few modifications. A service's address consists of the Entryway number, the processor ID, plus the name of the service and an optional service modifier. A sample service address is shown below and explained in the paragraphs that follow.

```
23A xfr 3
↑ ↑  ↑  ↑──Service Modifier (optional)
│ │  └─────Service Identifier
│ └────────Processor Board A
└──────────Entryway #23
```

- *Entryway Number.* The address of the Entryway on which the service runs. The address, which is also the Entryway's serial number, is printed on a label on the Entryway's rear panel.

- *Processor ID.* A letter, A through D, which corresponds to the processor board on which the service runs. The board ID is printed on the Entryway's rear panel. ■ **Note:** You must type a space between the processor ID and the service identifier.

- *Service Identifier.* The name you assign to a particular service. It is usually the name of the service loaded onto the particular processor board. Service identifiers are limited to six characters, the first of which must be a number or a letter. ■ **Note:** When you enter a service address as part of a network command, you must type a space between the service identifier and the service modifier.

- *Service Modifier.* If your network contains several versions of a service, this modifier identifies the specific version of the software. A service modifier is optional and does not have to be included in a service's address.

CONNECTION RESOURCE NAMES

While the resource addresses just described identify any device or service in a directly unambiguous fashion, most users will find it easier to use a descriptive or mnemonic name to identify resources they wish to connect to. Codex LAN provides a naming facility you can use to create connection resource names for the devices and services in your network. An example of a connection resource name is TOM'S CRT or LINE PRINTER 1. You assign these names when you run the Configure program. After you have configured these names, you can supply a list of them to network users.

Defining mnemonic connection resource names has certain advantages. When you use a device's name as part of a network command, you do not have to know exactly where the device is or to what port on an Entryway it is connected. Thus, if you want to connect your terminal to TOM'S CRT, you do not have to know that yesterday its address was 254B3 and that today, because it has been moved, its address is now 4456C2.

The naming facility is implemented via a distributed approach, whereby each Entryway processor board maintains a local table of name strings associated with its devices and services. During network operation, when the software needs to match a name to a network resource address, the Codex LAN software automatically sends broadcast packets to all processors. Processors ''look up'' the name in their name tables and respond if the name is listed there.

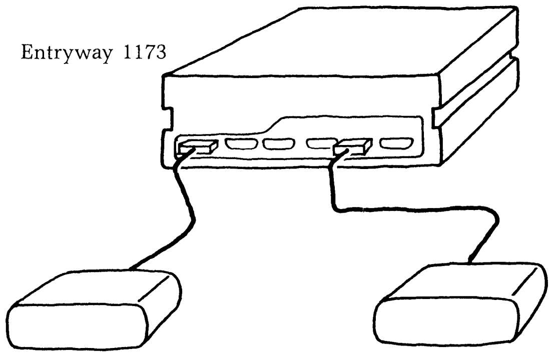
Names are stored exactly as you enter them in the Configure program, and uppercase characters are distinguished from lowercase. However, the name lookup function in the operating network matches upper and lowercase alphabetic characters. Thus, TTY1 matches tty1.

You can configure resource names to be unique or rotored. A unique name is assigned to a single device or service. A rotored name may be assigned to several devices or services.

When a connection attempt is made to a resource with a unique name, the circuit is initiated to the one resource with that name. If the resource is busy, i.e., connected to another resource, the circuit must wait until the resource is free. When a connection attempt is made to a rotored name, the LAN software seeks out the first available (nonconnected) port that shares the name (see Figure 3-2).

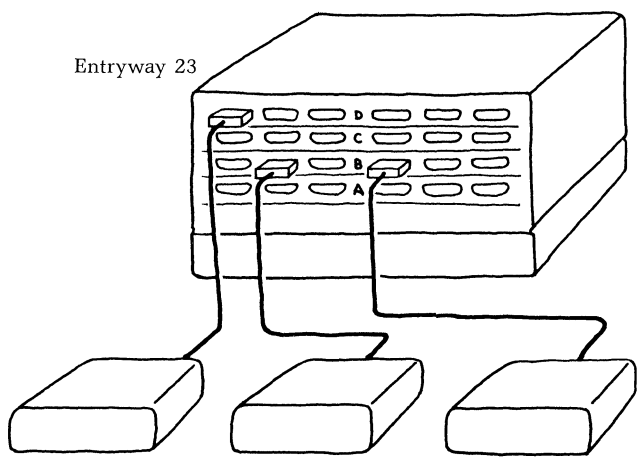
The devices connected to the 4001 and 4002 Entryways in Figure 3-2 each have a physical address, a unique name, and a rotored name. If a connection is initiated to the rotored name Personnel, Device 1 and Device 2 on Entryway 23 or Device A on Entryway 1173 can respond. If a connection is made to the unique name Device A or to the address 1173A1, only the device on Port 1 of Board A in the model 4002 Entryway can respond.

When you run Configure, you have a reasonable degree of freedom in choosing connection resource names. A device or service name is generally a string of up to 30 printable characters. The first character of any name must be a letter or number. Names are delimited by any nonprinting ASCII character such as a Carriage Return or Line Feed, with two exceptions: the Backspace and Line Delete characters do not function as name delimiters. Because nonprinting ASCII



Address:	1173A1	1173A5
Unique Name:	Device A	Device B
Rotored Name:	Personnel	Payroll

(A) 4002



Address:	23D1	23B2	23B4
Unique Name:	Device 1	Device 2	Device 3
Rotored Name:	Personnel	Personnel	Payroll

(B) 4001

Figure 3-2. Connection Resource Names

characters delimit names, you cannot use control characters or non-ASCII characters as part of a name. Other restricted characters are : * (asterisk), / (slash), $ (dollar sign), and @ (at sign).

Punctuation characters, including blanks, are allowed except in the first character position, which permits you to use multiple-word names. However the name typed by the user must match your assigned name character for character (except that upper and lowercase characters may be interchanged). If a resource name appears on a list as VAX 1, a user may enter vax 1. Do not, however, enter VAX1 or vax1 because the space is a character in the resource's name.

USING THE NETWORK

This section explains how you and other network users can connect to the network, using network commands to make and break virtual circuits. All network user commands are discussed. You may configure certain terminals or workstations to use a special or extended command set. These commands are discussed in the Extended Command Set section at the end of this chapter. IBM PC workstations use a subset of the standard command set, listed in Table 2-1 in Chapter 2. References to this command set are also included. How you and other network users connect to the LAN is influenced by some of the choices you make when you run the Configure program. Remember that a terminal must be in Command mode to issue network commands.

Connecting a Terminal to the Command Interpreter

The right angle bracket prompt (>) indicates that a terminal is in command mode and that the LAN command interpreter is waiting for you to issue a command. Codex LAN software includes a timeout feature: if nothing is typed for a predetermined period of time in response to the command prompt, the command interpreter will time out, i.e., the command prompt will leave the screen. The timeout period is 288 seconds or the number you supplied in the Configure program. To display the command prompt again, enter a Carriage Return or a Command mode initiation character (see Chapter 5).

If a terminal has been configured as a command/data device (one that can act as the target for a connection request as well as initiate one), the timeout feature returns it to Idle mode. To enter Command mode, it must enter the Command mode initiation character.

Connecting a PC to the Command Interpreter

To use an IBM PC or PC XT as a network terminal, you must install special software that enables it to communicate on the LAN. This software, the Codex ANSI Terminal Emulator (CATE), allows the PC to emulate terminals connected to Entryways. This software is also included in the NOSS software package you install on the PC XT LAN Manager.

To connect to the network, the PC is booted from a floppy diskette or, in the case of a PC XT, from software installed on its fixed disk (software installation instructions are included in Chapter 2.)

After the PC or PC XT is booted correctly, it displays a DOS prompt similar to:

```
A>
```

or, for the PC XT LAN Manager:

```
C:\ codex
```

The terminal emulator is invoked at this prompt by entering CATE (or cate) and pressing the Return or Enter key.

This screen displays:

```
CODEX LAN TERMINAL EMULATOR

Exit Sequence
/Ctrl-w/quit/CR/
Press any key to continue.
```

To enter Command mode, you would press any key. The standard Codex LAN prompt displays:

```
You may enter Codex LAN commands
>
```

You are now able to use the commands explained in the next section.

ENTERING COMMANDS

Codex LAN commands are used to create and break virtual circuits and to list and change some of the characteristics of your terminal.

You issue these commands by typing a single line of text from your terminal when it is in Command mode. Your terminal is in Command mode when you see a right angle bracket character (〉) in the left margin of the screen.

You may use upper or lowercase letters to enter the command. End the command line with your terminal's Return or Enter key (shown here as [CR]).

You can enter commands in a number of ways. Following are three samples of command lines you could use to establish a virtual circuit between your terminal and the host computer VAX 1 (characters in black are those returned on your screen):

Connect vax 1 [CR]

C〈TAB〉onnect VAX 1[CR]

c vax 1[CR]

In the first example, you would type in the full name of the command and the name of the device, ending the line with a Return or Enter. The other examples illustrate two features of the Codex LAN software: Word Completion and Abbreviation.

Using Word Completion

To use word completion, type any portion of a command word, then type the completion character (shown here as 〈TAB〉). Codex LAN will automatically complete the spelling of the word. If the portion you typed is ambiguous, that is, when two or more commands begin with the same characters, word completion is not performed. Codex LAN returns an error message, aborts the command, reissues the right angle bracket prompt and waits for a new command. The command is also aborted if the network software does not recognize the abbreviation you used.

You select the Word Completion character when you run the Configure program. When you invoke the List command, the character you selected for this feature is displayed. The List command is explained later in this chapter. ■ **Note:** The Word Completion feature is not available on the IBM PC or PC XT.

Using Abbreviations

You can abbreviate LAN command words to a few characters or even one character. Since all LAN commands begin with a different letter, entering the first character is enough to distinguish it from other commands. You can abbreviate the Connect command, for example, by typing "c" instead of "Connect". When you abbreviate, Codex LAN does not display the complete command word as it does in Command Completion.

Using the Backspace Character

If you type part of a command word incorrectly, use the Backspace character to back up and correct the error. The Backspace character works only if you are positioned within the word you are correcting. You cannot back up from another word to correct a typing error in a previous word. You would correct the error using the Line Delete character.

The default Backspace character is either your terminal's Backspace key, or if your terminal does not have a Backspace key, Ctrl-H (hold down the Ctrl key and press the H key). When you use the Configure program, you can define some other character as the Backspace character (see Chapter 5).

Using the Line Delete Character

To delete an entire command or command line, use the Line Delete character. The command is deleted, and Codex LAN reissues its prompt and waits for a new command. The default Line Delete character is ⟨DEL⟩, the Delete key on most terminals. You can change this character when you run the Configure program.

Data Link Escape Character

Certain character strings, known as sequences, act as commands when you transmit them across a virtual circuit. These sequences break or hold the circuit currently being used. If you want to send these sequences as data (rather than use them to affect the circuit), you must precede them with a character known as the Data Link Escape Character. ■ **Note:** The Data

Link Escape Character for an IBM PC or PC XT is Ctrl-P and cannot be changed. You can change the Data Link Escape Character for other network workstations using the Configure program.

HELP FEATURE

Two commands allow you to review the parameters set for your terminal and the list of commands you may use.

Using the Help Command

This command gives you a list of all network commands you can enter from your terminal. To request Help, type the following:

```
?[CR]
```

A typical response to the Help command is as follows:

```
>?
connect
list
?
```

Here, the only commands you can issue are Connect, List, and ? (Help).

Using the List Command

The List command displays your terminal's physical address, connection names, and the current values for Disconnect and Hold Sequences (explained later in this chapter). If you are using a terminal different from the one you usually use, it is a good idea to use List to review the terminal's characteristics before you begin working. ■ **Note:** Although LAN commands may be entered as a single character using the Command Abbreviation feature, all of the following command examples will spell the complete command word for instructional purposes.

To use this command, enter:

```
LIST[CR]
```

The response will resemble the one shown below.

```
>LIST
backspace = /bs/  line-delete = /del/  disconnect = /esc/P/

echo = (on)  word  complete = /ht/  hold = /stx/

identities =
(1)  479b3
accounting  7
chris's  crt
```

In this example, the terminal's connection resource names are accounting 7 and chris's crt. Its address is 479b3. The (1) indicates that chris's crt is on Network 1.

The List command also displays the characters you would use for the Disconnect and Hold sequence and for command entry and editing. Most of the characters displayed are ASCII nonprinting characters and the keystrokes required to enter them may vary depending on the type of terminal you are using. For example, the Word Completion value shown as /ht/ represents the Tab key on many terminals but can also be entered as Ctrl-I on some terminals. Refer to Table 5-4 in Chapter 5 for a list of ASCII nonprinting characters and their keyboard equivalents.

CONNECTING TO NETWORK RESOURCES

The commands discussed below enable you to establish a virtual circuit, put a circuit on hold, and reestablish a connection currently on hold.

Using the Connect Command

Use this command to establish a connection (virtual circuit) between your terminal and another device or service. You can refer to the device or service by using its network address or by using its rotored or unique name. Sample commands are shown below. The command abbreviation feature allows the use of "c" instead of typing the entire command word "CONNECT," which is used here for instructional purposes.

To connect to a resource using its unique name, enter:

CONNECT SUSAN'S CRT[CR]

To connect to a resource using its rotored name, enter:

CONNECT HOST 1[CR]

To connect to a resource using its physical address, enter:

CONNECT 324b2[CR]

A series of connected networks is known as an Internet. You can connect to a resource on a different but connected network by issuing a global Connect command. To do so, place an asterisk before the name of the device or service:

CONNECT *SUSAN'S CRT[CR]

Codex LAN responds to the Connect command with the message:

connecting . . .

If the connection is successful, you will see a response message which displays the network address of the device, such as:

connecting . . . (1) 6b1 success

Both devices then enter Connected mode. If the connection was not successful, an error message is displayed. Samples are shown below:

Sample 1 **>CONNECT susan's crt**
connecting...
No response for specified name

Sample 2 **>CONNECT 66b2**
connecting...
Resource is busy

Under some circumstances, you may receive a decimal error code instead of a message. These numbers are usually associated with problems on the network. Error codes and their meanings are listed in Chapter 13 of this guide.

Using the Hold Sequence

The Hold sequence works like the hold button on a telephone. It is a special set of characters that you transmit to the device with which you are connected. It allows you to establish a second connection without breaking the first one you created. The first connection is put on hold (the device remains in Connected mode), and your terminal returns to Command mode.

A terminal can use this feature only if it has been configured for multiple virtual circuits (see Chapter 6). Through the Configure program you can allot up to three circuits at one time to one user. You also configure the Hold sequences for devices on the LAN. This sequence must be entered exactly as you enter it in the Configure program.

If you configured your terminal's Hold sequence as QXZ, and it is connected to the network device Host 1, when it transmits the string of characters QXZ the following message displays:

```
Virtual circuit 1 on hold
You may enter Codex LAN commands
>
```

Your terminal is in Command mode, but Host 1 is still in Connected mode. If you then issue a List command, this connection would appear as:

```
Virtual circuit 1 to
(1) 352b1
host 1
```

Even though you are not using the circuit, no other device can connect to Host 1 because Host 1 remains in Connected mode.

Codex LAN software assigns a sequential number to each circuit you create, beginning with Circuit 1 and ending with Circuit 3. If you have established three virtual circuits, and you disconnect Circuit 2, the List command will display Circuits 1 and 3. The software does not renumber circuits to fill numerical gaps. The next circuit you create becomes the new Circuit 2.

If you want to transmit your Hold sequence as data, you must precede it with the Data Link Escape Character discussed in the Entering Commands section earlier in this chapter.

Using the Resume Command

Use this command to reestablish a connection you have placed on hold. When you establish more than one circuit, Codex LAN identifies each circuit by number. You must use this number to identify the circuit you wish to resume. If you have created more than one virtual circuit, use the List command to determine the circuits' numbers. This command displays the circuits that are on hold, the connection names of the resources at the other end of the circuit, and the number of the circuit.

To reestablish a connection with Virtual Circuit 2, enter the following command line:

RESUME 2[CR]

A typical Resume command and network response would appear as:

```
>RESUME 2[CR]
Resumed
```

DISCONNECTING

The commands discussed below enable you to break circuits that you have created.

Using the Abandon Command

You use the Abandon command to disconnect a virtual circuit that you have placed on hold. You must use the number of the virtual circuit that you wish to break. If you have established more than one circuit, use the List command to display the circuits on hold and their numbers. Enter the command as shown below:

ABANDON 2[CR]

When the command is successful, you will not receive a screen message. When you enter a number that does not correspond to an existing circuit, you receive the message:

```
No such virtual circuit
```

Using the Disconnect Sequence

The Disconnect sequence is a special set of characters that you transmit to the device with which you are connected. When you transmit a Disconnect sequence, you break the active virtual circuit between your terminal and another device. Your terminal enters Command mode, and the other device enters Idle mode.

As Network Administrator, you determine a device's Disconnect sequence. As with the Hold sequence, it must be transmitted exactly as you enter it in the Configure program. ■ **Note:** The Disconnect sequence for an IBM PC or PC XT is always Ctrl-T and cannot be changed.

If your terminal's Disconnect sequence is yqy, and you are connected to Host 1, when you transmit the string of characters yqy, the standard Codex LAN prompt (>) appears on your screen. Host 1 enters Idle mode, and other users can connect to it.

To transmit your Disconnect sequence as data, i.e., to negate its control function, you must precede it with the Data Link Escape Character (explained in the Entering Commands section earlier in this chapter).

Using the Quit Command

If your terminal is a command/data device, you would use the Quit command to exit Command mode and enter Idle mode. When your terminal is in Idle mode, other devices can establish a connection to it. This command also disconnects all virtual circuits that you have placed on hold.

The Quit command and system response are shown below:

```
>QUIT[CR]
Idle
```

If you have circuits on hold when you issue this command, the system asks if you want to abandon those circuits. If you answer "Y" for Yes, the circuits are broken and your terminal enters Idle mode. The screen will look like this:

```
>QUIT
Abandon existing virtual circuits? (Y/N) Y
Idle
```

If you answer "N" for No, the Quit command is aborted and your terminal remains in Command mode. The screen will look like this:

```
>QUIT
Abandon existing virtual circuits? (Y/N) N
>
```

The Quit command causes all values that were changed through the Set command (explained in the next section) to return to the default values you established in Configure.

Using the Personal Computer Quit Command

If you have an IBM PC or PC XT and you want to exit the Terminal Emulator program and return to PC DOS, enter the following command. ■ **Note:** The command word "quit" must be entered in lowercase characters.

Ctrl-Wquit[CR]

Quitting the emulator program does not break the connection between your computer and the resource with which you were connected. Thus, you can switch back and forth between PC DOS and the network resource. When you reinvoke the emulator (using CATE), you are still connected.

Because Ctrl-Wquit does not break a circuit, the network resource you were using remains in Connected mode, and no other user can connect to it. Remember to break the circuit when you are finished using it by transmitting a Disconnect sequence.

MAKING CHANGES WITH THE SET COMMAND

Under certain circumstances network users may want to make temporary changes to particular characteristics of their terminal. They would do so by using the Set command. You should urge network users to consult with you before they use the Set command. You can use this command to turn a terminal's local echo on or off and to change the Disconnect and Hold sequences. ■ **Note:** This command cannot be used with the IBM PC or PC XT.

Local Echo

Codex LAN has an automatic Local Echo facility where an Entryway echoes back all characters typed by the devices connected to it. Since different types of devices may or may not echo characters, the Local Echo facility allows network users local control over echoing. Because most intelligent resources such as host computers already have their own echo facility, a terminal's local echo is usually turned off.

To turn the echo on, use the Set command as shown below:

SET ECHO ON[CR]

To turn the echo off, use this form of the Set command:

SET ECHO OFF[CR]

Remember that the List command displays the current status of the local echo.

Disconnect and Hold Sequences

Disconnect and Hold sequences may also be entered locally using the Set command. The new sequences, however, are only valid during one virtual circuit connection. They revert back to the sequences you selected in the Configure program once they are used to disconnect the current circuit.

Since IBM PCs and PC XTs cannot initiate multiple virtual circuits, Hold sequences do not apply.

To change the Disconnect or Hold Sequence locally, enter the Set command, specify which sequence you are changing, then enter a new value enclosed in slashes.

For example, to change a Disconnect sequence to ++@@, enter the following command line:

SET DISCONNECT /++@@/[CR]

To change a Hold sequence to !$!, enter the command as shown below:

SET HOLD /!$!/[CR]

You should be careful not to choose sequences that are likely to occur as part of the data you are transmitting. Certain other rules govern your choice of a Hold or Disconnect sequence.

The Disconnect and Hold sequences can be up to four characters in length, and can be made up of both printing and nonprinting ASCII characters. A printing ASCII character is any of the alphabetic, numeric, or special characters on your keyboard. A nonprinting ASCII character is one that ordinarily has no displayed representation, such as an Escape or Carriage Return keystroke.

To include nonprinting characters in your Disconnect or Hold sequence, enter the keystrokes as you normally would. For example, to change the Disconnect sequence to Ctrl-TCtrl-P, proceed as follows:

1) Type SET.
2) Press the Spacebar and type DISCONNECT.
3) Press the Spacebar and type a slash (/).
4) Hold down the Ctrl key and type T.
5) Hold down the Ctrl key and type P.
6) Type another slash (/).
7) Press your Return or Enter key

The first character of a sequence cannot be repeated in either that or another sequence except as the first character. The following examples illustrate this restriction:

- A Disconnect sequence of "abad" is illegal because the letter "a" is repeated in the sequence.
- A Disconnect sequence of "abc" and a Hold sequence of "daf" is illegal because the "a" in the Hold sequence is not the first character in that sequence.

The same character string cannot be used for both a Disconnect and a Hold sequence.

EXTENDED COMMAND SET

When you run the Configure program, you can select a privileged, or extended, command set for some users that is not available to other users. These commands are considered privileged because they allow you to drastically modify network operation. You can, for example, disconnect a virtual circuit with no intervention from an ordinary user, or likewise, create virtual circuits by interconnecting two other devices.

If your network is large, you may configure a number of application processors for the privileged command set, or, with a smaller network, you may configure just one board for extended command set use (see Chapter 6). ■ **Note:** The 4810 PC XT LAN Manager terminal does not support access to this extended command set. You should, therefore, configure at least one application processor board in an Entryway for these commands (see Chapter 6).

The extended commands are Examine, Interconnect, and Disconnect, and are discussed in the following paragraphs.

Using the Examine Command

This command allows you to display the current status of a network device. A global Examine command displays the status of any device on the Internet.

To display the current status of device 432A3, enter the following command line:

```
EXAMINE 432A3[CR]
```

If the device is on a different but connected network, enter an asterisk before the resource's name or address:

```
EXAMINE *host1[CR]
```

The system displays a message consisting of the resource's network address, a list of its names, and its status. Sample responses are shown below:

Sample 1
432A3
vax7
is free

Sample 2
21b2
tom's crt
is attempting to set up a permanent circuit

Sample 3
56C1
host3
is connected to
65B2
tty4
donna

In Sample 3, the system displays the physical address, rotored name, and unique name of the resource Host 3 is connected to.

■ **Note:** If the specified resource is connected to a PC or PC XT, the Terminal Emulator name you supply using PC Names is the name displayed.

When the system cannot determine the status of the resource, the following message is displayed:

```
65b3
tty7
is in an unknown state
```

Using the Interconnect Command

The Interconnect command allows you to create a virtual circuit between two other resources on the network. With the Connect command, you establish a circuit between your terminal and another network device or service. With Interconnect, you create a circuit between two other devices or services.

When you issue an Interconnect command, you must specify the name or address of both endpoints of the circuit. End each name or address with a Return or Enter as shown in the following example:

```
INTERCONNECT 7302B1[CR] 42C3[CR]
```

After the first Return or Enter, Codex LAN displays an "and" and waits for you to enter the second resource name. As with Connect, the system returns a response to your command. A sample command and response sequence is shown below:

```
>INTERCONNECT james and host3[CR]
connecting . . . (1) 4478A4 and (1) 821B3 success
```

The response displays the address of the two resources. Numbers in parentheses identify the network on which the resources reside.

To issue a global Interconnect, enter an asterisk before the names or addresses. The response line displays the network number for each resource. A sample command and response sequence is shown below:

```
>INTERCONNECT *VAX1[CR] and *JAMES[CR]
connecting. . .(4) 32C1 and (1) 478A4 success
```

Using the Disconnect Command

The Disconnect command allows you to break a virtual circuit between two other resources anywhere in the Internet. You issue the command by specifying a device at one end of the circuit you wish to break, as follows:

DISCONNECT 483A4[CR]

The system displays a status message as it does with the Examine command and asks you to confirm the Disconnect command. You must respond with a "Y" (yes) to break the circuit; otherwise, the command is ignored. A sample command and response sequence is shown below.

```
>DISCONNECT vax7[CR]
(1)56C3
vax7

is connected to
(1) 77D3
tom's crt
disconnect? y
disconnecting...success
```

If you specify a resource that is not connected, the confirming prompt does not appear.

This command also disconnects a bound circuit. Initial and demanded bound circuits remain disconnected. In the case of a permanent bound circuit, the Entryway will immediately attempt to reestablish the connection.

Do not confuse the Disconnect command with a Disconnect sequence. You transmit a Disconnect sequence to break the connection between your terminal and another resource. You use a Disconnect command to break the connection between two other network resources.

Chapter 4
Configuration Files

Contents

OVERVIEW

The Configure program is a tool the Network Administrator uses to build configuration files for each Entryway on the network. When an Entryway powers up, it transmits a download request over the LAN. This request includes its physical address. The Download Server responds to the request by looking for the configuration files associated with the Entryway's address. These files list the software modules the Download Server will transmit, or download, over the LAN to the requesting Entryway.

LAN Entryways support different combinations of devices and services, as shown in Figure 4-1. In this example, a host computer is connected to Entryway A, a 4002. Three printers are connected to Entryway B, a 4001. Three video terminals and a printer are connected to Entryway C, a 4001. No user devices are connected to Entryway D, a 4002, which supports the File Transfer service. The devices connected to Entryways A and B, and the video terminals connected to Entryway C are serial asynchronous and use the RS-232 interface. The printer is connected to Entryway C through a parallel interface. Entryways A, B, and C provide Virtual Circuit service for their devices.

Since Entryways A, B, C, and D support a unique combination of devices, services, and communication types, the software which allows them to manage the activities involved in network communication is also unique. It is your task, when running Configure, to define each Entryway's unique operating environment so that the Download Server can transmit the appropriate software when the Entryway is powered on. Configure asks you questions about each individual application processor board within an Entryway, since each board is loaded separately with its own networking software. From your answers to its questions, Configure builds a profile of each Entryway, each application processor board, and each device connected to its I/O module. When you finish each configuration task, Configure writes one of three types of configuration files based on this profile: Device Definition files, Load Characteristics files, and Load Specification files.

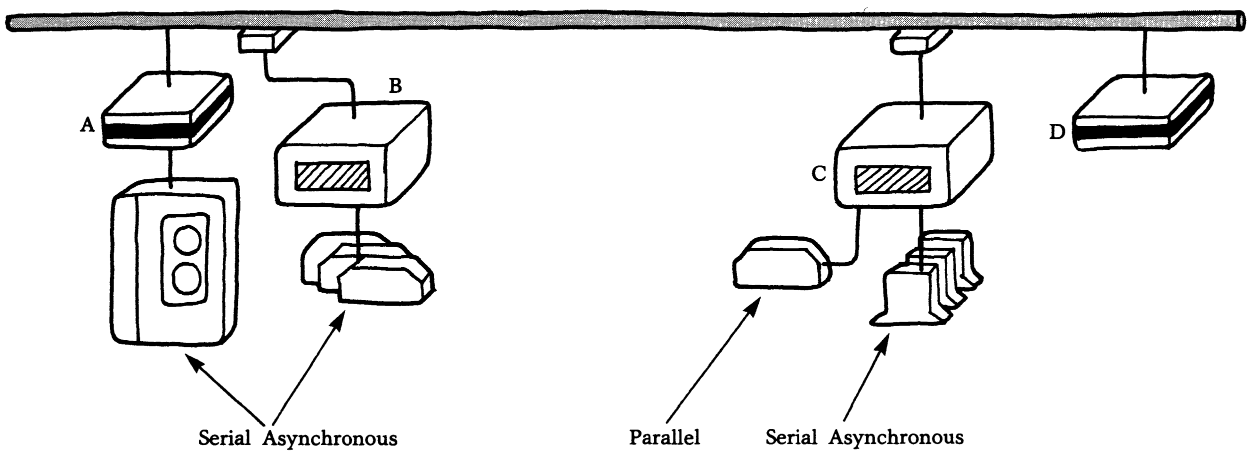


Figure 4-1. Entryway Devices and Services

DEVICE DEFINITION FILES

Using your answers to questions about the device you have connected to an Entryway port, Configure builds a Device Definition file. The device characteristics you supply define a set of conditions with which the networking software must interact in order for the Entryway to communicate with the device. The Device Definition file is similar to DIP switch settings which define the interface between a device and the application processor board to which it is connected.

It is not necessary to create a separate Device Definition file for every device on your network. You can create generic Device Definition files to use as masters and then modify them in the Configure program.

Figure 4-2 shows Entryway C from Figure 4-1 with its Device Definition files labeled. The three video terminals connected to Entryway C in this figure are identical. They are all asynchronous, they have the same character format, the same type of flow control, and so on. They are all connected to serial ports. You only run Configure once to create one generic Device Definition file for all three. The printer is a parallel device with programmable handshaking. You must run Configure again to create a different Device Definition file. The questions asked about the printer will be very different from those asked about the video terminals.

The Configure program will ask you to supply a name for the device you are configuring. You can supply any upper or lower case name, as long as it is seven or fewer numbers or letters, and starts with a letter. It is a good practice to supply a mnemonic name which recalls the type of device you are configuring. The names supplied for the devices in Figure 4-2 are CRT and PRINTR. When the system writes the Device Definition file, after you exit from the Configure program, a "D" is appended to the device name.

LOAD CHARACTERISTICS FILES

Load Characteristics files are created when you configure the individual application processor boards in an Entryway.

To configure a board, you tell Configure the Device Definition filenames of devices connected to the board. You also tell

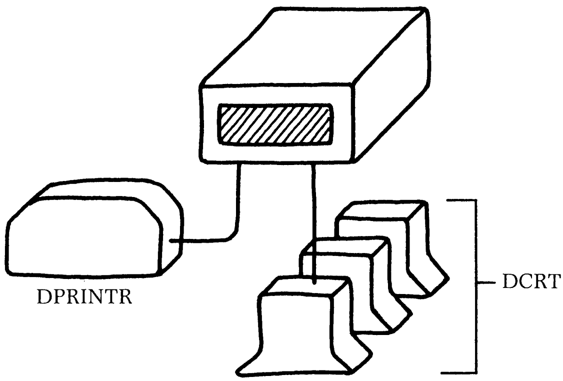


Figure 4-2. Device Definition Files

Configure what communication type and service software the board should be loaded with, what I/O module it has, the connection resource names of its ports, the type of command interpeter it has if it is running the Virtual Circuit service, and other information Configure asks you for.

From this information Configure writes a Load Characteristics file. This file is actually a table of board and device characteristics. You supplied the device characteristics when the Device Definition file was created. One of these tables is downloaded to each active board in an Entryway. The networking software references this table when it manages the various networking activities the board supports. ■ **Note:** If you configure an Entryway to serve as a Local or Remote Bridge (see the sections on Network and Bridge Configuration in Chapter 6), no Load Characteristics file is written. Instead, Configure writes a Load Specification file (discussed in the following section).

Figure 4-3 shows a rear view of Entryway C from Figure 4-2. Callouts identify the LC files written when you finish configuring each board and the LS file written when you have finished the configuration task (discussed in the following section).

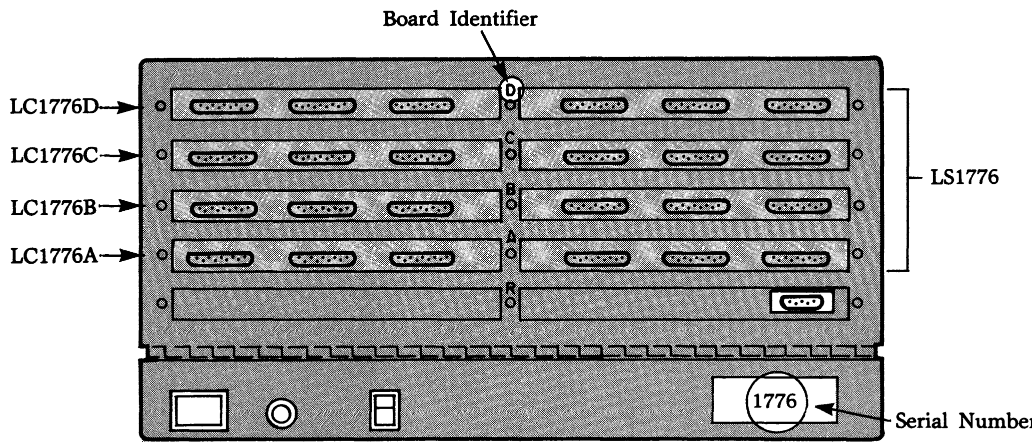


Figure 4-3. LC and LS Files

The name of the Load Characteristics file is the prefix LC, the serial number of the Entryway, and the letter of the board. The serial number of the Entryway shown is 1776. When you configure Board A, the Load Characteristics file LC1776A is written. When you configure Board B, the file LC1776B is written, and so on.

Load Specification Files

Load Specification Files are written after you configure all boards in an Entryway. If you have a 4002 Entryway, you configure only Board A. If you have a 4001, you configure all active boards. When you have finished your configuration, a Load Specification file is written. The name of this file is the prefix LS and the serial number of the Entryway you are configuring. In the example in Figure 4-3, the Entryway has a Load Specification file called LS1776.

This file resides on the fixed disk of the LAN Manager. It lists the names of the software modules required to support the services and devices you have configured for the Entryway. The list includes the general-purpose networking software required by every Entryway. It also includes the special software required, based on your configuration. This includes, for example: Virtual Circuit Service software or Datagram software, asynchronous software or synchronous software, and so on. The list also includes the names of the Load Characteristics files or tables which are to be downloaded to each board.

Chapter 5
Device Configuration

Contents

OVERVIEW

The first step of network configuration is device definition. Each time you add a new device to your network, and a generic Device Definition file does not already exist, you must create one.

The main tasks you will have to perform in configuring a device for the first time are outlined below.

- ***Define device dependent characteristics.*** These characteristics include the data format of the device, its method of flow control, whether it is a DCE or DTE, the behavior of handshaking signals, and so on. Most of the answers to these questions are to be found in your equipment's technical reference manual. An explanation of many of these elements is included here to acquaint you with LAN features with which your device must interact.
- ***Define command interpreter characteristics.*** You do not have to complete this section of the menus if you have not configured your device for virtual circuit initiation via the LAN command interpreter. The questions here allow you to modify control characters used for LAN command entry and editing.
- ***Specify device input/output buffer sizes.*** These questions concern Entryway device buffer sizes which affect data throughput. Normally, you would simply accept the default buffer sizes listed in the menus. Special cases are desribed in which you might wish to change these defaults.

Each of these tasks is discussed in the following sections. Configure menus are displayed when each menu topic is discussed. There is no attempt to present every Configure menu and every item in each menu. Many of the menus shown here are *excerpts only* and are presented only to familiarize you with the type of questions you must answer about your devices.

DEVICE DEPENDENT CHARACTERISTICS

Device dependent characteristics are determined by the communication a device supports and the type of I/O module it is connected to. Configure questions differ depending on which item you select from the following menu.

```
- - - - - -
Entering Menu: Communication Service
- - - - - -
1) Select a communications service for this
   device:
     a) Asynchronous for serial/parallel
        I/O module
     b) Asynchronous for six serial I/O
        module
     c) Synchronous Communication
        Service I (bisync)
     d) Synchronous Communication
        Service II (HDLC/SDLC)
     e) Synchronous Communication
        Service III (DDCMP)
     f) Eight bit parallel for serial/
        parallel I/O module
     g) user driver
```

Asynchronous Device Characteristics

The menu listed on the next page asks general questions about the characteristics of all asynchronous devices, independent of whether they are connected to a Six-Serial or Serial/Parallel Port I/O module.

\- - - - - -
Entering Menu: Device Dependent Characteristics
\- - - - - -

5) Number of data bits per character:
  a) 8 data bits
  b) 7 data bits
  c) 6 data bits
  d) 5 data bits

6) Parity:
  a) Odd parity
  b) Even parity
  c) No parity
  d) Mark parity
  e) Space parity
  f) Ignore parity

7) Number of stop bits:
  a) 1 stop bit
  b) 1.5 stop bits
  c) 2 stop bits

Character Format. The questions illustrated in the preceding menu ask you to define the asynchronous character format of your device. All asynchronous characters begin with 1 start bit, are followed by from 5 to 8 data bits, 0 or 1 parity bit, and end with 1, 1.5, or 2 stop bits. When an Entryway receives data from a device, it strips the start, stop, and parity bits from each character before it is transmitted over the LAN. The receiving Entryway attaches the start, stop, and parity bits to each character before the data is passed on to its device. When a parity check bit is specified and the Entryway detects a parity error, it discards the character. ■ **Note:** Your answers to character format questions are based on the character format of your equipment. You must ensure that your answers reflect the way your equipment has been set up—either by device software configuration, setup commands, or DIP switch settings. If this is not done, the Entryway will not recognize device data.

Flow Control. Flow control allows a device or Entryway, when its buffer is full, to send a signal to stop the flow of data. When the buffer is low again, another signal is sent to restart the transmission. In general, four occurrences on the Codex LAN could cause a buffer overrun of data:

- The source device is transmitting at such a high rate, the source Entryway cannot keep up with it.
- There is heavy traffic over the LAN itself.
- The receiving Entryway cannot keep up with the transmitted data rate.
- The receiving device cannot keep up with the transmitted data rate.

Codex LAN allows you to choose from four different flow control options:

— — — — — —

Entering Menu: Device Dependent Characteristics

— — — — — —

8) Type of flow control this device supports:
a) Characters
b) Dual character flow control
c) RS-232 line signals
d) None

Refer to your equipment technical reference manual for a description of the type of flow control your device supports.

Character flow control uses XON and XOFF characters to stop and restart the flow of data.

— — — — — —

Entering Menu: XON and XOFF

— — — — — —

1) XON character from and to device
2) XOFF character from and to device

Character flow control employs the same XON and XOFF characters from device to Entryway as it does for Entryway to device communication. The XON default character is ASCII ⟨DC1⟩; the XOFF default character is ASCII ⟨DC3⟩. If the device you are configuring uses other XON/XOFF characters, Configure allows you to substitute them for the defaults. To do so, follow the instructions included in the section of this chapter titled Device Command Mode Definition.

Dual Character flow control uses different XON and XOFF characters from Entryway to device, and from device to Entryway.

– – – – – –
Entering Menu: XON and XOFF
– – – – – –
1) XON character from device (<NUL>)
2) XOFF character from device (<NUL>)
3) XON character to device (<DC1>)
4) XOFF character to device(<DC3>)

If you specify any XON/XOFF pair to be <NUL>, you may disable flow control in one direction. This accommodates devices that will accept XON/XOFF characters, but cannot generate them.

RS-232 line signals are also supported by Codex LAN.

– – – – – –
Entering Menu: RS-232 Line Signals
– – – – – –
1) CTS:RTS or DSR:DTR:
   a) CTS:RTS
   b) DSR:DTR

Check your equipment technical reference manual to see which line signals your device employs.

Break Key. If your device has a Break key, Configure will ask you if it should be recognized or ignored. When you answer yes, the port will ignore the signal. When you answer no, the port will recognize it.

If the port recognizes the Break signal, it can be used in several ways:

– – – – – –
Entering Menu: Circuit Deactivation
– – – – – –
1) Disconnect sequence (<ESC>OP)
2) Hold sequence (<NUL>)
3) Break signal should cause:
   a) Disconnect
   b) Hold
   c) Neither

You can configure it as a disconnect or hold sequence (see the paragraph on Circuit Deactivation in Chapter 6). If you choose

"Neither" from the above menu, the port passes the signal across the circuit to the destination device. This is useful when the destination device uses a Break signal for some other control function. Check your equipment technical reference manual to see if your device requires Break key recognition.

Data Link Escape Character. A Data Link Escape character must precede any control character you wish to send over a virtual circuit as transparent data, i.e., negating its control function. Examples of this are the XON/XOFF characters or a disconnect sequence. The ASCII character ⟨DLE⟩ is the default Data Link Escape character; however, you may choose another ASCII nonprinting character. Refer to the section titled Device Command Mode Definition, later in this chapter.

DCE/DTE Port Configuration and Signaling

The ports on a processor board can simulate either a data communications equipment (DCE) interface or a data terminal equipment (DTE) interface. When a port on a board simulates a DCE, it acts like a modem for an attached terminal or host (DTE). When it simulates a DTE, it acts like a terminal or host for an attached modem (DCE). Figure 5-1 illustrates a Serial/Parallel Port I/O Module with DCE and DTE ports.

Most of the devices you connect to your network will be DTE devices using the DCE interface. Examples of these devices would be printers or the video terminals shown in Figure 5-1. One important use of the DTE port to DCE device interface is Codex LAN access for remote devices via a modem.

DCE/DTE Configuration questions differ depending on whether the communication service you select for your device is six-serial asynchronous, serial/parallel asynchronous, or serial/parallel synchronous.

Serial Asynchronous Devices on the Six-Serial Port I/O Module. The Six-Serial Port I/O Module provides six asynchronous RS-232 serial ports, all permanently wired for DCE port to DTE device operation. You can change the operation of a port by connecting to it a null modem or crossover cable. Two pins are jumpered together on the null modem connecter. This indicates to the network software that

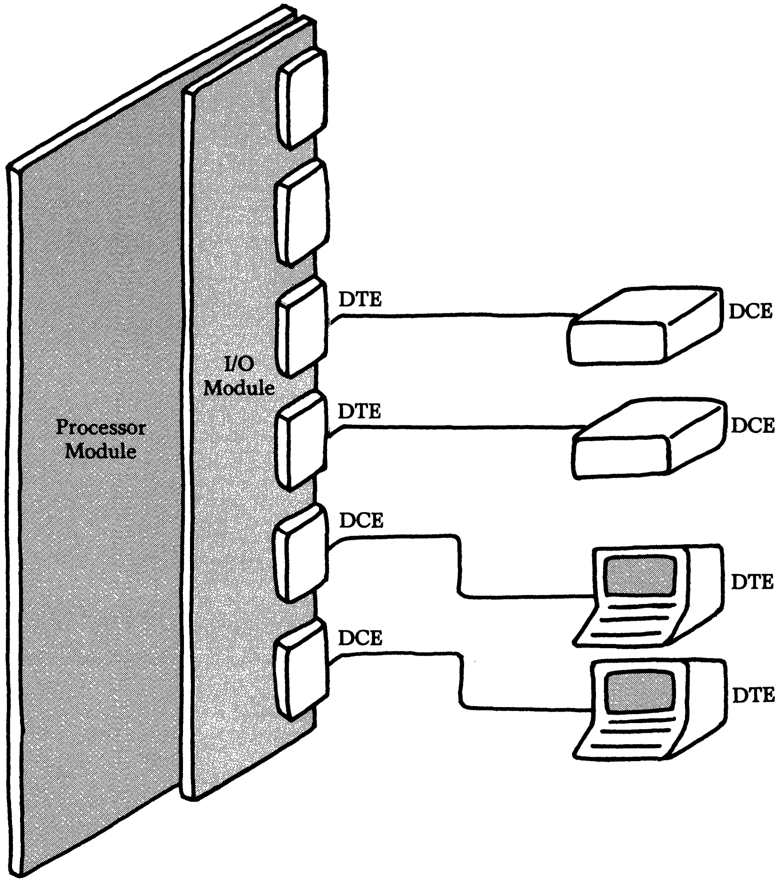


Figure 5-1. DCE and DTE Ports on the Serial/Parallel I/O Module

the port is to be used as a DTE interface, supporting a DCE. Figure 5-2 illustrates a possible configuration of devices attached to the six-serial I/O.

When you configure a device, you must select one of the following options for each I/O port:

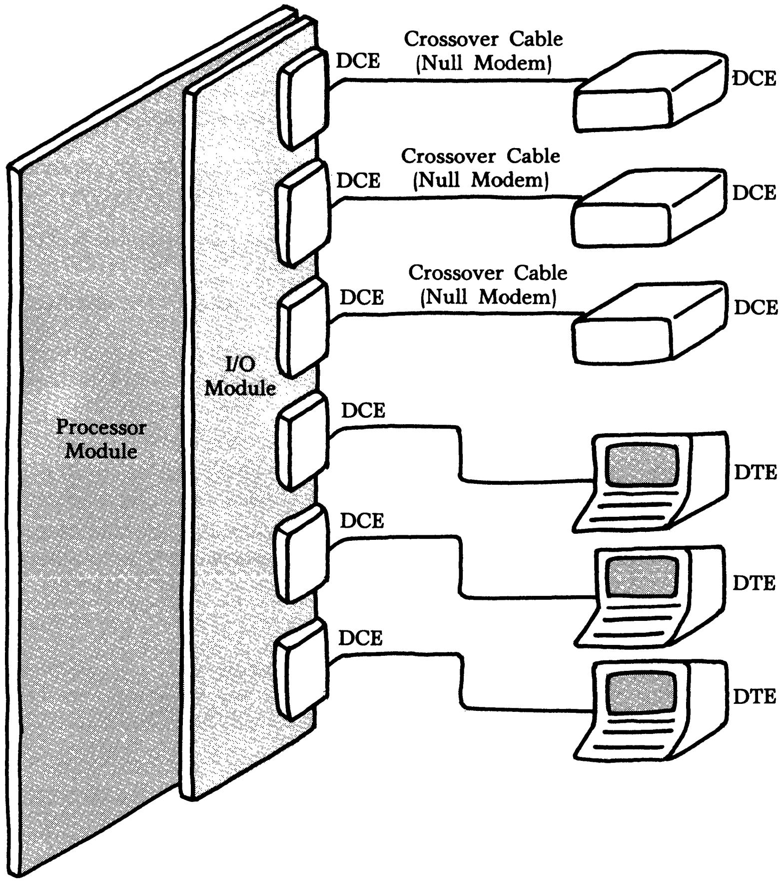


Figure 5-2. Six-Serial I/O Module with Null Modems

\- - - - - -

Entering Menu: Device Dependent Characteristics

\- - - - - -

1) Port should be configured for this device as:
- **a) Port always DCE**
- **b) Port always DTE**
- **c) Port switchable DCE or DTE**

- Select "a)" if you are connecting a DTE.
- Select "b)" if you are connecting a DCE.
- Select "c)" if you want the software to determine mode of operation by the presence or absence of a null modem.

If you select "a)" or "b)" and you later decide to connect a different type of device, you will have to go back into the Configure program and select a different port option. The advantage of the "c)" option is that you can use the port in DCE or DTE mode without being required to run the Configure program each time you switch from one mode to another.

Configure asks you to define the behavior of handshaking signals on the RS-232 lines for each active port you are configuring:

```
- - - - - -
Entering Menu: Signal Behavior
- - - - - -
1) Signal behavior when port is DCE (device is DTE):
   a) If listening or open: CTS, SQ, DSR and DCD
      are true.
   b) If listening or open: RI, CTS, SQ, DSR and
      DCD are true.
   c) If listening and DTR goes true: DCD, SQ, DSR
      CTS go true, command mode enabled.
      Upon connection request: RI goes true,
      await DTR. Then, when DTR goes true:
      RI goes false, DCD, SQ, DSR, CTS go true,
      data transmission enabled. Then, when DTR goes
      false: connection breaks.
   d) If listening: DCD SQ, DSR go true. Then
      when DTR goes true: CTS goes true, command
      mode enabled.
      Upon connection request: RI, DCD, SQ, DSR
      go true, await DTR. Then, when DTR goes
      true: RI goes false, CTS goes true, data transmission
      enabled. Then, when DTR
      goes false: connection breaks.
```

2) Signal behavior when port is DTE (device is DCE):
   a) If listening or open: DTR and RTS are true.
   b) If listening: DTR, RTS go true. Then when DCD is true: connections are enabled. Then, upon connection request: DTR, RTS go true, data transmission enabled.
   c) Same as 2, plus: Then when DCD goes false: connection breaks.
   d) If listening and RI goes true: command mode enabled. Then, when initiation character seen: DTR, RTS go true.
      If DCD is true: connections are enabled. Then upon connection request: DTR, RTS go true, data transmission enabled. Then when DCD goes false, connection breaks.
   e) If listening: DTR, RTS go true. Then, when RI goes true: command mode mode enabled.
      If DCD is true: connections are enabled.
      Then, upon connection request: DTR, RTS go true, data transmission enabled. Then,
      when DCD goes false: connection breaks.
   f) Same as 5, except: Then, when DSR goes false: connection breaks.

The DSR, SQ, and DCD signal pins are permanently wired together. Refer to the *4000 Series LAN Hardware Reference Guide* for a complete list of pin assignments on the RS-232 connector.

The signal behavior options differ for DCE and DTE port modes of operation. Choice "a)" in both menus is the most flexible and can be selected for most command, command/data, and data devices. Check your equipment technical reference manual for information on signal behavior.

Serial Asynchronous Devices on the Serial/Parallel Port I/O Module. The Serial/Parallel Port I/O module is a flexible I/O interface with four serial RS-232 ports and two 8-bit parallel ports. Each serial port can be configured as a DTE (DCE device) or a DCE (DTE Device) interface, and can operate in asynchronous or synchronous mode. The parallel ports can be configured as inputs or outputs, and one of the parallel ports can operate in bi-directional mode.

You can order any of the four serial ports strapped for standard or extended RS-232 signaling. Ensure that the options you select in the Configure program agree with the strapping.

\- - - - - -

Entering Menu: Device Dependent Characteristics

\- - - - - -

1) RS-232 signaling used by this device:
a) Standard
b) Extended RS-232

When you select standard RS-232 signaling, the Entryway recognizes only the CTS, RTS, DTR, and DSR handshaking signals. When you select extended signaling, the Entryway recognizes the standard signals plus RI, DCD, and SQ. Refer to the *4000 Series LAN Hardware Reference Guide* for a list of pin assignments on the RS-232 connector and for strapping information. ■ **Note:** When you configure Port 1 and/or Port 2 for extended signaling, parallel Port 5 cannot be used. Port 1 extended signaling uses four bits of the 8-bit Port 5 interface. Port 2 extended signaling also uses 4 bits of the Port 5 interface. When you configure Port 3 and/or Port 4 for extended signaling, parallel Port 6 cannot be used because the extended signals on each port use 4 bits each of the Port 6 interface.

When you select standard signaling, Configure does not ask you whether the port is a DCE or DTE. The strapping of the board determines the signal behavior of the ports. Unlike the Six-Serial Port I/O module, there is no option for switchable DCE or DTE mode based on the presence of a null modem cable. You switch the operation of the board by changing the strapping.

If you select extended RS-232 signaling, Configure asks you to define the handshaking signal behavior of the RS-232 lines for each active port you are configuring.

\- - - - - -

Entering Menu: Extended RS-232

\- - - - - -

1) Port should be configured for this device as:
a) DCE (device is DTE)
b) DTE (device is DCE)

Configure displays two menus from which you can choose signal behavior options. These menus are somewhat similar to the menus displayed for the Six Serial Port I/O module.

```
- - - - - -
Entering Menu: Signal Behavior
- - - - - -
1) Extended RS-232 signal behavior when
   port is a DCE (device is a DTE):
     a) Always: RI, DCD, SQ are true.
        If listening or open: DSR,
        CTS are true.
     b) If listening or open: RI, DCD,
        SQ, DSR, and CTS are true.
     c) If listening and DTR goes true:
        DCD, SQ, DSR, CTS go true, command
        mode enabled.
        Upon connection request: RI
        goes true, await DTR. Then,
        when DTR goes true: RI goes
        false, DCD, SQ, DSR, CTS go
        true, data transmission enabled.
        Then, when DTR goes false:
        connection breaks.
     d) If listening: DCD, SQ, DSR go
        true. Then, when DTR goes
        true: CTS goes true, command
        mode enabled.
        Upon connection request, RI,
        DCD, SQ, DSR go true, await
        DTR. Then, when DTR goes
        true: RI goes false, CTS goes
        true, data transmission enabled.
        Then, when DTR goes false:
        connection breaks.
```

```
- - - - - -
Entering menu: Extended RS-232, port DTE
- - - - - -
1) Extended RS-232 signal behavior when
   port is DTE (device is DCE):
    a) If listening: DTR, RTS go true. Then when
       DCD or RI go true: command mode enabled. If
       DCD is true; connections are enabled.
       Then, upon connection request: DTR,
       RTS go true, data transmission
       enabled.
    b) Same as 1, plus: Then, when DCD goes
       false: connection breaks.
    c) If listening: DTR, RTS go true. Then,
       when RI goes true: command mode
       mode enabled. If DCD and SQ are true:
       connections are enabled. Then upon
       connection request: DTR, RTS go true,
       data transmission enabled.
       Then, when DCD or SQ go false:
       connection breaks.
    d) Same as 3, except: If DSR is true:
       connections are enabled. Then, when DSR goes
       false: connection breaks.
    e) Same as 4, except:
       Then when RI or DSR goes true: command
       mode enabled.
```

The signal behavior options differ for DCE and DTE port modes of operation. Choice "a)" in both menus is the most flexible and can be selected for most command, command/data, and data devices. Check your equipment technical reference manual for information on signal behavior.

Parallel Devices on the Serial/Parallel I/O Module

The Codex LAN software supports input, output, and bidirectional parallel devices. A typical input device would be a line printer which accepts data from the LAN. A typical output device would be a host computer which sends data over the LAN. A bidirectional device can both send and receive data, simultaneously if desired.

— — — — — —

Entering Menu: Parallel Device Dependent Characteristics

— — — — — —

1) Type of device:
- **a) Input**
- **b) Output**
- **c) Bidirectional**

2) Does the device use external status indicators (Y or N)
3) Should the port delete NUL's from its output data (Y or N)

■ **Note:** Parallel Port 5 of the Serial/Parallel Port I/O module cannot be used if you selected extended RS-232 signaling for serial Port 1 or Port 2. Parallel Port 6 cannot be used if you selected extended RS-232 signaling for serial Port 3 or Port 4.

Extended Parallel Operation. If your device's data handshaking signals are incompatible with the PIO handshaking scheme used by the Codex LAN, you can select programmable handshaking. However, it should be noted that programmable handshaking uses lines from the second parallel port (Port 6) on the Serial/Parallel processor board. For this reason, programmable handshaking is only available on Port 5, and Port 6 cannot be used.

When you configure a parallel port for bidirectional operation, you cannot select programmable handshaking or external status. This is because PIO Channel A absorbs some of the capacity of PIO Channel B when configured in bidirectional mode.

Handshaking. If you select programmable handshaking from the following menu, you must specify the desired polarity of the handshaking signals.

— — — — — —

Entering menu: Handshaking

— — — — — —

1) Type of Handshaking:
- **a) Normal**
- **b) Programmable**

Table 5-1 summarizes the value of these signals.

Table 5-1.
Parallel Port Handshaking Signals

Device Type	*Signal*	*Value*
Input	Data Ready	Used by the Entryway to tell the device there is valid data on the eight data lines.
Input	Data Accepted	The line the device uses to tell the Entryway that it has received the data and is ready for the next byte.
Output	Data Ready	Used by the device to tell the Entryway that there is valid data on the eight data lines.
Output	Data Accepted	Used by the Entryway to tell the device that it has received the data.

External Status Indicators. You may require external status lines for the proper operation of your parallel device. Check your equipment technical reference manual. Like programmable handshaking, external status uses lines from the second parallel port. You would connect your device to Port 5. Port 6 would not be used. Port 5 may be configured for both external status indicators and programmable handshaking.

Synchronous Device Characteristics

The questions asked in the menus that follow refer to devices supporting any of the following synchronous communications protocols:

SCS I	Binary Synchronous Communications (BSC)
SCS II	SDLC, HDLC, UDLC, or BDLC
SCS III	Digital Data Communications Message Protocol (DDCMP).

The synchronous software runs on an application processor board with a Serial/Parallel Port I/O module. Only one type of software (SCS I, II, or III) is downloaded to a board. You should connect to this board only those devices which support that board's synchronous protocol. These devices can be connected to the four serial ports on this module, which must be strapped for extended RS-232 operation. The parallel ports cannot be used. Refer to the *4000 Series LAN Hardware Reference Guide* for further I/O information.

DCE/DTE Port Configuration. Synchronous devices may be connected to the Serial/Parallel Port I/O module through a DCE or DTE interface. You can order this board with each port strapped for either mode of operation. Ensure that the choices you make in the Configure program match the way the board is strapped.

```
- - - - - -
Entering Menu: Device Dependent Characteristics
- - - - - -
1) Port should be configured for this device as:
    a) Port is DCE (device is DTE)
    b) Port is DTE (device is DCE)
```

Switched Line Support. Switched line support refers to the type of modem or type of communications line used in device-to-Entryway communication.

```
- - - - - -
Entering Menu: Device Dependent Characteristics
- - - - - -
2) Switched line support (Y or N)
```

Table 5-2 lists the conditions for which you would answer yes or no to the switched line support question.

Table 5-2.
Synchronous Line Support

Interface	*Option*
DTE Port/DCE Device	Answer Y when a modem is connected to the port via a switched (dial-up) line. Answer N when a modem is connected to the port via a non-switched line.
DCE Port/DTE Device	When you answer Y, the software simulates a switched modem. When you answer N, the software simulates a nonswitched modem.

Half/Full Duplex Device Communication. You are asked to select half duplex or full duplex for the communication facility between an Entryway and a device. This question does not refer to the communication protocol of the synchronous software. The intent is to define the communication line if the port is DCE, or to define the modem if the port is DTE.

```
- - - - - -
Entering Menu: Device Dependent Characteristics
- - - - - -
3) Physical characteristics
     a) Half duplex
     b) Full duplex
```

Table 5-3 summarizes the options available.

Table 5-3.
Half/Full Duplex Device Communication

Interface	*Option*
DCE Port/DTE	Select "a)" when the communication line between Entryway and device is half-duplex.
	Select "b)" when the communication line between Entryway and device is full-duplex.
DTE Port/DCE	Select "a)" when the modem connected to an Entryway is half-duplex.
	Select "b)" when the modem connected to an Entryway is full-duplex.

DEVICE COMMAND MODE DEFINITION

When you configure an asynchronous command or command/data device, you are asked to choose or accept default control characters for command entry and editing.

```
- - - - - -
Entering Menu: Command Interpreter Characteristics
- - - - - -
1) Initiation character (<NUL>)
2) Backspace character (<BS>)
3) Line delete character (<DEL>)
4) Command completion character (<HT>)
```

The above menu lists the system defaults for command interpreter control characters. They are listed as nonprinting ASCII characters which correspond to keys on the keyboard of the terminal you are configuring.

Chapter 3 discusses the functionality of these control characters. The menu defaults for these characters are described below.

- ***Command Initiation.*** If you accept the default ASCII <NUL> command initiation sequence, striking any key will invoke the command interpreter. You might wish to change this to add a measure of security to a terminal. This ensures that only users who know the command initiation sequence can invoke the LAN command interpreter.

- ***Backspace Character.*** The default Backspace character, an ASCII ⟨BS⟩, implements a terminal's Backspace key. If the terminal you are configuring does not have a Backspace key, a Ctrl-H performs the same function.
- ***Command Completion.*** The default command word completion character is an ASCII ⟨HT⟩, which translates to a terminal TAB character or Ctrl-I.
- ***Line Delete Character.*** The standard line delete character is an ASCII ⟨DEL⟩. This is equivalent to a terminal's Delete key. If the terminal you are configuring does not have a Delete key, you will have to assign a new ASCII control sequence.

As a Network Administrator, you may wish to change these control characters. You might do so, for example, if your users are familiar with their own terminal's control characters and you wish to preserve these for ease of operation. You can give each device a different set of control characters, including those devices attached to ports on the same board. Or, you can assign the same control characters to every user. When the users of the devices you have configured invoke the List command, the control characters you have configured for them will display.

To change the menu defaults, you would enter next to each menu question the standard ASCII abbreviation for the new control character, enclosed within angle brackets. For example, if you wish to change the control character to a Ctrl and X (holding the Ctrl key down while pressing x), you would enter ⟨CAN⟩ next to the menu question.

Table 5-4 lists the ASCII nonprinting characters you may choose from and their keyboard equivalents.

Table 5-4.
ASCII Non-Printing Characters

ASCII	*Keyboard*	*ASCII*	*Keyboard*
NUL	Any key	SUB	Ctrl-Z
DLE	Ctrl-P	LF	Ctrl-J
SOH	Ctrl-A	VT	Ctrl-K
DC1	Ctrl-Q	ESC	Escape key or Ctrl-[
STX	Ctrl-B	FF	Ctrl-L
DC2	Ctrl-R	FS	Ctrl-\
DC3	Ctrl-S	CR	Carriage Return or Ctrl-M
DC4	Ctrl-T	GS	Ctrl-]
EOT	Ctrl-D	SO	Ctrl-N
ENQ	Ctrl-E	RS	Ctrl-↑
NAK	Ctrl-U	SI	Ctrl-O
ACK	Ctrl-F	US	Ctrl-←
SYN	Ctrl-V	DEL	Delete key
BEL	Ctrl-G	BS	Backspace key or Ctrl-H
ETB	Ctrl-W	HT	Tab key or Ctrl-I
CAN	Ctrl-X	EM	Ctrl-Y

■ **Note:** Do not select the control characters ⟨DC1⟩ or ⟨DC3⟩ if you have configured your device with character flow control and have selected the default ⟨DC1⟩ and ⟨DC3⟩ characters as your XON and XOFF.

- *Command Echoing.* Any command or command/data device may be configured for command echoing.

```
– – – – – –
Entering Menu: Command Interpreter Characteristics
– – – – – –
5) Should NIU commands be echoed (Y or N)
6) Should NIU responses be suppressed (Y or N)
7) Lower case (Y or N)
8) Command interpreter timeout (*Y or N)
```

If you select echoing, the device will display data sent to the Entryway. A terminal with command echoing, for example, would allow a user to see what is being typed. Some terminals, however, have their own local echo facility. Do not configure these devices for command echoing. If you do, all characters typed will display twice on the screen. You should only select command echoing for a terminal which does not have a local echo facility.

- ***Entryway Response Suppression.*** When you suppress Entryway responses to a device, the device will not receive Entryway messages in response to its commands.
- ***Lowercase Support.*** If your device does not support lowercase characters, answer N to this question. Ordinarily, Entryway responses display in upper and lowercase characters. Devices that do not support lowercase characters (i.e., do not map them to uppercase) receive all Entryway command responses in uppercase letters.
- ***Command Interpreter Timeout.*** If you choose a command interpreter timeout for the device you are configuring, you can accept the Codex LAN default timeout of 288 seconds, or you can change it. To change it, enter a decimal number between 1 and 1048 when the Command Interpreter Timeout menu is displayed. If you are configuring a command device, the timeout value has no effect (see the section titled Using the Network in Chapter 3).

BUFFERS AND THRESHOLDS FOR ASYNCHRONOUS DEVICES

Each application processor board allocates device input and output buffer areas for each active port. The input buffer stores data the device sends to the Entryway. The output buffer stores data the Entryway sends to its device. Asynchronous device buffers range in size from 2 to 254 bytes. Configure allows you to select input and output buffer sizes.

\- - - - - -

Entering Menu: Asynchronous Device

\- - - - - -

3) Input buffer size (6...*64...254)
4) Output buffer size (2...*64...254)
5) Flow control: input buffer stop device threshold (4...*42...63)
6) Flow control: input buffer restart device threshold (2...*21...41)

The respective sizes of the input and output buffer can affect data throughput. You would select the default buffer sizes for character mode terminals. If, however, you are configuring a device that sends data to its Entryway in high-speed bursts, select an input buffer size approximately four times larger than the output buffer size.

The asynchronus software controls the flow of data into the input buffer. The software stops the data flow from the device (via character or line-signal flow control) when a user-defined number of bytes are stored in the buffer. This number is called an Input Buffer Stop Device Threshold. The software sends a signal to the Entryway to resume sending data when the byte count in the buffer reaches a specified lower limit. This is called the Input Buffer Restart Device Threshold.

The default values for the two thresholds are based on the buffer sizes you select. The default Stop threshold is two-thirds of the input buffer size. The default Restart threshold is one-third of the buffer size. It is recommended that you accept the default values for the two thresholds. If you do select an input buffer size larger than the default, you should select a Restart threshold of one-third the input buffer byte count.

BUFFERS AND THRESHOLDS FOR SYNCHRONOUS DEVICES

Entryways supporting the synchronous software begin transmitting data to a destination device as soon as the Entryway receives a user-defined number of bytes, rather than buffering an entire message.

– – – – – –

Entering Menu: SCS I, SCS II, SCS III

– – – – – –

2) Input buffer size (256...*312...4108)
3) Output buffer size (256...*312...4108)
4) What is the receiver threshold (Count at which transmission begins to the net)? (8...*30...4108)
5) What is the transmitter threshold (Count at which transmission begins to the device)? (8...*30...4108)

This data transmission technique is called pipelining. The receiver threshold is the number of data bytes the Entryway buffers before network transmission begins. The transmitter threshold is the number of data bytes the Entryway buffers before it transmits data to its device.

Each application processor board allocates input and output buffer areas for each active port. Synchronous device buffers range in size from 256 to 4108 bytes. The device input buffer stores data the device is transmitting to its Entryway. The output buffer stores data the Entryway is sending to its device.

The default values for buffer sizes and thresholds are recommended for most synchronous devices. However, if you expect heavy communication between two devices with different data rates, you may need to adjust the byte counts for better data throughput.

The example in Figure 5-3 shows a remote terminal communicating via a 2400 bps modem (connected to Entryway A) with a host operating at 9600 bps (connected to Entryway B).

The mismatch between the speeds of the two devices is 1 to 4 (2400 x 4 = 9600). This can result in a data overrun at the remote terminal and a data underrun at the host computer. Let's suppose that the largest expected message sent between the two devices is 512 bytes (including the SYN characters and BCC). To correct the situation, you would do the following:

1) Set the receiver threshold of Entryway A to three-fourths of the largest expected message (.75 x 512 = 384). This avoids a data underrun at the 9600 bps device because it increases the size of the message sent.

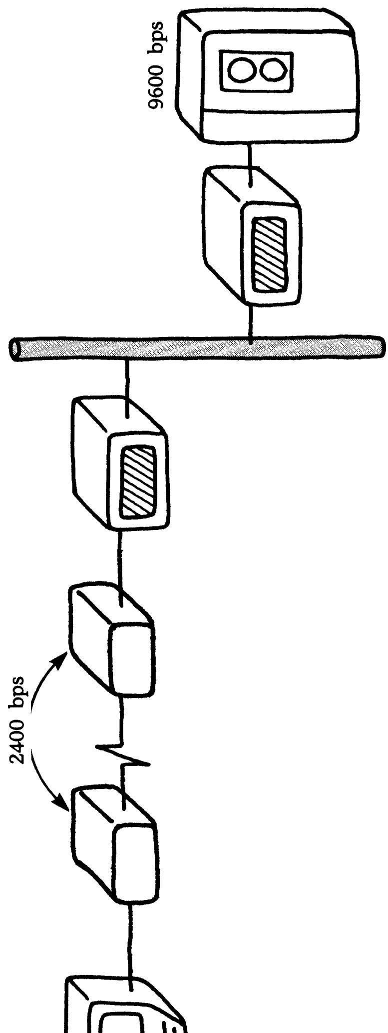


Figure 5-3. Potential Data Overrun and Underrun

2) Set the transmitter threshold of Entryway A to the default: 30 bytes. This allows the Entryway to send smaller messages to its device so that it is not overrun by data.

3) Set the receiver and transmitter thresholds of Entryway B to the default: 30 bytes.

If you had a device sending data at 4800 bps to a 9600 bps device, you would configure the receiver threshold of the slower device's Entryway at one-half of the largest expected message. For a device sending at three times the speed of another device, configure the slower device's Entryway for a receiver threshold equal to two-thirds of the largest expected message.

Note that whenever you configure one Entryway for a speed mismatch, you must configure the other with corresponding receiver and transmitter thresholds.

You should select input and output buffers that are the same size or larger than their corresponding thresholds. The device input buffer corresponds to the Entryway receiver threshold and the device output buffer corresponds to the Entryway transmitter threshold.

The number of active ports on a board can affect the amount of buffer space available to each device. When you allocate more buffer space than the board can provide, Configure displays the following error message:

Space required by this configuration is too large by # bytes.
Reconfigure with smaller buffers.

The # sign represents the number of bytes which exceed the buffer limit. To solve this problem, you must reconfigure one or more active ports for a smaller buffer size.

Chapter 6
Entryway Configuration

Contents

OVERVIEW

Each time you add an Entryway to the LAN, you must create Entryway configuration files through the Configure program. If you change the type of equipment connected to an Entryway that has already been configured, you must modify the Entryway's configuration files. If you change any of the service characteristics of an Entryway, such as its connection resource names or the type of command interpreter, you must modify its configuration files.

The main tasks you will have to perform in configuring an Entryway for the first time are outlined below:

- ***Board Configuration.*** Define the communication service software which will be downloaded to the application processor board on system startup (Asynchronous, SCS I, SCS II, or SCS III). Select the service software which will be downloaded to a board (Virtual Circuit Service, File Transfer Service, and so on.)
- ***Port Configuration.*** Define the devices attached to each port of an I/O module. To do so, supply the name of the generic device definition file you have created for the device you will be connecting. Select the port data rate appropriate to it. Define the service characteristics of each active port. For example, if you select Virtual Circuit Service, supply the type of command interpreter, the port connection resource name, the method of circuit initiation, and so on.
- ***Entryway Configuration.*** Configure all active ports on the application processor board. Configure all active processor boards within the Entryway.

BOARD CONFIGURATION

Configure asks you to choose one service for each application processor board you configure. Notice in the following menu that the Virtual Circuit Service, Datagram Service, and Ethernet Data Link Service are services provided for devices connected to the I/O module ports of the board. The File Transfer Service runs on a board with no connected devices. You must still configure this board.

```
- - - - - -
Entering Menu: Board Function
- - - - - -
1) What function will this board provide:
     a) Virtual Circuit Service for attached devices
     b) Datagram Service for attached devices
     c) Ethernet Data Link Service for attached devices
     d) Network Debugging Service
     e) File Transfer Service (SFT)
     f) Network Configuration Facility
     g) Supporting System board in dual board environment
     h) User Programmable Board in dual board environment
     i) user written service in single board environment
     j) user defined load
     k) none (ignore this board)
```

Configure, however, does not ask you for the names of existing Device Definition files. You can configure a 4001 Entryway with inactive boards, i.e., boards that will not support devices or services at the present time. To do so, choose "k) none (ignore this board)". If you configure the board up to this point in the menus, the Download Server will download enough software to the board so that the status lights will not continually blink as they do when the board is requesting a download. (See the *4000 Series LAN Hardware Reference Guide* for an explanation of status lights.)

The rest of this chapter discusses Configure questions relating to boards running the Virtual Circuit Service. Chapters 10, 11, and 12 describe items b), c), and e) in the above menu. Chapter 8 provides a guide to operating the Configure program.

The asynchronous communication software on boards configured for Virtual Circuit Service can run on a Six-Serial Port or Serial/Parallel Port I/O module. The synchronous software runs on a Serial/Parallel Port I/O module. When you configure a board to support synchronous devices, no other types of devices (asynchronous, parallel) can be connected to this board.

If you are configuring an asynchronous device, Configure asks you, on a per board basis, whether the devices connected to I/O ports will have access to a standard or extended command interpreter.

```
– – – – – –
Entering Menu: Asychronous driver
– – – – – –
1) Select a command interpreter:
     a) Standard
     b) Extended
     c) None
```

Chapter 3 describes these Codex LAN command sets. When you select a command interpreter type for the board, every port on the board will have the same command set. You cannot, for example, select an extended command interpreter for the board, and then configure Port 5 to drive a device with a standard command interpreter. You cannot configure a port for a command interpreter without first configuring the board for one. If you are configuring a 4001 Entryway, which has multiple boards, you can answer this asynchronous driver question differently for each board.

PORT CONFIGURATION

Configure will ask you to select port data rates appropriate to the devices you are connecting to a processor board. Configure's questions differ for the Serial/Parallel and Six-Serial Port I/O Modules.

Port Data Rates on the Serial/Parallel Port I/O Module

When you configure the Serial/Parallel Port I/O module to support serial or parallel asynchronous devices, you must first select a data rate grouping for the board as a whole.

```
– – – – – –
Entering Menu: Asychronous Driver
– – – – – –
1) Select a data rate grouping:
     a) 37.5, 75, 150
     b) 75, 150, 300
     c) 110
     d) 150, 300, 600
     e) 300, 600, 1200
     f) 600, 1200, 2400
     g) 1200, 2400, 4800
     h) 2400, 4800, 9600
     i) 4800, 9600, 19200
     j) 9600, 19200, 38400
```

2) Select a command interpreter:
- **a) Standard**
- **b) Extended**
- **c) None**

All devices you connect to the board must operate at one of the three data rates you select. For example, if you select "h)", you can connect devices operating at 2400, 4800, or 9600 bps, but you cannot connect a device operating at 1200 bps.

When you go on to configure each port, you select a data rate for the port (and its device) from the group of three data rates you chose for the board.

– – – – – –

Entering Menu: Serial Asynchronous Port
Parallel Port

– – – – – –

5) Data rate for this port
- **a) 4800**
- **b) 9600**
- **c) 19200**

When you configure a board for synchronous device support, you configure it for one data rate. All devices connected to its ports must operate at the same data rate. ■ **Note:** For applications making use of external clocking, the user device provides the clock on the RS-232 interface. The speed you select here does not apply.

– – – – – –

Entering Menu: SCS-II devices (HDLC/SDLC)
SCS-III devices (DDCMP)

– – – – – –

1) Data rate for internally clocked ports:
- **a) 600**
- **b) 1200**
- **c) 2400**
- **d) 4800**
- **e) 7200**
- **f) 9600**
- **g) 19200**

Port Data Rates on the Six-Serial Port I/O Module

When you configure a board with a Six Serial I/O module, you supply data rates on a per port basis. This is because you can configure each port for a different speed.

You can supply Configure with a port data rate or you can select automatic baud rate detection on a per port basis.

— — — — — —

Entering Menu: Serial Asynchronous Port

— — — — — —

Automatic baud rate detection for device initiated circuit?

Auto baud detection is used in cases where the port is initiating a virtual circuit. When you select automatic baud rate detection, the Entryway will automatically detect the data rate of a device connected to the port when the device enters a Carriage Return several times in succession. You would select automatic baud rate detection for devices which may communicate at different data rates in different situations.

One situation in which you might select automatic baud rate detection is when a dial-up modem is connected to a port. Let's suppose a remote terminal with a data rate of 1200 bps calls into the modem. The modem operates at different speeds depending on the speed at which devices call into it. The user at the remote terminal enters a Carriage Return several times in succession prior to issuing the Connect command. The modem connected to the Entryway transmits the Carriage Return signal at the data rate of the remote device: 1200 bps. The Entryway samples the data at one speed after another until it recognizes the Carriage Return signal at 1200 bps. The command prompt is then displayed on the remote terminal.

When you configure automatic baud rate detection for a port, you must configure its command mode initiation character as a ⟨NUL⟩. ■ **Note:** It is recommended that you choose automatic baud rate detection only for devices which operate at one of the following data rates: 300, 1200, 1800, 2400, 4800, 9600, or 19,200 bps.

Even when you specify automatic baud rate detection, Configure still asks you to specify a port data rate because the Entryway needs to know the device speed in cases where a connection is network-originated.

```
- - - - - -
Entering Menu: Network Originated Circuit
               Data Rate
- - - - - -
1) Data rate when the circuit is network generated:
     a) 50
     b) 75
     c) 110
     d) 134.5
     e) 150
     f) 300
     g) 600
     h) 1200
     i) 1800
     j) 2000
     k) 2400
     l) 3600
     m) 4800
     n) 7200
     o) 9600
     p) 19200
```

When you do not choose automatic baud rate detection, Configure asks you the port data rate. Supply the data rate of the device you are connecting to this port.

Generic Device Names

Configure will ask you to supply the name of the device connected to the port you are configuring.

The name you enter must be the name of a device for which a Device Definition file exists. If you have not already configured a device, you would answer Y to the question:

```
Modify generic device characteristics?
```

This gives you the opportunity to create a device definition file during Entryway configuration.

Connection Resource Names

Configure next asks you if you wish to add or delete any connection resource names for the port you are configuring. Refer to Chapter 3 for a discussion of connection resource names before you complete this section of Configure menus. Configure asks about connection resource names by displaying a list of existing names for the port you are configuring, and prompting you to add or delete names from the list. If you are configuring a port for the first time, Configure's list of names will look like these:

```
- - - - - - -
Entering Menu: Add Port Name
- - - - - - -
1) The following names exist for this port or service:
     a) 8756a1

Type name, or N: (N) ?
Is this a rotored (non-unique) name? (Y or N)


- - - - - - -
Delete Port Names
- - - - - - -

1) The following names exist for this port or service:
     a) 8756a1
Select the name to be deleted, or type N: (N) ?
```

The physical address of the port (Entryway serial number 1776, Board A, Port 1) is listed as a port name. You cannot delete this name, since it is a device address. You can, however, add other names, or aliases, to the port. Naming conventions are discussed in Chapter 3.

To add a name, you enter Y next to the question "Add any port names?" and then enter a connection resource name. Configure stores the name, and then displays an updated list. The new list contains the physical port address and the alias you have just assigned. You can repeat this process to add other names to the list.

To delete a name, you enter Y next to the question "Delete any port names?" and enter the letter of the name you wish to delete. Configure displays an updated list of names.

Configure asks you whether the name you entered is a rotored name:

```
Is this a rotored (non-unique) name?
```

Enter Y when the name will refer to more than one port; enter N when the name refers to one port only.

Configure performs consistency checks to ensure that when you assign a unique name to a port, you have not assigned this name to other ports as a unique or rotored name. If you have done so, the following error message is displayed:

```
This name already exists on this board and is not unique.
```

When you assign a rotored name, and the name is the unique name of another resource you have configured, this error message is displayed:

```
This name already exists on this board and is unique.
```

The connection resource names you supply here are not tied to the specific port you configure, although the connection address is. You may want to change the location of a device while keeping the device connection name. To do so, you must run Configure to delete that connection resource name from the old port, and add the name to the new port.

Virtual Circuits for Asynchronous Devices

When you configure a port, you must tell Configure whether the device on that port can initiate a virtual circuit connection. If the device can initiate a connection, you must choose one of four methods of circuit initiation. The first three types of circuits are bound circuits. The fourth choice implements a LAN command interpreter for that port.

```
------
Entering Menu: Circuit Initiation:
------

1) Method of circuit initiation:
     a) Initial circuit
     b) Permanent circuit
     c) Demanded circuit
     d) Command interpreter

2) Will this device also be able to accept incoming connection
   requests (Y or N)
```

Bound Circuits. Bound circuits are described in Chapter 3. Remember that the method of bound circuit initiation you select determines what the Entryway will do when the circuit is deactivated.

- An Entryway does not attempt to recreate an initial circuit once it has been broken.

- An Entryway continually attempts to recreate a permanent circuit each time it is broken.
- An Entryway will attempt to recreate a demanded circuit only when the terminal user requests it by entering a demanded circuit initiation character.

If you choose to configure an initial, permanent, or demanded circuit for a port, you must supply the Configure program with the name of the resource at the other end of the circuit. If you do not supply a name, there will be no bound circuit.

```
- - - - - -
Entering Menu: Bound Circuits
- - - - - -
1) Name of the resource at the other end of
   the circuit (<NUL>)?
2) Should the circuit be allowed to cross
   bridges (Y or N)?
```

The name you supply can be either the name of a device or a service. Refer to Chapter 3 for a discussion of device and service names. You can use the resource address (1776A2) or the resource name (TOM'S CRT). You must also indicate whether the circuit can cross bridges. Your answer to this question determines whether a global search will be performed for the named resource.

When you configure a bound circuit, you configure only the port which initiates the circuit. You would not configure the connection resource to initiate the same bound circuit. If you do configure both end points of a bound circuit, data will not be passed and the Disconnect sequence will not work.

When you configure a demanded circuit, you must also configure a demanded circuit initiation character. When the Entryway sees this character, it attempts to set up the demanded connection. You can enter an ASCII nonprinting character, or a string of up to four printing characters for your initiation sequence. Refer to Table 5-4 in Chapter 5 for a list of permitted ASCII characters.

Session-Oriented Circuits. Session-oriented circuits are initiated by devices issuing the Codex LAN Connect command. A device can issue Codex LAN commands only if the port to which it is attached is configured for a command interpreter, and if you have configured the board for one. If you select a command interpreter for a port on a board which has not been configured for a standard or extended command set (see the section titled Port Configuration), Configure displays the following error message:

Command Interpreter is required but not present.

To correct this error, you would go back to the Asynchronous Driver section of the menu and configure a standard or extended Command Interpreter for the board as a whole.

You should normally guarantee at least one circuit for each port you configure for circuit initiation. Otherwise, there is no guarantee that the port will have network access if all of the circuits available on the board are being used (see the following paragraph titled Multiple Virtual Circuits).

```
— — — — — —
Entering Menu: Minimum Number of Circuits
— — — — — —
1) Number of guaranteed circuits for this port:
     a) None
     b) One
```

Multiple Virtual Circuits. Any asynchronous device with a standard or extended command interpreter can be configured for multiple virtual circuits. You first configure multiple virtual circuits for the board as a whole, and then configure, on a port by port basis, each port supporting multiple circuits.

```
— — — — — —
Entering Menu: Multiple Virtual Circuits
— — — — — —
1) Will any port support more than one circuit (Y or N)
```

```
— — — — — —
Entering Menu: Command Interpreter
— — — — — —
1) Will this port support more than one circuit (Y or N)?
```

If you select multiple virtual circuits for a port and you did not select this feature for the board to which the port is attached, Configure displays the following error message:

Multiple virtual circuits are not available on this board.

When you configure a port for multiple virtual circuits, you must allot it a maximum number of circuits. You can allot up to three circuits to each port; however, there is no guarantee that a device can use an allotted circuit if the circuit is already in use. You can, however, guarantee virtual circuits to a port. When you do so, the Entryway reserves the circuits. No other device will be allowed to establish a circuit that will reduce the number of guarantees you have configured.

- - - - - - -

Entering Menu: Number of Circuits

- - - - - - -

1) Maximum number of circuits allotted to this port (0...*1...3)
2) Number of circuits to guaranteed for this port (0...*1...1)

When the number of virtual circuits guaranteed for a port is less than the number of circuits allotted to a port, the port can establish more than the guaranteed number of circuits, but only if the extra circuits are available. For this reason, you should always guarantee at least one circuit to each port so that it is not denied network access.

A circuit's availability depends on the number of circuits guaranteed to other ports and the number of circuits the board supports. The type of I/O module and the command interpreter you select affect this number. Table 6-1 summarizes this relationship.

Configure will warn you with the following error message if you configure more guaranteed circuits on a board than the board can support:

This board can only support X guaranteed circuits. You have configured Y. You must lower some of your guarantees.

For example, suppose a Six-Serial Port I/O Module with an extended command interpreter supports eight virtual circuits, and you are connecting a terminal to each port. You guarantee one virtual circuit to each terminal. When you do so, there are only two circuits left on the board. Let's say that you allotted and guaranteed an extra circuit to CRT1, and an extra circuit to CRT2. You cannot guarantee an extra circuit to CRT3, since your guarantees have used up all the circuits available. If you allotted CRT3 an extra circuit and if CRT1 and CRT2 were not

Table 6-1.
Number of Circuits Supported

I/O Module	*Command Interpreter*	*Number of Circuits*
Serial/Parallel	Standard	12
	Extended	6
Six Serial	Standard	12
	Extended	8

using their circuits, CRT3 would still not be able to use the second allotted connection because it would reduce the guarantees you configured.

Circuit Deactivation. Asynchronous devices configured for initial, demanded, or session-oriented virtual circuits can be configured for circuit deactivation. Devices configured for permanent virtual circuits can deactivate the circuit, but the Entryway will immediately attempt to reestablish the connection. You would not, therefore, configure this device for virtual circuit deactivation.

Asynchronous devices deactivate virtual circuits by entering a string of characters called a Disconnect sequence that signals the Entryway to break the active circuit between the issuing device and another device. When a device is configured for multiple virtual circuits, it can enter a string of characters, called a Hold sequence to signal the Entryway to put the active circuit on hold. (Refer to Chapter 3). ■ **Note:** Synchronous devices deactivate virtual circuits in a very different manner. Refer to the section titled Virtual Circuits for Synchronous Devices later in this chapter for a discussion of this.

— — — — — —
Entering Menu: Circuit Deactivation
— — — — — —

1) Disconnect sequence (〈ESC〉OP)
2) Hold sequence (〈NUL〉)
3) Break signal should cause:
a) Disconnect
b) Hold
c) Neither

The Codex LAN default Disconnect sequence is (<ESC>OP). You would enter it on your keyboard as follows:

1) Press the Escape key or Ctrl-[.
2) Press the Shift key and o.
3) Press the Shift key and p.

This Disconnect sequence was chosen because it is not likely to appear in the data stream. You may choose another Disconnect sequence, perhaps to implement the standard control character of your terminals. You can configure different Disconnect sequences for different terminals.

The default Hold sequence is a <NUL>. If you are configuring a device that does not support multiple virtual circuits, you would accept the ASCII <NUL> for your Hold sequence. If you are configuring a device which disconnects a circuit by using the Break key, you would enter <NUL> for both the Disconnect and Hold sequences.

The following rules apply when you define either a Disconnect or Hold sequence.

- You can define a string of up to four characters for the Disconnect and Hold sequence. You can use printing or ASCII nonprinting characters. Refer to Table 5-4 in Chapter 5 for a list of permitted ASCII characters. Do not use a <NUL>.
- Neither sequence can include a character that is its own first character or the other sequence's first character. For example, "ABLE" is legal, but "AABL" is illegal. You can begin the Hold sequence with "A". A disconnect sequence of "ABLE" and a Hold sequence of "ATOM" is legal. A Hold sequence of "BOOK", however, makes the Disconnect sequence illegal because "ABLE" includes the first letter of the Hold sequence: "B". Selecting "L" or "E" as the first letter of the Hold sequence would also make the Disconnect sequence invalid.

- The Disconnect and Hold sequences must not be identical. When you configure identical sequences, Configure displays the following message:

  Hold and disconnect sequences are identical, hold is eliminated.

Configure stores the sequence as the Disconnect sequence and eliminates the Hold sequence.

To change the current value of the Disconnect or Hold sequence, enter the new value next to the Configure menu question. To enter a nonprinting ASCII character, type it in uppercase enclosed within angle brackets. For example, to set the Disconnect sequence to the sequence of characters generated by the F1 key on a Beehive terminal (STX, ESC, p, CR), enter it this way:

〈STX〉〈ESC〉p〈CR〉

To set the Disconnect sequence to the uppercase characters A, B, L, E, enter it this way:

ABLE

If you do not configure a Disconnect or Hold sequence and you do configure a device for virtual circuit deactivation, the only other method of circuit deactivation is through the use of the Break key. You can configure the Break key signal as a Hold or Disconnect sequence only if you have configured the device you are attaching to this port (see the section titled Asynchronous Device Characteristics in Chapter 5) to recognize the signal.

If you configure the Break signal as a Hold sequence, you must also configure a Disconnect sequence, using the method just described.

Virtual Circuits for Synchronous Devices

Synchronous devices may be interconnected by the third-party administrative Interconnect command or they may initiate virtual circuits. There are three ways in which synchronous devices can initiate virtual circuits as shown in the following menu:

```
- - - - - -
Entering Menu: Circuit Initiation
               (SCS-II, SCS-III)
- - - - - -
      1) Circuit initiation mode:
      a) Initial circuit
      b) Permanent circuit
      c) Demanded circuit
```

Bound Circuits. You configure bound circuits for synchronous devices in a similar manner to asynchronous devices. Refer to the section titled Virtual Circuits for Asynchronous Devices earlier in this chapter for a discussion of configuring bound circuits, and naming the connection resource name at the other end of the circuit.

Session-Oriented Circuits. Some bisynchronous (BSC) devices can be configured with a limited command interpreter.

```
- - - - - -
Entering Menu: Circuit Initiation (SCS-I)
- - - - - -
1) Circuit initiation mode:
      a) Initial circuit
      b) Permanent circuit
      c) Demanded circuit
      d) Command interpreter
```

Remote Job Entry devices enter a connection resource name into a job stream and transmit it to the Entryway to initiate a virtual circuit. Some BSC devices do not support the BSC dialect that the SCS-I software uses to communicate with the command interpreter, and cannot be configured to initiate circuits on command. Refer to Chapter 7 for a description of the BSC dialogue for session-oriented virtual circuits.

The Connection Resource Name must be preceded and followed by a unique character that does not occur within the name. These delimiters allow the BSC software to distinguish the name from the data contained in a record.

```
– – – – – –
Entering Menu: Command Interpreter Characteristics
– – – – – –
1) The connection resource name is specified as:
     a) First data record
     b) Separate record

2) What character delimits the connection resource name (%)
```

The percent sign (%) is the default delimiter. You can select a different character if you like, but the character you choose must correspond to an ASCII character with a value greater than 20 (hex), and cannot be a character that occurs in the connection resource name. ASCII control characters are not allowed. When the transmission code used is EBCDIC, the ASCII value translates into the corresponding EBCDIC value.

You also have to choose the method the device will use to transmit the connection resource name. You can configure the device to send it in the first data record or to send it in a separate record.

No matter which method you choose, the first block a device sends to its Entryway must contain at least the connection resource name so that the Codex LAN SCS-I software can create the desired virtual circuit.

The default choice is to send the connection resource name as part of the first data record. This choice is appropriate if your installation normally sends a beginning control command, such as a sign-on or log-on record, before sending data. When you transmit the connection resource name as part of the first record, the BSC software searches the record for the delimiters, reads the connection resource name, and creates the virtual circuit connection to the named device. The BSC software deletes the connection resource name and its delimiters from the record before sending the record to the destination device.

For example, suppose the normal sign-on record is as follows:

/*SIGNON REMOTE23 PASSWORD %HOSTA%

The BSC software finds the desired destination name, HOSTA, bracketed by the % character, creates the requested virtual circuit, and then forwards the sign-on record to the host, with the characters %HOSTA% deleted from the record.

When the BSC software removes the connection resource name from the record, it deletes the space it occupied. The absence of this space can cause problems when a field or fields that follow this space must be in a numbered position within a record. It is safest to add the connection resource name to the end of the record, but you can insert it somewhere else within the record as long as no position-sensitive fields follow it.

If your system does not normally send a beginning control command, such as a sign-on or log-on record, it may be difficult or impossible to include the connection resource name within the first data record. When this is the case, the second connection option is appropriate. With this option, the name of the desired destination device is sent to the Entryway as a separate file, preceding the user's data file. The destination device never sees this file.

For example, assuming the character % is the delimiter, the connection resource name used above would look as follows when sent as a single record:

%HOSTA%

The BSC software finds the desired destination name, HOSTA, bracketed by the % character and creates the requested virtual circuit.

The BSC software expects the initiating device to terminate a single record transmission with EOT. After ending the transmission with the BSC software, the initiating device can communicate normally with the destination device over the virtual circuit.

Circuit Deactivation. The Configure program will ask you if the device you are configuring can deactivate virtual circuits. If you answer Y for Yes, you will be asked to select from one of two methods of circuit deactivation, as shown in the following menu:

```
- - - - - -
Entering Menu: Circuit Deactivation
- - - - - -
1) Circuit timeout value (*0)?
2) Disconnect after DLE - EOT (Y or N)?
```

- If you enter "n" seconds next to "Circuit timeout value," the connection between two devices is broken after "n" seconds pass with no data transmission activity. "n" is any decimal number of seconds between 0 and 1092. A timeout value of "O" disables the timeout feature, preventing the Entryway from disconnecting two devices after circuit inactivity.
- If you enter "Y" to the second Configure question, a BSC device can deactivate a circuit with another BSC device by using DLE EOT as a Disconnect sequence. Both BSC devices must be configured to recognize this. This is not a Configure choice for devices supported by the SCS II and SCS III software.

Two other methods of circuit deactivation are always available, although you do not have to configure them:

- When two devices are configured for switched line support, they are disconnected whenever the ready signal goes false. The ready signal is DTR for a terminal or host, DSR for a modem.
- A third party, configured for the extended command interpreter, can issue a Disconnect command.

NETWORK CONFIGURATION

Each local network connected by a bridge to another network must be given a unique network ID.

```
– – – – – –
Entering Menu: Network ID
– – – – – –
1) Enter network ID (*0)?
```

If your LAN is composed of several bridged networks, called an Internet, you will have a LAN Manager for each network segment. The Network Administrator must configure a network ID from the LAN Manager console connected to that network segment.

To define a network ID, enter a decimal number in the range of 1 to 32767. Ensure that no two networks are configured with the same network ID. ■ **Note:** Prior to configuring a bridge (described next), you must reset every Entryway in the network and then invoke the Download Server. This downloads a network ID configuration file to each application processor.

BRIDGE CONFIGURATION

There are two types of bridges: Local and Remote. A Local Bridge links two networks not geographically separate. For example, two network segments within the same building can be connected by a Local Bridge. A Remote Bridge links two geographically separate networks via a remote communications link such as a dial-up or leased line.

Local Bridges

A Local Bridge is a specially-equipped 4001 Entryway which connects two local networks. This Entryway has two processor boards, each part of a dedicated board set. One set, the lower set pictured in Figure 6-1, consists of a Transmitter Interface Board (TIB), a Receiver Interface Board (RIB), and a processor board installed in the Board A slot of the Entryway. The other set, the upper set pictured, includes another Transmitter Interface Board (TIB), a Receiver Interface Board (RIB), and a processor board installed in the Board D slot of the Entryway. One network is connected to the lower RIB, and the other network is connected to the upper RIB.

The Local Bridge software is downloaded to processor Board A which in turn, sends Board D its software load via the internal databus. The network connected to the lower RIB, associated with Board A, is responsible for the bridge configuration files. The LAN Manager on that network segment, Network ID #1 in Figure 6-1, is used to configure the bridge. ■ **Note:** Prior to configuring a bridge, the Network Administrator must have configured a network ID from the LAN Manager on each network segment connected by the bridge.

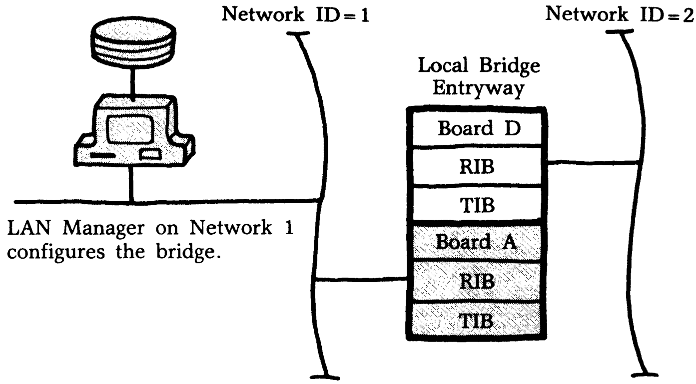


Figure 6-1. Locally Bridged Networks

To configure a Local Bridge:

- Enter the serial number of the Entryway serving as the bridge.
- Enter the network ID for the network on the *other* side of the bridge.

```
– – – – – –
Entering Menu: Local Bridge
– – – – – –
1) What is the network id for the far side (*0)?
```

Remote Bridges

Figure 6-2 illustrates a Remote Bridge (two 4002 Entryways) connecting two geographically-separate networks via a phone line.

Each Entryway is equipped with one processor board with a High-Speed Serial Port I/O module. This I/O module has four ports: an RS-232 DCE port, an RS-232 DTE port, a V.35, and an RS-449 port. You would connect a modem to one of these ports.

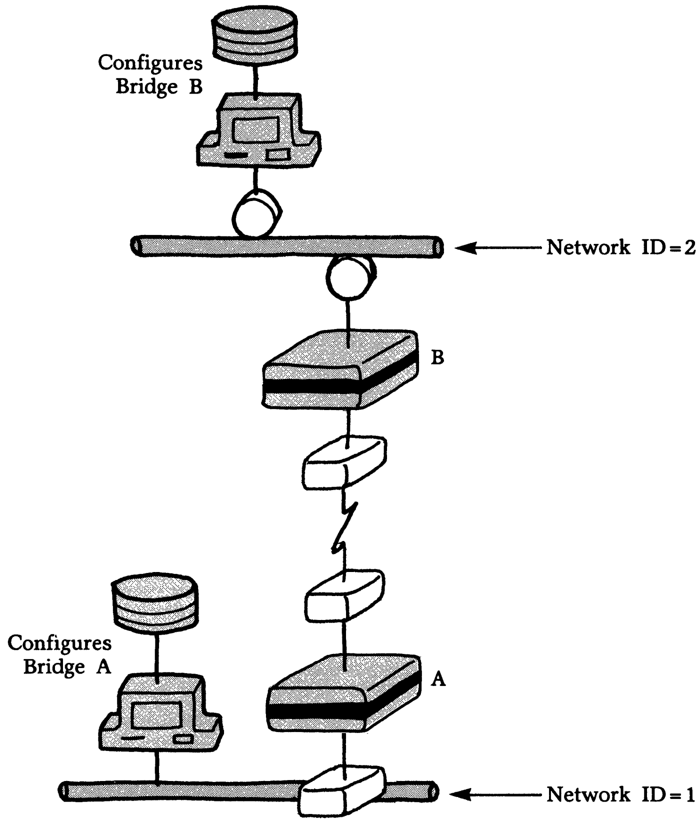


Figure 6-2. Remote Bridge Between Two Networks

The Remote Bridge software is downloaded from the LAN Manager connected to each local network segment. The Network Administrator for each network segment configures that network's bridge Entryway.

— — — — — —

Entering Menu: Remote Bridge

— — — — — —

1) Which EWY model:
- **a) 4001**
- **b) 4002**

2) Which communications interface:
- **a) V35**
- **b) RS232**
- **c) RS449**

When you configure a Remote Bridge, Configure will ask you for the following information:

- the Entryway serial number,
- the type of bridge (local/remote),
- the Entryway model,
- and the active port.

Chapter 7
Bisynchronous Dialogue for Session-Oriented Virtual Circuits

Contents

OVERVIEW

Only devices configured as command devices can establish session-oriented virtual circuits. Certain devices do not support some of the bisynchronous (BSC) features required for a command device. This section explains the BSC dialect that the BSC software expects a command device to use. This dialect is typical of batch-oriented BSC devices such as the IBM 2780 and IBM 3780.

BSC DIALOGUE FOR CREATING SESSION-ORIENTED VIRTUAL CIRCUITS

This section presents the sequence of BSC control characters and text blocks corresponding to each of the methods of transmitting the connection resource name. Note that the dialogue that occurs between the Codex LAN BSC software and a device is usually transparent to the user.

The following explanations are presented to help a Network Administrator understand how the software works and to choose the method that is appropriate for the devices used.

When you configure a device as a command device, Configure asks whether the connection resource name will be part of the first data record sent to the destination device, or a separate file or transaction. Figure 7-1 illustrates the dialogue that occurs when the connection resource name is sent as part of the first data record. Figure 7-2 illustrates the dialogue that occurs when the connection resource name is sent as a separate file or transaction.

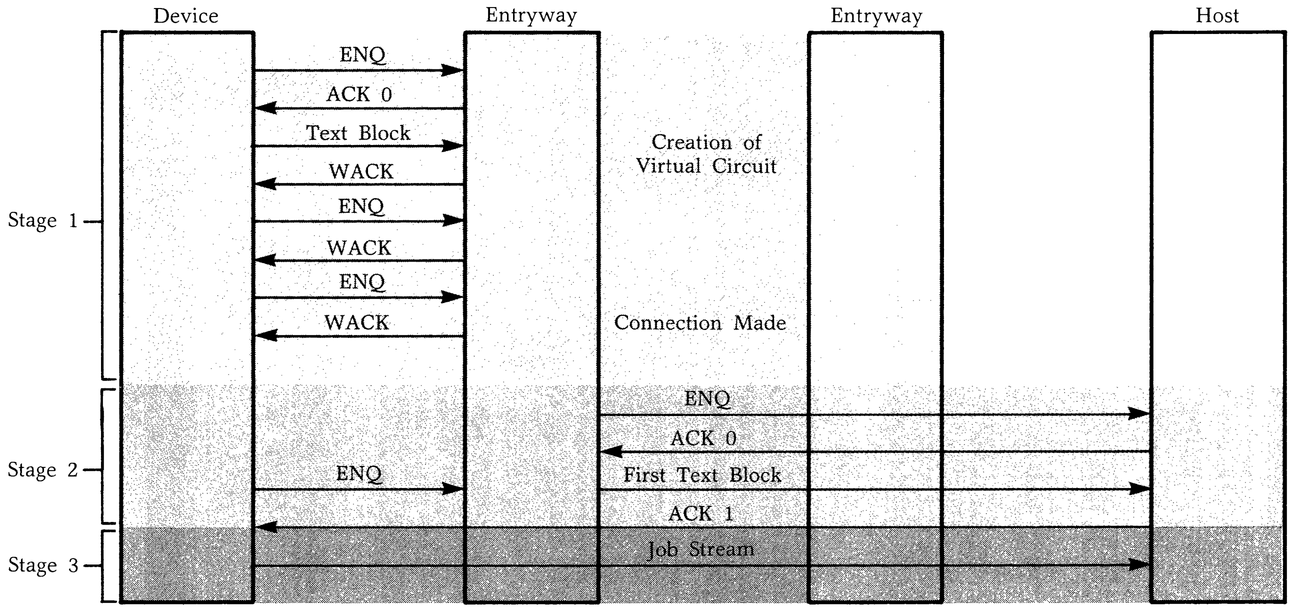


Figure 7-1. Dialogue for First Record Option

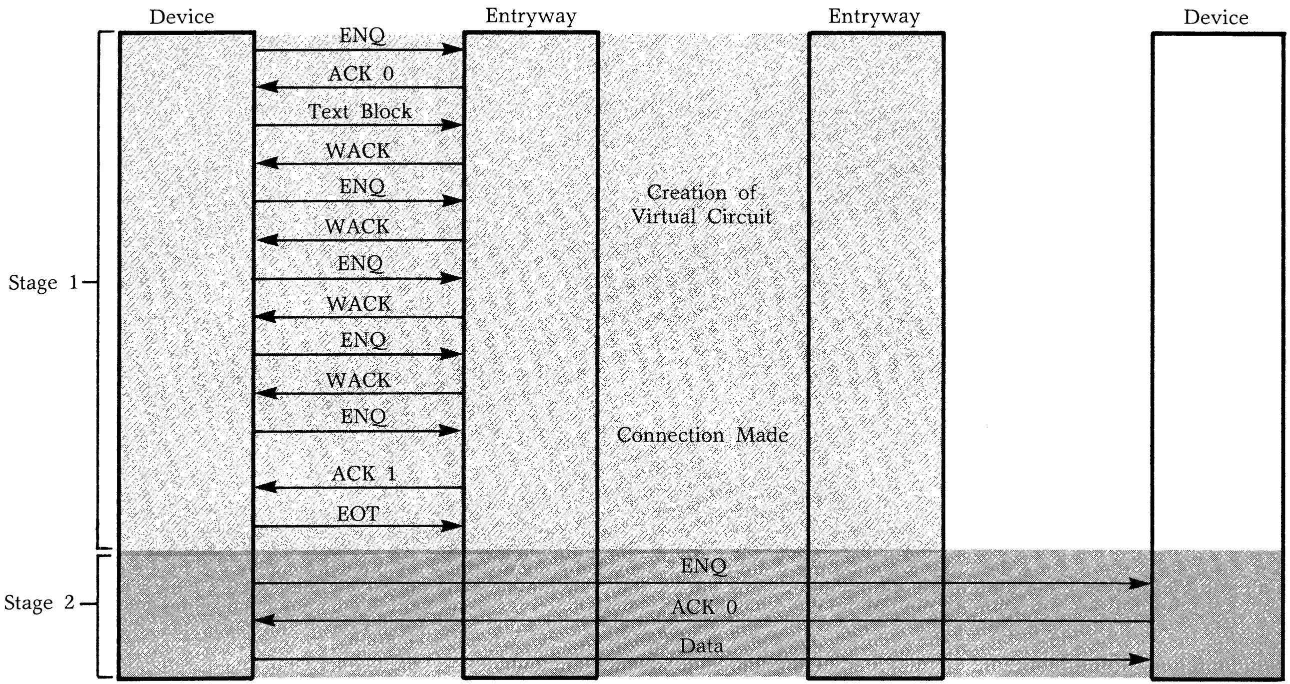


Figure 7-2. Dialogue for Separate Record Option

Figure 7-1 shows a remote device sending a job stream to a host. The Stage 1 portion of the figure shows the dialogue that takes place between the device and the BSC software running in the Entryway to which the device is attached.

During this dialogue, the device bids for the line (ENQ), and sends its sign-on or log-on image with the embedded connection resource name. The BSC software acknowledges the successful transmission of the first block (ACK 0), extracts the connection resource name from it, and creates the virtual circuit connection between the device and the named host.

Until the virtual circuit connection is made, the BSC software will not accept more than one block from the device. The Codex LAN BSC software responds to the first block with WACK to prevent the device from sending any more data. WACK acknowledges the successful transmission of the block, but asks that the device wait before transmitting any more data. The device sends ENQ to request permission to send the next block. The BSC software responds with WACK to each such ENQ the device sends until the virtual circuit connection is made.

In Stage 2 of the dialogue, the BSC software running in the Entryway attached to the device establishes the virtual circuit connection between the device and the host.

The ENQ and BSC software sends a line bid. In fact, it is the original line bid sent by the device. The host responds with ACK 0, accepting the bid. The BSC software than sends the host the first block it received from the device. The block does not contain the connection resource name when the BSC software sends it to the destination device.

Once the device receives ACK 1 from the host, it can begin sending the rest of the data. This direct communication is the third and final stage of the BSC dialogue.

Figure 7-2 shows a device sending data to another device. The sending device sends the connection resource name as a separate transaction. The Stage 1 portion of the figure shows the dialogue that takes place between the device and the BSC software running in the Entryway to which the device is attached.

During this dialogue, the device bids for the line (ENQ), and sends the connection resource name as a separate file. The BSC software acknowledges the successful transmission of the first block (ACK 0), and creates the virtual circuit connection between the requesting device and the named device.

The BSC software will not accept more than one block from the device until the virtual circuit connection is made. The Codex LAN BSC software responds with WACK to the first block to prevent the device from sending any more text blocks. WACK acknowledges the successful transmission of the block, but asks that the device wait before transmitting any more information. The device responds with ENQ to request permission to send the next block. The BSC software responds with WACK to each such ENQ the device sends until the virtual circuit connection is made. ■ **Note:** When the connection resource name is sent as a separate record, that record should contain only the connection resource name. The record should not contain any data meant for the destination device because the destination device never receives this record.

Once the virtual circuit connection is made, the BSC software sends ACK 1 to the device. The user device must then send EOT to terminate the session between the device and the BSC software.

The two devices can now communicate directly with one another over the virtual circuit connection. But before the initiating device can begin to send data, it must rebid the line and wait for ACK 0. The Stage 2 portion of Figure 7-2 illustrates this second stage of the dialogue.

ERROR MESSAGES

When a device attempts to establish a session-oriented virtual circuit, the BSC software sends an error message to the device if either a connection error or transmission error occurs. In order to receive these messages, a device must be able to respond to RVI (Reverse Interrupt).

RVI is a positive acknowledgement of the last text block sent. RVI also asks the sending device to send EOT, ending that dialogue and relinquishing control of the line, so that the current receiving data can send a message. If a device does not recognize the RVI sequence and relinquish control of the line, it cannot receive error messages.

Once the sending device relinquishes the line, the BSC software bids for the line. After the device accepts the line bid by sending ACK 0, the BSC software sends a text block containing the error code. The format of this block is as follows:

STXxxxETX

where xxx stands for the error code. The length of the text string between STX and ETX varies from one to three characters depending on the number of digits in the error code.

The BSC software transmits the error message using the transmission code chosen for the device in Configure. The BSC software relinquishes control of the line by sending a DLE EOT sequence once it receives an ACK 1 from the device.

A command device receives error messages that pertain to BSC connections and transmission. Table 7-1 defines the meanings of each of the error codes.

Table 7-1.
BSC Error Codes

Number	*Meaning*
0	**Failure.** The operaton did not complete successfully. The most common cause of failure is a timeout in the network software. Try again.
3	**Bad parameter.** A bad parameter value, such as an illegal version number or an out-of-bound value, was passed internally to the network. This is a programming error. If this occurs, contact your Codex representative.
9	**Timeout.** The remote Entryway failed to respond to a network message within the timeout limit (about 10 seconds). This condition results when the processor is temporarily overloaded or when the Entryway is short of memory space. This condition can occur any time Entryways try to communicate. Try again.

(continued on next page)

Table 7-1 (continued).
BSC Error Codes

Number	*Meaning*
11	**Network resource not available.** The named device or service is currently busy and cannot accept a connection. Codex LAN returns this error code only when a unique name is specified. Try again later.
58	**Local connection descriptors exhausted.** The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
65	**Local buffers exhausted.** The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
69	**Remote buffers exhausted.** The remote Entryway has no memory space available in which to perform the operation. No corrective action is possible.
100	**No ENQ from the connecting device.** The BSC software expected an ENQ sequence from an attached device, but received another control character, text, or meaningless characters.
101	**No delimiter in the first block.** The block that should contain the connection resource name did not contain the defined delimiter.
102	**No end delimiter in the first block.** The BSC software found only one delimiter in the block.
103	**Connection name longer than 30 characters.** The text of the connection resource name was longer than the 30-character limit.
104	**No characters between delimiters.** The BSC software found two delimiter characters in the block, but there were no characters between the two delimiters.
105	**First block has a block BCC/LRC check (15 retries).** The BSC software attempted to receive a block 15 times. Each time the BCC/LRC of the first block was bad.

Table 7-1 (continued).
BSC Error Codes

Number	*Meaning*
106	**Destination device did not respond with ACK 0 to the line bid.** This error can occur when the first record option is chosen. After the virtual circuit connection is made, the destination device must respond with ACK 0. The initiating device receives this error message if the destination device does not send the ACK 0 sequence.
107	**Device using the separate record option did not send EOT.** After receiving the positive acknowledgement for the connection resource name, the initiating device should send EOT to the BSC software to end the dialogue between them.
108	**Command device's ENQ sequence is larger than 15 characters.** The line bid sequence is longer than the 15-character maximum.
109	**First block was invalid.** The first block, which should contain the connection resource name, was not a valid text block, and did not begin with valid BSC control characters.
150	**Local device data structures exhausted.** The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
151	**No such resource.** Either the specified resource is inactive or it is not configured.
152	**Local process descriptors exhausted.** The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
199	**No memory.** The internal Entryway memory resources that communicate with the user programmable board are not available. No corrective action is possible.
201	**No response for specified name.** The specified name cannot be found on the network. The Entryway with that resource may be powered down or disconnected. The Entryway also may be temporarily overloaded and unable to respond.

(continued on next page)

Table 7-1 (continued).
BSC Error Codes

Number	*Meaning*
202	**All ports busy.** All ports or services associated with the name specified are busy. This code is returned only when the specified resource is not a unique name. Try again.
203	**No response from address.** No Entryway responds to the specified address.

Chapter 8
Using the Configure Program

Contents

BEFORE YOU START

Ensure that NOSS has been installed on the fixed disk drive of the LAN Manager you are using to invoke Configure. Software installation instructions are included in Chapter 2. You can invoke Configure from the LAN Manager console or from any remote terminal connected to it. Your DOS prompt should be the letter of the fixed disk drive, followed by a colon. For example:

```
C:
```

INVOKING CONFIGURE

1) Ensure that your directory is \ codex. If it is not, enter at the DOS prompt:

   cd\ codex [CR]

   This prompt displays:

   ```
   C:\ codex
   ```

2) Type:

   configure [CR]

   This message is displayed:

   ```
   CONFIGURE
   Type ? for HELP
   ```

SELECTING OPTIONS FROM THE TOP MENU

Configure's menus are organized into a tree structure, similar to the structure illustrated in Figure 8-1.

When you configure an Entryway, device, or bridge, Configure expects you to add, modify, or remove a configuration. To initiate Configure's questions, select one of these options and supply the program with a serial number for an Entryway or bridge and a Device Definition filename for a device.

When you configure a network ID, Configure expects you to supply a decimal network number.

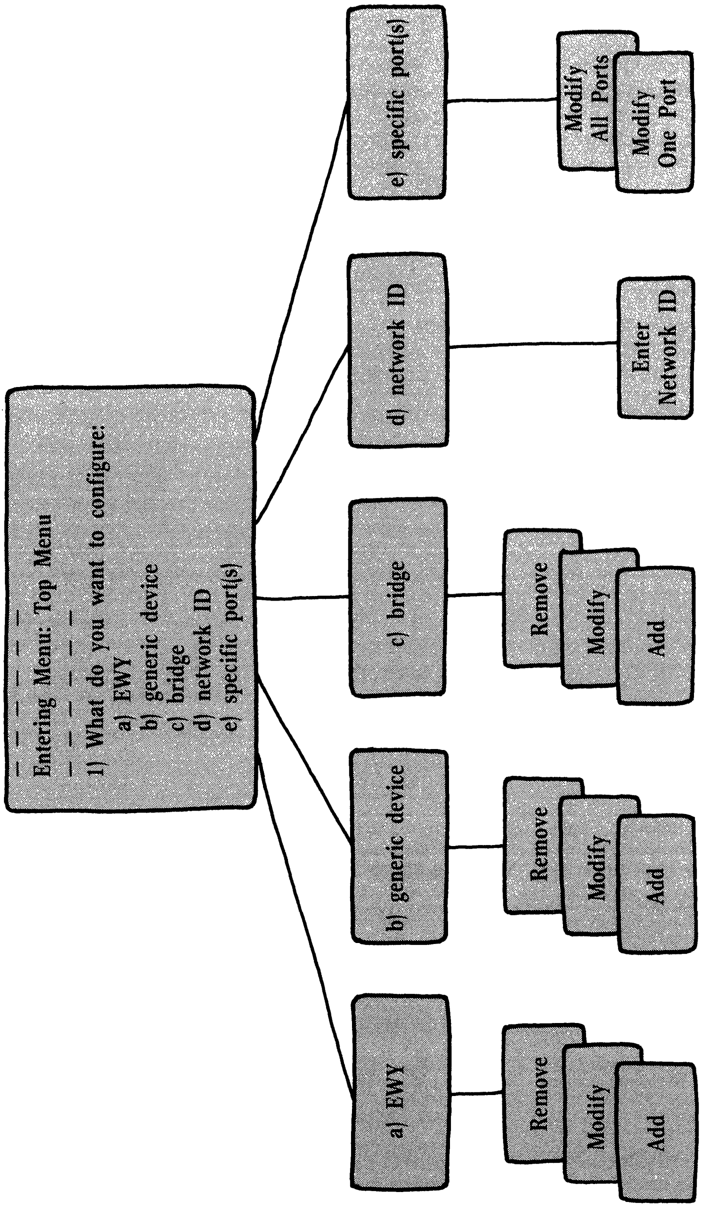


Figure 8-1. Configure Menu Structure

When you configure a specific port, Configure expects you to modify one or all port descriptions in an existing Entryway configuration file. To initiate Configure's questions, select one of these options and then enter the Entryway serial number, and letter of the board (see the section in Chapter 3 titled Device and Service Addresses).

Adding a Configuration

Select this option when you add a new Entryway to the LAN, when you connect a new device to an Entryway for which there is no generic Device Definition file, or when you add a bridge to the network.

Modifying a Configuration

Select this option when you have already configured a device or Entryway and you wish to change some of your answers to Configure's questions. The menu will display your previous choices by prefacing them with an asterisk or enclosing them within parentheses. For example, if you previously configured a device to *not* echo Entryway responses, Configure's question would look like this:

Should Entryway commands be echoed (Y or *N)?

If you configured a device with a ⟨DLE⟩ command mode initiation character, this question would be displayed:

Initiation Character (⟨DLE⟩)?

Removing a Configuration

Select this option when you wish to remove a Device Definition file, Load Characteristics, or Load Specification file from the fixed disk of the LAN Manager.

Configuring a Specific Port

Select this option when you wish to modify port characteristics of a previously configured Entryway. Do not select this option when the characteristics you wish to modify are board characteristics (see Chapter 6). Your previously configured answers are higlighted with an asterisk or enclosed within parentheses.

RESPONDING TO MENU QUESTIONS

Most of Configure's questions list several options from which you can choose. For example, this question asks you to choose a) or b):

```
RS-232 Signaling used by this device:
     a) Standard
     b) Extended RS-232
?
```

To respond, enter the letter of your choice next to the question mark, then press your terminal's Return or Enter key. You can type your responses in upper or lowercase characters.

As you move to each menu question, the cursor is displayed at the space following the question mark.

Other questions ask you for a yes or no response. For example:

```
Modify the command interpreter characteristics (Y or N)?
```

The appropriate answer is to enter Y or N, followed by a Return or Enter. In this example, the format of the correct response is enclosed within parentheses (Y or N).

Other questions ask you to enter a short word or a number, as in the following:

```
Input Buffer Size (6...64...254)
```

You would respond to this question by entering the number which corresponds to your choice, followed by a Return or Enter. The range of acceptable responses is enclosed within parentheses. If you select an answer that is not within this range, this message displays:

```
Invalid option.
```

You are then prompted to supply another response.

MENU DEFAULTS

Configure provides you with default answers to its questions even if you are configuring a new device or Entryway. The asterisked N is the default choice in the following menu:

```
- - - - - -
Entering Menu: Entryway (Device) Defaults
- - - - - -
1) Similar Entryway (Device) (Y or *N)?
```

To accept the default, press the Return or Enter key. To change it, type Y, followed by a Carriage Return.

If you tell Configure that a configuration file exists for a similar Entryway or device, the menu defaults are the answers you provided in that configuration session. If a similar file does not exist, the menu defaults are based on default configuration files provided for you with the NOSS software. ■ **Note:** When you tell Configure there is an Entryway similar to the one you are configuring, that Entryway's parameters will display as defaults but not the parameters of devices connected to the Entryway. Configure will ask you later in the program to supply the name of a Device Definition file.

CONFIGURE COMMANDS

Each Configure menu has a title which consists of the name of the menu and the modifier: "Returning To Menu" or "Entering Menu". This allows you to locate yourself more easily in the complex menu structure.

When you press the Enter key after answering a Configure question, the next menu which displays and all subsequent menus are labeled "Entering Menu." There is, however, a Configure command which allows you to back up one question or one menu at a time: the Backspace command. Each menu you return to after you press your Backspace key is then labelled "Returning to Menu."

In addition to the Backspace command, Configure provides other commands that allow you to control the menu displays. At any time during a session, you can type a question mark (?) to display the list of available commands. Table 8-1 lists these commands and the format in which you enter them.

You do not have to press a Return or Enter when you use these commands. They must, however, be the first character typed in response to a menu question. For example, you cannot type part of a response, backspace over it, and then enter Ctrl-R.

Table 8-1.
Configure Commands

Keystrokes	*Action*
?	Question Mark. Displays a list of Configure commands and the ASCII equivalents of the keyboard characters you use to invoke them.
⟨BS⟩	Use of the Backspace key, or Ctrl-H if your terminal does not have a Backspace key, allows you to back up to the previous question in the current menu. If there are no previous questions in this menu, it displays the previous menu.
∧	Caret. Accepts the indicated default answers to the current menu questions, then returns you to the previous menu.
Ctrl-R	Lists all the questions in the current menu.
Ctrl-C	Aborts the Configure program and returns you to PC-DOS.
@	At sign. Allows you to randomly select a question by number within the current menu.
⟨CR⟩	Return or Enter. Accepts the menu default value and displays the next menu question. Entered after your answer to a question, Return or Enter executes your choice.

SPECIAL CHARACTERS

The backslash character can be used as a Configure Escape Character. When entered as a prefix to a Configure command, it negates its special-character functionality. For example, let's suppose you want to supply Configure with the name "@ACT" as a connection resource name. Since "@" is a Configure command, you must tell Configure to ignore its command function and quote it. You would enter it this way:

\ @ACT

You could use the backslash as the first character of a name such as "\ ACT", if you prefix it with another backslash:

\ \ ACT

The left-angle bracket (⟨) tells Configure to treat the following string of characters up to a closing right-angle bracket as a non-printable ASCII character. For example, enter ⟨HT⟩ to supply Configure with the ASCII equivalent of the Tab character. Refer to the section titled Device Command Mode Definition in Chapter 5 for a discussion of ASCII characters.

EXITING FROM CONFIGURE

After you answer all of Configure's questions about the device, port, application processor board, or Entryway you are configuring, a question similar to the one below is displayed:

Configuration completed (Y or *N) ?

If you enter Y and a Return or Enter, Configure displays a message telling you what device you have configured:

File Written: DTERML

Or, if you have finished configuring a board:

File Written: LC1776A

Or, if you have finished configuring an Entryway:

File Written: LS1776.

Refer to Chapter 4 for a discussion of the files Configure creates. In the above sample messages, the first file is a Device Definition file for the device "TERML." The second file is a Load Characteristics file for Board A of an Entryway with the serial number 1776. The third file is the Load Specification file for Entryway #1776.

When you exit from Configure after modifying an existing configuration, Configure displays one of the messages listed below:

File already exists: DTERML

Or,

File already exists: LC1776A

Or,

File already exists: LS1776.

Configure asks you to select one of the following actions:

```
Select an option:
    a) Overwrite existing file
    b) Rename existing file
    c) Abandon writing of new file
```

This menu provides no defaults. You must respond with the letter that corresponds to the desired action. When you select a), Configure overwrites the existing file with a new file based on the current configuration. When you select b), Configure saves the existing file under a new filename. When you select c), your present configuration is aborted, and the existing file is left unchanged.

FILE CONSISTENCY CHECKS

Before writing a configuration file, Configure performs a consistency check on the answers you supply. The purpose of this check is to verify that your choices do not contradict each other. An example of an inconsistent choice would be to assign a command interpreter to a device that is not allowed to initiate virtual circuits.

Configure will display an error message for inconsistent configurations. Some messages display interactively during the configuration session. Other messages display only when you exit and the file writing cycle begins. This type of message looks like this:

```
This menu is inconsistent. Do you want to proceed anyway?
```

When you answer Y, the file is written despite the inconsistency. When you answer N, Configure returns you to the first menu in the particular group of menus where the problem was found. You can then correct the error, and exit the program.

Chapter 9
Tutorial

This chapter includes a representative configuration which may be of assistance in preparing yourself for your first Configuration session. Answers which you must provide to the program appear in blue. This session configures Port 1 of Application Processor Board A in a 4001 Entryway. An asynchronous video terminal is connected to this port. The port is given a standard command interpreter and can initiate and deactivate multiple virtual circuits.

```
A:\ codex configure

CONFIGURE
Type ? for help.



– – – – – –
Entering Menu: Top Menu
– – – – – –
1) What do you want to configure:
     *a) Entryway
      b) generic device
      c) bridge
      d) network ID
      e) specific port(s)
? a
Entryway number: 1776



– – – – – –
Entering Menu: Action
– – – – – –
1) Select an action:
     *a) Add new Entryway
      b) Modify existing Entryway
      c) Remove existing Entryway
? a
```

```
- - - - - -
Entering Menu: Entryway defaults
- - - - - -
1) Similar Entryway (Y or *N)? n
```

■ **Note:** Since a similar Entryway was not selected, the defaults which display in the following menus (marked with an asterisk) are Codex LAN system defaults.

```
- - - - - -
Entering Menu: Entryway model
- - - - - -
1) Select an Entryway model:
      *a) 4001
       b) 4002
? a
```

```
- - - - - -
Entering Menu: Per Board Questions
- - - - - -
1) Select a board:
      *a) A
       b) B
       c) C
       d) D
? a
```

```
– – – – – –
Entering Menu: Board Function
– – – – – –
1) What function will this board provide:
      a) Virtual Circuit Service for attached devices
      b) Datagram Service for attached devices
      c) Ethernet Data Link Service for attached devices

      d) Network Debugging Service
      e) File Transfer Service (SFT)
      f) Network Configuration Facility

      g) Supporting System Board in dual board environment
      h) User Programmable Board in dual board environment
      i) user written service in single board environment
      j) user defined load

     *k) none (ignore this board)
? a
```

```
– – – – – –
Entering Menu: Virtual Circuit Service
– – – – – –
1) Select an action:
     *a) Add new board
      b) Modify existing board
      c) Remove existing board
? a
```

```
– – – – – –
Entering Menu: Board defaults
– – – – – –
1) Similar board (Y or *N)? n
```

■ **Note:** Since a similar board was not selected, the defaults which display remain Codex LAN system defaults. If you answered Y to this question, the defaults would revert to those of the Entryway number and board that you supply to the program when prompted.

\- - - - - - -

Entering Menu: I/O Module

\- - - - - - -

1) Select an I/O module:
a) Serial/Parallel
***b) Six Serial**

? a

\- - - - - - -

Entering Menu: Serial/Parallel I/O Module

\- - - - - - -

1) Should initial diagnostics be run (Y or *N)? n

■ **Note:** Internal diagnostics have not been selected because LAN Entryways automatically perform diagnostics when powered on or reset.

2) Select a driver:
***a) Serial asynchronous**
b) Serial asynchronous and 8-bit parallel
c) SCS-I (BSC)
d) SCS-II (HDLC/SDLC)
e) SCS-III (DDCMP)
f) User written driver

? a

■ **Note:** The serial asynchronous driver was selected because parallel devices will not be connected to this board.

```
- - - - - -
Entering Menu: Asynchronous Driver
- - - - - -
1) Select a data rate grouping:
      a) 37.5, 75, 150
      b) 75, 150, 300
      c) 110
      d) 150, 300, 600
      e) 300, 600, 1200
      f) 600, 1200, 2400
      g) 1200, 2400, 4800
      h) 2400, 4800, 9600
     *i) 4800, 9600, 19200
      j) 9600, 19200, 38400
? i
```

■ **Note:** The device which is to be connected to this port has a data rate of 9600 bps.

```
2) Select a command interpreter:
     *a) Standard
      b) Extended
      c) None
? a
```

```
- - - - - -
Entering Menu: Multiple Virtual Circuits
- - - - - -
1) Will any port support more than one circuit (Y or *N)? y
```

```
- - - - - -
Entering Menu: Serial/Parallel Ports
- - - - - -
1) Select a port:
     *a) 1
      b) 2
      c) 3
      d) 4
      e) 5
      f) 6
? a
```

```
– – – – – –
Entering Menu: Serial Asynchronous Port On
               Serial/Parallel I/O Module
– – – – – –
1) Port is not currently configured.
     Select an action:
      a) Modify port configuration
      b) Remove port configuration
      c) Attach new device to port
     *d) Accept current port configuration
? c
Enter generic device name: CRT
```

■ **Note:** Entering CRT selects the Device Definition file DCRT, a default Device Definition file which is included in the NOSS software. The device characteristics defined in this file apply to serial asynchronous devices connected to the Serial/Parallel I/O Module. Another file, DCRT6, defines a serial asynchronous device connected to the Six-Serial Port I/O Module. This configuration assumes the characteristics of CRT are accurate.

```
– – – – – –
Entering Menu: Serial Asynchronous Port
– – – – – –
1) Modify generic device characteristics (*Y or N)? n
2) Will this device be able to initiate virtual
   circuits (*Y or N)? Ctrl-R
```

■ **Note:** Pressing Ctrl-R lists the contents of the current menu, as shown below. The program then repeats the current question and prompts you for a response.

```
- - - - - -
List of Menu: Serial Asychronous Port
- - - - - -

1) Modify generic device characteristics (*Y or N)
2) Will this device be able to initiate virtual circuits (*Y or N)
3) Will this device be able to deactivate virtual circuits (*Y or N)
4) Should circuit data be echoed locally (Y or *N)
5) Data rate for this port:
     a) 4800
     b) 9600
    *c) 19200

6) The following names exist for this port or service:
     a) 1776a1
   Delete any port names (Y or *N)
7) The following names exist for this port or service:
     a) 1776a1
   Add any ports names (Y or *N)

Current question is:
2) Will this device be able to initiate virtual circuits (*Y or N)? y



- - - - - -
Entering Menu: Circuit Initiation
- - - - - -
1) Method of circuit initiation:
     a) Initial circuit
     b) Permanent circuit
     c) Demanded circuit
    *d) Command interpreter
? d
```

■ **Note:** Since a standard command interpreter was previously selected for the board, selecting Command Interpreter-initiated circuits for this port will allocate to it the standard LAN command set.

\- - - - - -
Entering Menu: Command Interpreter
\- - - - - -

1) Will this port support more than one circuit (Y or *N)? y

■ **Note:** Since multiple virtual circuits were configured for the board, the port can be configured for multiple circuits.

\- - - - - -
Entering Menu: Number of Circuits
\- - - - - -

1) Maximum number of circuits allotted to this port (0...*1...3)? 2
2) Number of circuits to be guaranteed for this port (0...*1...2)? 1

■ **Note:** This port is guaranteed one circuit and, if the circuit is available, can initiate another circuit after the first circuit is placed on hold.

\- - - - - -
Returning to menu: Circuit Initiation
\- - - - - -

2) Will this device also be able to accept incoming connection requests (*Y or N)? y

■ **Note:** This menu is labeled Returning to Menu because it displays the second question of the Circuit Initiation menu displayed previously. That menu branched to two submenus before displaying its second question.

\- - - - - -
Returning to menu: Serial Asynchronous Port
\- - - - - -

3) Will this device be able to deactivate virtual circuits (*Y or N)? y

\- - - - - -

Entering menu: Circuit Deactivation

\- - - - - -

1) Disconnect sequence (⟨ESC⟩OP)? DC4
2) Hold sequence (⟨NUL⟩)? Backspace
1) Disconnect sequence (DC4)? ⟨DC4⟩

■ **Note:** Question 1 in the menu above displays the Codex LAN default Disconnect sequence: ⟨ESC⟩OP. The ASCII nonprinting character ⟨DC4⟩ (Ctrl-T) was selected to replace it. DC4 was typed next to the question, but the angle brackets were not typed in. This would result in a Disconnect sequence of the printing characters DC4. To correct the error, the Backspace character was typed and the angle brackets were then included.

2) Hold sequence (⟨NUL⟩)? ⟨SOH⟩
3) Break signal should cause:
  a) Disconnect
  b) Hold
  ***c) Neither**
? @
Randomly access question number: 2
2) Hold sequence (⟨SOH⟩)?
3) Break signal should cause:
  a) Disconnect
  b) Hold
  ***c) Neither**
? c

■ **Note:** The "at" sign (@) was typed to randomly access a question in the current menu. The program prompted you to supply a question number. The menu then returned to Question #2 as directed. Notice that the default displayed for the Hold sequence is the ASCII character previously chosen: ⟨SOH⟩ (Ctrl-A).

```
------
Returning to menu: Serial Asynchronous Port
------
4) Should circuit data be echoed locally (Y or *N)? n
5) Data rate for this port:
     a) 4800
     b) 9600
    *c) 19200
? b
```

■ **Note:** The port data rate grouping displayed is the group of three data rates that was chosen previously for the board.

```
6) The following names exist for this port or service:
     a) 1776a1
   Delete any port names (Y or *N)? n
7) The following names exist for this port or service:
     a) 1776a1
   Add any port names (Y or *N)? y


------
Entering menu: Add port name
------
1) The following names exist for this port or service:
     a) 1776a1
   Type name, or N: (N)? TOM'S CRT

Is this a rotored (non-unique) name? (Y or *N) n
1) The following names exist for this port or service:
     a) 1776a1
     b) TOM'S CRT
   Type name, or N: (N)? N


------
Returning to menu: Serial/Parallel Ports
------
2) Board configuration completed (Y or *N)? y
   File written: LC1776A
```

```
- - - - - - -
Returning to menu: Per Board Questions
- - - - - - -
2) Entryway configuration completed (Y or *N)? y
   File written: LS1776



- - - - - - -
Returning to menu: Top Menu
- - - - - - -
2) Configure completed (Y or *N)? y
   Exit CONFIGURE

A:\ codex
```

Chapter 10
File Transfer Service

Contents

OVERVIEW

The File Transfer Service (SFT for Simple File Transfer) provides a mechanism for transfer of user programs between host computers. It can transfer load modules to the network Download Server for subsequent loading into the Entryways. SFT is not used for direct loading of Entryways; the Download Server performs that function. SFT does the following:

- Transfers both ASCII and binary files
- Ensures reliability and provides flow control
- Transfers all control information and data for a particular host over a single virtual circuit port
- Maintains contact with the user's terminal throughout the session and reports progress or problems as they occur
- Implements as much as possible of the protocol and control logic in a Codex LAN service, permitting host programs to remain relatively simple
- Provides a single service with which each host can converse and through which it can converse with other hosts
- Accommodates any idiosyncrasies various hosts might have with respect to special input or output characters
- Avoids the necessity for any host-specific knowledge within the central SFT service
- Uses only ASCII characters in the protocol itself, which helps in the debugging of new host programs

Performance was subordinated to these goals. Other techniques, such as using the Datagram service, are more appropriate when performance is of primary importance, such as in large file transfers between computers.

IMPLEMENTATION

The heart of SFT is a service running on a processor board somewhere on the network. The user connects to this service (just as though you were connecting to any other device on the network), and the SFT service automatically creates the other network connections necessary to transfer the files.

During a typical transfer, three SFT components are active: the Codex LAN service and a program at each host. The service responds to user commands and directs transfers from the source to itself and from itself to the destination. The host programs implement a protocol that performs several of the tasks mentioned above (multiplexing of commands and data over a single port, reliability, flow control, binary transfers, and accommodation of host control sequences).

Once started, SFT has three nested command levels:

- A connection setup level that permits you to specify the participants in the transfer(s)
- A Host Command level that provides an opportunity to prepare both hosts for the transfer(s)
- A File Transfer level that performs the actual file transfer(s)

HARDWARE REQUIREMENTS

SFT requires its own processor board. No other devices or user services may be enabled on that board.

USING SFT

The following paragraphs are a step-by-step guide to using SFT.

Invoking SFT

If the SFT service has been installed (using Configure) on an Entryway somewhere in your network, you may invoke it by activating your command parser and typing:

C SFT

SFT Connection Level

Upon entry to and return from Connection level, SFT types:

From port

This prompt should be answered with the full network address of the source host, followed by ⟨CR⟩. SFT then displays the following on the same line:

To port

Answer this query in the same way. Upon completion of this command, SFT enters Host Command level.

To leave SFT and return to Codex LAN command level, enter Ctrl-D in response to the From prompt. If you are at a deeper level of SFT and wish to leave it, Ctrl-D can be used to move up a level. A series of Ctrl-Ds can be used to move up multiple levels. Typing Ctrl-D in response to the To prompt causes SFT to repeat the From prompt.

SFT Host Command Level

Upon entry to Host Command level, or upon return from File Transfer level, SFT types:

Enter commands for source host; terminate with Ctrl-D.

At this point, SFT places your terminal in direct communication with the source host. Anything that you type (except for Ctrl-D or Ctrl-V) is transmitted to the source host, and the host's response appears on your terminal. You can type anything you want at this point (e.g., you can log on or you can list your directory), as long as what you type is intelligible to the source host. The last sequence you type should be:

SFT ⟨CR⟩

The source host enters SFT protocol mode so that it can communicate with the SFT service. SFT responds with:

Enter commands for destination host; terminate with Ctrl-D.

Respond to this as outlined above.

If you wish to return to Connection level instead of beginning or resuming a dialogue with the hosts, you should terminate the host dialogue (either one) by typing Ctrl-D. SFT permits you to talk with both hosts even if you type Ctrl-D.

If for some reason you need to send a Ctrl-D character to either host, you may do so if you precede it with a Ctrl-V, which causes SFT to ignore the significance of the next character; this works for a Ctrl-V also.

If you get an error message when you type the SFT command, try it again — there may have been something in the host's input buffer. If that does not work, try lowercase, since some systems distinguish lowercase filenames from uppercase filenames.

SFT File Transfer Level

Once you have started both SFT host programs, SFT types:

From file(s)

Respond with the names of the source file(s) you want transferred, using the syntax of the source host, and terminating the entry with ⟨CR⟩. A Ctrl-D in this field returns you to Host Command level. A ⟨DEL⟩ character cancels anything you have typed on this line and causes SFT to repeat the prompt. Anything other than ⟨CR⟩, ⟨DEL⟩, or Ctrl-D is transmitted to the source host without change. The CP/M host program accepts a string of file names up to 256 bytes long. Future hosts may accept only one name at a time, or may accept even more complex specifications (e.g., wild card specifications).

Following your ⟨CR⟩, SFT displays the following on the same line:

To file(s)

If you want to assign the transferred file(s) the same name(s) at the destination as they have at the source, respond with ⟨CR⟩. If the destination filename must be different (e.g., because the two hosts use different filename syntaxes), enter the destination name, terminated by ⟨CR⟩. In this case, only one name may be given in the source field. As for the previous field, a ⟨DEL⟩ cancels all data entered on this line.

Once the To File(s) field is entered, SFT asks that you specify any file transfer options:

with options

There are two options you can specify: Overwrite and Translate. When the Overwrite option is specified, the contents of the existing file are replaced by the new file. The Translate option is used to deal with different host conventions for ASCII line termination (UNIX uses ⟨LF⟩, RIO uses ⟨CR⟩, and CP/M uses ⟨CR⟩ ⟨LF⟩). When the translate option is selected, hosts translate line terminators as shown on the next page:

	Stored in Files	*Transmitted/ Received*
UNIX	LF	CR
RIO	CR	LF
CP/M	CRLF	LF

Enter the options you want, terminated by a ⟨CR⟩. Any other punctuation is optional. All options are sent to both hosts, and those which are understood are acted upon. Options remain in effect only for the line on which they are entered. ⟨DEL⟩ works as above.

When you type ⟨CR⟩, SFT proceeds to transfer the file(s) you specified. As each file is being transferred, SFT types its name and periodically displays the number of bytes transferred thus far. Because SFT uses a 16-bit counter, the byte count can go past 65536; when this occurs SFT displays an asterisk (*) to the right of the byte count. You can abort the transfer once it has started by typing Ctrl-D.

When the specified transfer has been completed, SFT repeats the From files prompt, at which point you may answer as before, or you may type Ctrl-D to indicate that you have no more files to transfer and wish to return to Host Command level.

DEVICE OPTIONS

SFT normally expects that both the source and the destination are hosts that support the SFT protocol. You may specify options of either a terminal or printer by preceding either address with t= or p=.

When either option is selected, SFT does not ask for host commands. It also transfers raw text, rather than SFT protocol, when moving data to or from the specified device.

When SFT transfers data to a terminal, it displays a brief heading and it expands ⟨LF⟩ or ⟨CR⟩ into ⟨CR⟩, ⟨LF⟩, ⟨NUL⟩. It does no other modification of data; in particular, it transfers ⟨TAB⟩ and ⟨FF⟩ as ⟨TAB⟩ and ⟨FF⟩, which on some devices will be displayed as ⟨SP⟩ and ⟨CR⟩, respectively.

ERROR CODES

Common errors are reported by messages such as the following:

Can't establish connection.

No connection can be made to the designated host for the reason indicated by the code given. Compare the code to those issued by the command interpreter, listed in Chapter 13. Since some failures are transient, up to three connection attempts are made.

File access error.

Either the file could not be opened or some error occurred during transfer. An open failure at the source host most often indicates that the file requested does not exist (or that the name is misspelled). An open failure at the destination typically means that the file requested already exists and that the O option was not selected. It may also indicate that the device or directory is full or that write access is denied. Any error message returned by the host is also displayed. In the case of an open failure, these messages replace the Bytes Transferred status line for the file. In the case of an error during transfer, they follow the status line.

Connection broken.

This message indicates that the connection to the specified host has somehow been broken. This could be due to a network hardware or software failure or it could be caused by the Network Administrator's issuing a Disconnect command.

Timeout during input.

Timeout during output.

These messages indicate that although the connection has not been broken, the specified host has stopped sending (input) or receiving (output). This could be caused by a failure at the host, by an unterminated flow-control condition, or by lost data on the host connection cable.

Each of the messages just described is followed by a message that reads:

Last stream referenced was xxx.

where xxx is replaced by source or destination.

Less common errors are reported by messages of the form:

Error ###.

Last stream referenced was xxx.

where ### is replaced by one of the error numbers listed in Table 10-1 and xxx is replaced by source, destination or user. None of these errors is likely to be seen in normal usage; they are most likely to be seen when you implement host programs.

ERROR RECOVERY

If SFT goes to an error state, reset the APB board by turning the Reset switch on the front panel. If a host program becomes locked because of a transfer failure, connect to the host port over the network and type the sequence 〈44〉. You might need to enter this sequence twice. This instructs the host program to relinquish control to the host operating system.

SFT HOST PROGRAMS

This section provides the information required by anyone who wishes to implement an SFT host program. As described in the previous section, any SFT transfer requires an SFT program to be running on both the source host and the destination host, in addition to the SFT service that exists within the network. The SFT host and the SFT service combine to perform the tasks outlined in that section.

Table 10-1.
SFT Error Number Table

Number	*Description*
20	Packet received from designated host was too large (more than 256 data bytes).
30	Too many successive bad packets from designated host.
31	Packet with bad type received from designated host.
40	No acknowledgement received from designated host.
50	Packet with invalid SFT command code received from designated host.
52	Fatal error at designated host; file transfer command aborted.
53	Error of undefined severity reported by designated host.
60	No response from designated host to most recent packet.
70	Invalid response to request for ''don't send list.''
71	No termination string packet provided by the host.
80	Invalid device type in port address specified by user.
90	Excessively long response to last prompt.
100	Data packet expected but not received.

Implementation

Each SFT host is implemented as a slave program which simply responds to commands from the SFT service. Except for initialization sequences, described below, the SFT host initiates no actions on its own. Together, the server and host programs use a pair of protocols nested in the general form indicated on the following page.

⟨ABCDtext ~ HH⟩

where: ⟨ indicates start of packet

⟩ indicates end of packet

~ indicates that the next two bytes will be interpreted as the hexadecimal digits of a single byte

HH is a checksum formed by taking the exclusive-or (XOR) of the successive bytes of ABCDtext, rotating left one bit after each XOR

A is a one-byte sequence number (range 0-9, ASCII)

B is a packet-transfer command byte

C is a result byte (blank on input; S or F on output)

D is a file-transfer command byte

text is the text of the packet

The host module named sftph (simple file transfer protocol, host) deal with the CDtext portion of each packet. It receives commands (byte D) from the service, and responds to them with a packet that has the same value for D, S, or F in the C position (for success or failure, respectively), including any relevent text in the remainder of the packet. Table 10-2 lists the file transfer command codes and their meanings.

The commands B and E are sent only once per session. The F command is sent to the source host each time the user enters specifications in response to the From file(s) prompt. The N command is sent to the source host after each F command and each C command, until a null response is received, indicating that there are no more files. An S command and an R command are sent to the source and destination respectively, after each non-null response to an N. These are followed by a series of G and P commands to the source and destination, respectively, until the source sends a G response with no text, indicating end of file. C commands are then sent to both hosts. Unless sftph encounters a problem, it should place an S in the C field; in case of a problem, an F should be sent. In the F case, the text field may optionally be used for an error message.

Table 10-2.
Command Code Descriptions

Code	*Description*
B	Begin a dialogue.
E	End a dialogue and exit.
F	Accept a file specification string and store it.
N	Return the next file name that matches the file specification string. Several possibilities exist: a host might process only a single-name file specification string; it might process a list of names (as do the modules included on diskette); or it might process strings with wild card characters.
S	Open a file and prepare to send its contents.
R	Open a file and prepare to receive its contents.
C	Close a file.
G	Get another packet from the file and send it.
P	Receive another packet and add it to the file.

Certain commands, namely F, S, and R, contain an options field, preceded by the string 〈space〉O=. The options field follows the file name field and contains any text entered by the user in the option field of the From file(s) command. Options are single characters; punctuation and case should be ignored. The meanings of currently defined options are described in the section titled Using SFT earlier in this chapter.

The host module named sptph (simple packet transport protocol, host) deals with the 〈AB and HH〉 portions of the packet. It is responsible for assuring reliable exchange of data, and for transparent transmission of any data values. Sptph adds 〈 and 〉 to the beginning and end, respectively, of each output packet, and removes them from input packets. Because these characters may never appear in any of the internal fields of the packet, they always indicate the beginning and end of a packet. A sequence number, command code, and checksum are also added on output and checked on input. Sptph must always acknowledge any packet it receives correctly. It may ignore bad packets, or it may send a negative acknowledgement, but it should discard packets having the same sequence as the previous good packet. Likewise, the SFT service will acknowledge correct packets from the host; it is sufficient for the host SFT to simply repeat the last packet sent when it receives something other than an expected acknowledgement. The service will handle timeouts, and will abort sessions if too many consecutive unsuccessful transfers occur.

On output, sptph scans the text field and replaces certain characters with their hexadecimal equivalents, each preceded by a ~ (e.g., ⟨ is replaced by ~ 3C, while ~ is replaced by ~ 7E); this provides for data transparency. The characters to be handled in this way always include ⟨, ⟩, ~, ⟨DLE⟩, ⟨DC1⟩, ⟨DC3⟩, and FD (hex); to these the host may add any characters it wishes to avoid sending or receiving. This list is maintained in the array dsl (don't send list). If 8 is included in the list, all characters with bit 8 set to 1 are sent in hexadecimal format.

On input, sptph converts sequences of the form ~ xx back to single bytes. Sptph recognizes and sends the commands listed in Table 10-3.

As a general rule, SFT works more efficiently if the host's console port can be placed in raw input mode. This can reduce the size of the don't send list, sometimes dramatically. For Unix, this is accomplished via the shell script sft, which enters raw mode, invokes the host program sfth, and upon termination restores normal mode. For CP/M, the macros put(c) and obtain(c), defined in sftio.ch, provide access to the raw mode.

Initialization

Several special requirements exist for initialization. The first requirement is that the SFT host program must send the following sequence immediately after starting up:

```
⟨OAXXReady ~ 6F⟩
```

This is a signal to the SFT server that an SFT program has been started. Without such a signal, users of the SFT program would have to provide one. Users of the SFT program should be careful not to have the host send the above sequence while at the SFT Host Command level. However, the above sequence may be sent during the file transfer.

Immediately after sending the Ready sequence, above, the SFT host program should send the following sequence:

〈1AXXnkkk... ~ xx〉

where n is the number of bytes in the string that follows; kkk... is a sequence of 0 to 10 line termination characters (required only if the host's I/O subsystem requies input in complete lines. Most do not; for these n is 0 and kkk... is omitted); and xx is the appropriate checksum.

Table 10-3.
Host (Sptph) Commands

Host Input Commands	*Host Output Commands*	*Meaning*
0	A	Data packet to/from next level (i.e., sftp)
1	B	Acknowledgement
2		Reset sequence numbers to 0
3	D	Specify additional characters for "don't send list"; this is an inquiry from the service, to which the host should respond with any special characters it does not wish to receive.
4		Abort. Useful when connections have been broken and host may be in input mode; since the sequence number need not be checked, 〈44〉 causes host SFT to terminate and return to Host Command level.
5		Enter debug mode. Turns on echoing and does not require checksums; useful when debugging a host SFT that is directly connected to a terminal; the sequence number is not checked for this packet.

Chapter 11
Ethernet Data Link Service

Contents

OVERVIEW

The Codex LAN Ethernet Data Link Service (EDLS) is for user programs running on hosts connected to Codex LAN Entryways; the service does not have a human interface. EDLS provides a simple host-to-host communications protocol via the Ethernet Data Link Layer. A host attached to an Entryway using EDLS can communicate with any other Ethernet-compatible host on the network whether or not the other host is also attached to an Entryway.

The EDLS software runs inside an Entryway. Each Entryway running the service runs its own copy of the software. While an Entryway is running the EDLS software, it cannot run any other Codex LAN software, or communicate with an Entryway running other Codex LAN software.

The Ethernet Data Link Service carries data via the Ethernet Data Link Layer. EDLS also responds to requests from a local host program for information which it must have to manage the data transfer operation, such as the host's network address.

EDLS sends information in structures called Ethernet packets. Any host or device that supports packet formats that conform to the Ethernet specification can receive these packets. To conform to this specification, EDLS does not append any headers to the packets other than the data link header defined in the specification. An EDLS user program does not need to know what the Ethernet-defined packet format is. A user program supplies the information the service needs, and the service formats and sends the information.

The Ethernet Data Link Service assures best effort delivery of packets for a host, but does not guarantee delivery. The service also does not guarantee that a series of packets will be delivered in the order sent, or that a packet will not occasionally contain corrupted data. This service is meant for user applications that can tolerate some errors, or provide their own recovery protocols.

EDLS does not supply internet headers for packets. Therefore, the service cannot transmit a packet via a Codex LAN bridge to other Codex LAN networks. When internet packet routing is needed, the Codex LAN Datagram Service may be a better choice because it interfaces to the network software at the

Network Layer (see Chapter 12). The two services can coexist on the same Codex LAN network, although not on the same Entryway.

Although EDLS is for user programs that run inside hosts, sending and receiving Ethernet packets is possible for user programs that run within an Entryway. A program running inside an Entryway can communicate with any Ethernet-compatible host on the network when the Entryway is configured to use the Ethernet Data Link Interface (EDLI).

EXAMPLE OF DATA FLOW

Figure 11-1 illustrates the data flow between a host computer (Host A), the Entryway to which it is connected, and another host computer (Host B). The user program running on Host A uses the Ethernet Data Link Service running on its Entryway to exchange data with Host B. Although a message travels from the program on one host to the program on the other host, a program using EDLS only exchanges data directly with the service. EDLS handles all necessary network communications.

Before the user program on Host A uses EDLS to send data, the program should perform certain administrative operations. The figure shows only one of these operations, the inquiry from Host A to its Entryway: Who Am I? The figure also shows the service response: You Are.

The user program running on Host A requests the host's address from the local Ethernet Data Link Service. After receiving the response, the user program on Host A sends a message to the user program running on Host B.

The message first goes to the EDLS running on the Entryway attached to Host A. Note that the service does not send an acknowledgement to the host program when it receives the message. It is up to the user program to detect and handle any errors. When the Ethernet Data Link Service receives the message, it formats and sends the message to Host B.

After receiving the message from Host A, the program running on Host B sends a message to the program running on Host A. This message first goes over the network to the Ethernet Data Link Service running on the Entryway attached to Host A, and finally, to the user program running on Host A.

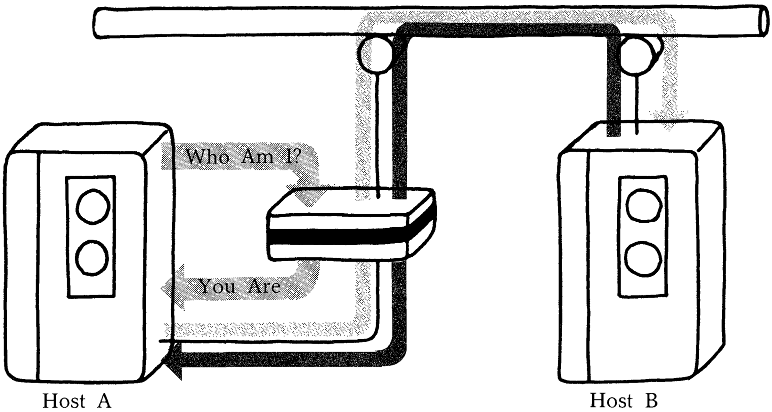


Figure 11-1. Flow of Data and Control Information

As you can see, the Ethernet Data Link Service on an Entryway is an intercessor for a user program running on an attached host. It communicates with the other Ethernet-compatible devices or hosts on the network in order to send and receive data. The local Ethernet Data Link Service also provides the administrative information and services a user program needs to manage the data transfer.

COMMUNICATIONS INTERFACES SUPPORTED

Currently, the Ethernet Data Link Service supports an IEEE-488 high-speed interface, and a 32-bit Digital Equipment Corporation (DEC) DR11-W or DR11-B parallel interface. In order to use the Ethernet Data Link Service, the host must use one of these two interfaces, and the attached Entryway must be configured with the appropriate Codex LAN I/O module. Each of these I/O modules contains a high-speed, block oriented driver that handles data in block units rather than byte-by-byte.

USING THE ETHERNET DATA LINK SERVICE

Before a host program can use the Ethernet Data Link Service, the attached Entryway must be configured to run the service. Use the Configure program to accomplish this task.

When you use the Ethernet Data Link Service, you must provide the host computer's driver interface for the hardware. This driver formats, sends, and receives information for host programs that use the service.

The communications interface between a host and an Entryway determines the method for transferring information between a user program and the service. You must be familiar with the interface your host uses in order to write the required driver.

A program using the Ethernet Data Link Service must format the information that it sends to the service. The information formats that the service recognizes are described next.

FRAMES

The data and information exchanged between a user program and EDLS is sent in structures called frames. A frame contains header information that identifies the contents of the frame, and the information a host is sending to another host or its Entryway. A frame that contains user data is called a Data Frame, and a frame that contains control information for the Entryway is called a Command Frame.

The frame is a data structure used only by a host and its Entryway. Only the data in a Data Frame, including the Ethernet address, is transmitted by the Entryway over the network. The EDLS user need not be concerned with the format of the packets sent over the network; EDLS does all the formatting necessary.

Frame Header

Both Data Frames and Command Frames begin with a 6-byte Frame Header. The Frame Header provides infomration such as the length of the frame, the type of frame, and the purpose of the frame. Figure 11-2 shows the layout of this header, and Table 11-1 explains each of the header fields. ■ **Note:** The most significant byte of all multibyte numeric fields is transmitted first.

To compute a checksum, first convert each byte of the header to an 8-bit binary quantity, then use the following algorithm to determine the value.

0	Frame Length	
2	Frame Type (02)	Reserved (00)
4	Function Code	Block Check Character

Figure 11-2. Frame Header Format

BCC computation;

bcc = 0;

for i = 0 to 4

    do bcc = bcc XOR headerbyte [i];

The host must compute the checksum and insert it in Byte 5 when sending a frame to the Ethernet Data Link Service. The Ethernet Data Link Service also computes the checksum of each frame it sends to a host.

EDLS verifies the checksum of each frame that it receives. When the exclusive OR of the entire 6-byte header is 0, the checksum is correct. When the checksum is not correct, the Ethernet Data Link Service discards the frame. A host should also verify the checksum of each frame that it receives, discarding any frame with an incorrect checksum.

Table 11-1.
Frame Header Field Descriptions

Bytes	*Description*
0-1	Length of the frame, including the header, in hexadecimal. The most significant byte of the value is Byte 0; Byte 1 is the least significant byte of the value.
2	Identifies the program within the Entryway that is sending or receiving the frame. This byte must be 02 (hex) for all EDLS frames.
3	Reserved by the system; must be 00 (hex).
4	Specifies the purpose of a frame. The code 04 (hex), for example, indicates that a host is requesting its network address. Table 11-2 lists all the codes and their meanings.
5	The result of a bytewise exclusive OR (XOR) of the first five bytes of the Frame Header. For example, the checksum of the header of a frame 30 bytes long (1E hex) with a Function Code of 01 is 1D, computed as follows: 00 XOR 1E XOR 02 XOR 00 XOR 01 = 1D

Table 11-2 lists all the Ethernet Data Link Service function codes and their meanings. The two "Here Is Data" codes identify a frame as a Data Frame. The remaining codes are the commands that the host and the Entryway use to manage the exchange of Data Frames. The functions are explained in detail in the following sections on the Data Frame and the Command Frame.

Table 11-2.
EDLS Function Codes

Command	*Code (in hex)*	*Description*
Here Is Data (Host to Entryway)	00	Indicates that the host is passing data to the Entryway for transmission.
Here Is Data (Entryway to Host)	01	Indicates that the Entryway is passing data to the host from another host.
NOP Request (Host to Entryway)	02	A host sends this command to find out whether its Entryway is operational.
NOP Response (Entryway to Host)	03	Entryway response to the NOP request; it informs a host that the Entryway is operational.
Who Am I (Host to Entryway)	04	Requests the host's 6-byte network address.
You Are (Entryway to Host)	05	Supplies the host's 6-byte network address.
Set E-type Filter (Host to Entryway)	16	Sets filter to a single value.
	26	Sets filter to all E-types except those the receiver hardware traps (see the section titled Packet Filtering later in this chapter).
Set E-type Filter Response (Entryway to Host)	07	Acknowledges the requested E-type filter setting.
	08	Disable receiver.
	18	Enable unique address.

Table 11-2 (continued).
EDLS Function Codes

Command	*Code (in hex)*	*Description*
Set Receiver Control (Host to Entryway)	38	Enable unique address and broadcast.
	78	Enable unique address, broadcast, and manufacturer's broadcast.
	B8	Enable unique address, broadcast, and multicast.
	F8	Enable promiscuous mode.
Set Receiver Control Response (Entryway to Host)	09	Acknowledges the requested receiver control.
Entryway__Reset (Entryway to Host)	0A	Informs host that Entryway was reset.
Entryway__Reset__Seen (Host to Entryway)	0B	Host acknowledgement of message that Entryway was reset.
Clear (Host to Entryway)	FE	A host request to the Entryway to clear the registered E-type and the registered receiver control mode.
Cleared (Entryway to Host)	FF	Informs host that Entryway has cleared the registered E-type and the registered receiver control.

DATA FRAME

A Data Frame consists of the Frame Header, addressing information, and user data. The addressing information allows a host program to address a packet to any Ethernet-compatible station on the network. Figure 11-3 shows the format of a Data Frame, and Table 11-3 explains each of the fields.

Note that the numbers of the bytes within a frame are in decimal format, and that the values within the fields of the frame are in hexadecimal format. The one exception to the

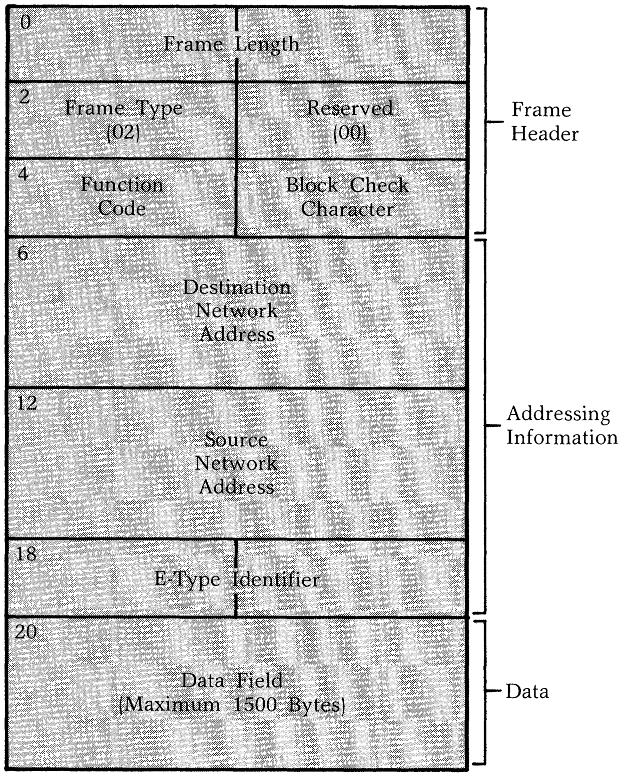


Figure 11-3. Data Frame Format

value format is the maximum number of bytes in the Data Field of a frame; this value is in decimal format. ■ **Note:** The most significant byte of all multibyte numeric fields is transmitted first.

In order to conform to the Ethernet standard Version 1.0, the Data Field of a Data Frame should not be shorter than 46 bytes or longer than 1500 bytes. When the Data Field of a Data Frame is shorter than 46 bytes, the service pads the Data Field to make it 46 bytes long. The contents of the added bytes are unpredictable. It is up to the user to supply at least 46 bytes or to use a delimiter to indicate the end of valid data. Packets with more than 1500 bytes of data are discarded.

Table 11-3.
Data Frame Field Descriptions

Field (bytes)	*Description*
Length (0-1)	The length of the frame, in binary, as a byte count including the header.
Frame Type (2)	This field must be 02 (hex) for the Ethernet Data Link Service.
Reserved Byte (3)	Reserved by the system; must be 00 (hex).
Function Code (4)	Defines the contents of the frame as data. This byte contains the value 00 (hex) when the host is sending data to the Entryway; this byte contains the value 01 (hex) when the Entryway is sending data to the host.
Block Check Character (5)	The block check character is the 8-bit result of a bytewise exclusive OR (XOR) of the first five bytes of the header.
Destination Network Address (6-11)	The 6-byte address of the destination host (see Table 11-4).
Source Network Address (12-17)	The 6-byte address of the source host (see Table 11-4).
E-Type Identifier (18-19)	The E-type of the frame (see the section titled Packet Filtering later in this chapter).
Data Field (20-end)	Any data format is allowed in this field. The minimum data length is 46 bytes; the maximum data length is 1500 bytes.

Data Frame Addressing

Each Data Frame contains the 6-byte network addresses of the source host and the destination host. A host program must supply these addresses for each Data Frame that it sends. A host issues the Who Am I command (described later) to find out the source address; there is no command that returns the destination address.

Table 11-4.
Host Address Format

Bytes	*Contents*
0-2	Vendor Identifier. For Codex LAN equipment, these bytes are 00 DD 00 (hex) for host-to-host addressing, and 01 DD 00 (hex) for multicast addressing.
3-4	Entryway node number, in hexadecimal. The most significant byte is first. For example, Entryway 55 is 00 37 hex.
5	Host port and board number within an Entryway. The upper four bits are the port number. Boards A through D are numbered 0 through 3, respectively. 0 is the only valid value for the port.

Both the source address and the destination address have the same format. Table 11-3 shows the format of the 6-byte Codex LAN address. The contents of the first three bytes are the same regardless of the manufacturer of the host; the contents of bytes three, four, and five are manufacturer-dependent.

EDLS allows a host to send data to one or more hosts. The three forms of addressing used are described below.

- ***Host-to-Host.*** The destination address specifies a single host.
- ***Broadcast.*** The destination address specifies all hosts. Set all six bytes of the destination address to FF (hex) to send a broadcast message.

  When the destination hosts are Codex LAN Entryways, it is possible to combine specified and broadcast values in a destination address. In fact, any of the destination address fields (Vendor Identifier, Entryway, board, or port) can contain a broadcast value or a specific value. This capability allows you, for example, to send a message to all ports on Board A of Entryway 213, all ports on Board A in all Entryways, or all ports on board A of Entryway 57. Set all bytes of a field to FF (Hex) to specify a broadcast value for that field.
- ***Multicast.*** The destination address specifies a group of hosts rather than a subset of Entryways, boards, or ports.

To send a multicast message, set the least significant bit of the first byte of the Vendor Identifier to 1. For example, the Codex Vendor Identifier is 00 DD 00 for a Host-to-Host message and 01 DD 00 for a multicast message.

The user defines the significance of the remaining address bits. Although all hosts receive a multicast message, each host uses the remaining bits of the destination address to determine whether or not it is included in the group of intended receivers.

EDLS does distinguish between a multicast message intended for Codex equipment and a multicast message intended for another manufacturer's equipment. Multicast message refers to all messages that have the least significant bit of the first byte set to 1. However, a multicast message sent to Codex equipment is also called a manufacturer's multicast message to distinguish it from a multicast message sent to another manufacturer's equipment.

Addressing Examples

The following examples show how to use the various addressing modes. For brevity, only the 6-byte Codex LAN Destination Station Address field of the header is given (hexadecimal notation). The six bytes given in each example are as follows:

- 3-byte Vendor Identifier
- 2-byte Entryway Number (in hex)
- 1-byte port and board number

When sending a Data Frame to equipment not produced by Codex, the first three bytes should contain the appropriate Vendor Identifier. However, each manufacturer defines the contents of the remaining three bytes.

Examples:

- The host sends a packet to Entryway 18, Board B, Port 0:

Send:

00	DD	00	00	12	01

00 DD 00 = Vendor Identifier
00 12 = Entryway 18
01 = Destination port/board
0 = Port 0
1 = Board B

- The host sends a broadcast message to every station on the network, regardless of vendor:

Send:

FF	FF	FF	FF	FF	FF

FF FF FF = Broadcast value
FF FF = Broadcast value
FF = Broadcast value

- The host sends a message to all ports and boards on Codex LAN Entryway 20:

Send:

00	DD	00	00	14	FF

00 DD 00 = Vendor Identifier
00 14 = Entryway 20
FF = Every port and board

- The host sends a multicast message to Codex LAN user-defined Group 2:

Send:	01	DD	00	00	02	FF

01 DD 00 = Vendor Identifier (multicast bit set)
00 02 = User-defined designation for Group 2
FF = Every port and board

The following chart shows a complete Ethernet Packet sent from a user program running on a host attached to Entryway 18 at Board D, Port 0. The program sends the message "HELLO". The program addresses the packet to Entryway 20 at Board A, Port 0.

Byte	*Contents*	*Description*
0	00	Length of header and data (25 decimal)
1	19	
2	02	Frame Type
3	00	Reserved by system; must be 00
4	00	Indicates data is sent from host to Entryway
5	17	Block check character
6	00	Vendor Identification of destination host
7	DD	
8	00	
9	00	Destination Entryway (14 hex, 20 decimal)
10	14	
11	00	Port 0, Board A

Byte	*Contents*	*Description*
12	00	Vendor Identification of source host
13	DD	
14	00	
15	00	Source Entryway (12 hex, 18 decimal)
16	12	
17	03	Port 0, Board D
18	XX	E-type of frame
19	XX	
20	48	H
21	45	E
22	4C	L —5 bytes of data in ASCII
23	4C	L
24	4F	O

Packet Filtering

EDLS always examines two fields in each Ethernet Packet: the Destination Address and the E-type. The values in these fields determine whether the host should receive the packet.

The format of the address and the types of addressing available are given above. The host chooses which type or types of addressing are valid. For example, a host that chooses Host-to-Host addressing receives only those packets that are addressed to that host. The host will not receive any Broadcast or Multicast packets.

The E-type of a packet is a 2-byte quantity defined by the communicating user programs. The host can choose to accept packets only if they have a particular E-type.

Arbitrarily choosing E-type values is not recommended. Xerox Corporation licenses and registers E-types so that no conflicts arise among vendors creating software for Ethernet-compatible equipment. A user writing a program that uses EDLS to send or receive packets should apply to Xerox for the E-type values needed.

0	Frame Length	
2	Frame Type (02)	Reserved (00)
4	Function Code	Block Check Character
6	Parameters (When Necessary)	

Figure 11-4. Command Frame Format

COMMAND FRAME

All Command Frames begin with the 6-byte Frame Header described previously. Some commands also require parameters. When parameters are necessary, they follow the Frame Header. Figure 11-4 shows the format of a Command Frame.

Ethernet Data Link Service Commands

The EDLS commands manage the exchange of Data Frames. The commands are divided into sets that consist of one or more requests and a response. The command definitions specify all necessary parameters, their length, and their order within the frame. The commands are presented in the order in which a host would typically use them.

The first four commands are presented as a set. They ensure that the Entryway and the host take appropriate actions whenever one or both of them are initialized. When any of these events occurs, the Entryway clears any previous E-type value and Receiver Control mode registered for the host, and the host registers or reregisters for these parameters.

The NOP/NOP Response command pair is presented next because a host should verify that the attached Entryway is working before it issues any of the other commands.

The next command pairs presented are the commands that allow the host program to specify the receiver setting and the E-type necessary. The E-type command pair precedes the receiver control command pair because a host program should issue the commands in this order. If a host program specified the Receiver Control mode before registering an E-type, the host might receive packets it could not handle. When a host program wants to change the current Receiver Control mode or register a different E-type, the order of the command does not affect the operation of the service.

This chapter ends with the definition of the command pair the host uses to obtain its 6-byte address.

Entryway_RESET, Entryway_RESET_SEEN, CLEAR, CLEARED. These commands are part of the Reset Handshake Protocol that a host and its Entryway use whenever either one is initialized.

Function Codes:	Entryway_Reset	0A (hex)
	Entryway_Reset_Seen	0B (hex)
	Clear	FE (hex)
	Cleared	FF (hex)

Parameters: none

Usage Notes: The host and Entryway algorithms for the Reset Handshake Protocol are described later in this chapter. These four commands are part of this protocol. The Reset Handshake Protocol ensures that the host and the Entryway are aware of the operation whenever the other is initialized. The protocol requires the Entryway to clear the E-type and Receiver Control mode registered for the host.

NOP, NOP Response. A host program sends the NOP request to verify that its Entryway is operational. The NOP Response is confirmation that the Entryway is operational.

Function Codes:	NOP	02 (hex)
	NOP Response	03 (hex)

Parameters: none

Usage Notes: When the Entryway sends the NOP Response, the host program knows that it can use the Ethernet Data Link Service. If the host receives no response to the NOP request, it knows something is wrong with the Entryway.

Set E-Type Filter. This command sets the E-Type filter to one of two states.

Function Codes:	Single E-type value	16 (hex)
	All E-types (except those trapped by receiver hardware)	26 (hex)

Parameters:

Bytes	***Description***
6-7	The E-type value for the single E-type filter mode (16 hex). This parameter is not present when the Function Code for the frame is 26 (hex), all E-types.

Usage Notes: The Set E-type Filter command allows a host program to restrict the packets it receives to those having a certain E-type. For example, when a host program registers with the local Ethernet Data Link Service for E-type 10, the Entryway passes to the host only those packets with an E-type Identifier of 10.

The default E-type filter value is 26 (hex), all E-types. However, when the host wants to receive only packets that have a particular E-type value, the host should register this value before setting the Receiver Control mode.

A user writing a program that uses EDLS to send or receive packets should not arbitrarily choose E-type values. Xerox licenses and registers E-type values so that no conflicts occur among vendors offering software for Ethernet-compatible equipment.

Set E-Type Filter Response. Acknowledges the two filter commands.

Function Code: E-type Filter Response 07 (hex)

Parameters: When the E-type Filter Request specifies a single E-type value:

Bytes	*Description*
6-7	The E-type value specified
8	The Function Code of the request, 16 (hex).

When the E-type Filter Request specifies 26 (hex), All E-types:

Bytes	*Description*
6	The Function Code of the request, 26 (hex).

Usage Notes: The position of the Function Code in the frame depends upon whether the request specifies one E-type or all E-types.

Set Receiver Control. Sets the Receiver Control Mode of the host.

Function Codes:	Disable	08 (hex)
	Enable unique address	18 (hex)
	Enable unique address and broadcast	38 (hex)
	Enable unique address, broadcast, and manufacturer's multicast	78 (hex)
	Enable unique address, broadcast, and all multicast	B8 (hex)
	Enable promiscuous	F8 (hex)
Parameters:	none	
Usage Notes:	The Receiver Control Mode is one of the two packet filtering mechanisms availabe to EDLS users; the other is frame E-type. EDLS discards any packets that do not match the registered Receiver Control mode. For example, when a host specifies Function Code 38 (hex), the host receives any Ethernet packet sent to its address and all broadcast messages, but does not receive multicast Ethernet packets.	

EDLS checks each incoming packet for its address mode: Host-to-Host, Broadcast, or Multicast. The service then determines whether this packet meets the requirements of the Receiver Control mode that the host set. When a packet passes these requirements, the service checks to see if the E-type of the packet matches the E-type registered by the host program. When the E-types match, EDLS passes the packet to the host. Packets that have an incorrect Receiver Control mode or E-type are discarded. ■ **Note:** When the receiver is set to promiscuous mode, the host receives all packets on the network that have the registered E-type.

Set Receiver Control Response. Confirms the requested Receiver Control Mode.

Function Code:	Set Receiver Control Response	09 (hex)

Parameters:

Bytes	*Description*
6	The Receiver Control Mode specified in the Set Receiver Control request.

Usage Notes: The Entryway response to the host program's request for a particular Receiver Control mode includes the mode requested so that the host can confirm that the correct mode was registered.

Who Am I? Requests the 6-byte Codex LAN address of the host.

Function Code:	Who Am I	04 (hex)
Parameters:	none	

Usage Notes: A host program should use this command to determine the 6-byte source host address for Data Frames. Although it is possible for a host program to code the Entryway address as a constant in a program, it is better programming practice to use the Who Am I? command to determine this address.

You Are. Returns the 6-byte Codex LAN address of the requesting host.

Function Code: You Are 05 (hex)

Parameters:

Bytes	*Description*
6-11	The 6-byte Codex LAN address of the requesting host.

Usage Notes: The You Are Command is the Entryway's response to a Who Am I? command. The Command Frame for the You Are command contains the 6-byte address of the host that sent the Who Am I? command. The format of the address is as follows:

Bytes	*Contents*
0-2	The Vendor Identifier. For Codex LAN equipment, these bytes are 00 DD 00 (hex) for host-to-host addressing, and 01 DD 00 (hex) for multicast addressing.
3-4	The Entryway node number, in hexadecimal. The most significant byte is first; the least significant byte is second. For example, Entryway 55 is 00 37 hex.
5	The host port and board number within an Entryway. The upper four bits are the port number. Boards A-D are numbered 0-3, respectively. 0 is the only valid value for the port.

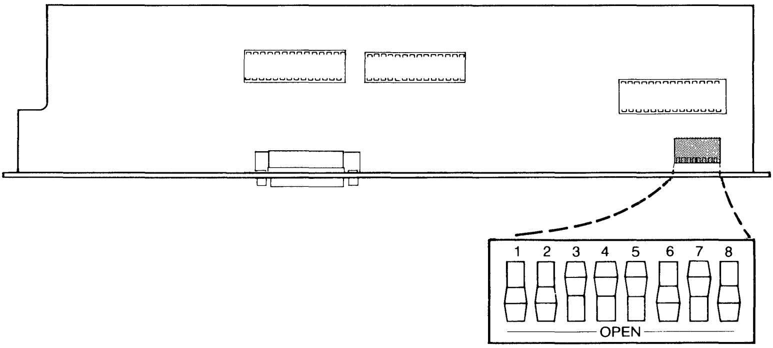


Figure 11-5. IEEE-488 I/O Module Address Sense Switch Location

IEEE-488 ETHERNET DATA LINK SERVICE INTERFACE

This section provides information on sending and receiving Ethernet Data Link Service frames for users with programmable IEEE-488 interfaces. You should have some knowledge of the IEEE-488 General Purpose Interface Bus (GPIB) before programming an application that uses this interface to the service. This manual does not provide a complete description of the interface.

Hardware Requirements

The IEEE-488 I/O Module and Processor Board combination must be installed in the Entryway to use this interface to the Ethernet Data Link Service. The GPIB address for the Entryway and the parallel poll response bit assigned to the Entryway are both set by an address sense switch located on the I/O Module. Refer to Figure 11-5 for the location of this switch. ■ **Note:** The sense switch is located as shown in Figure 11-5. Do not confuse this switch with the processor board sense switch.

The address sense switch is an 8-bit DIP switch. The bit positions on the switch are zero when placed to Off, and one when placed to On. Bit positions 1 through 5 on the address sense switch correspond to the GPIB bus addresses, 0 through 31. Bit positions 6, 7, and 8 on the address sense switch define

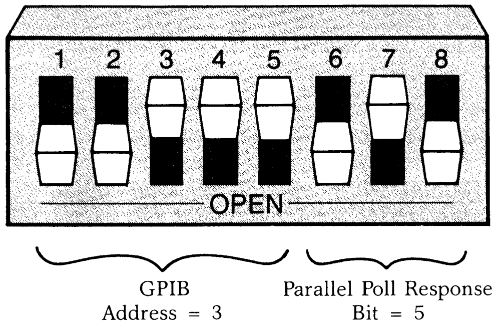


Figure 11-6. Address Sense Switch Setting Example

the parallel poll response bit (0-7) that the Entryway will assert when the controller conducts a parallel poll. Figure 11-6 shows a sample address sense switch setting.

Switch 1 on the DIP switch sets the least significant bit of the Bus Address. Switch 6 sets the least significant bit of the parallel poll response bit.

Software Requirements

You must use the CONFIGURE program to configure the Entryway for the Ethernet Data Link Service software. The host computer must implement all frame construction, sequencing, acknowledgement, and error recovery logic.

Using EDLS with the IEEE-488 Interface

ELDS/488 is a software module that allows the Entryway to act as an IEEE-488 peripheral to the host. This section assumes that the host program uses an IEEE-488 driver that provides some degree of standard control over an IEEE-488 I/O port.

Host-to-Entryway Frames. When the host has a frame to send, it should conduct a parallel poll on the IEEE-488 interface to verify that the Entryway is operational and has an input buffer available. When the parallel poll response bit is set, the Entryway is ready for data.

Once the host determines that the IEEE-488 interface is ready, the host or controller should first assert ATN (Attention), and then UNTALK any devices that might have data gated onto the GPIB. The host/controller should then UNLISTEN any devices for which the frame is not included. The host/controller then issues the LISTEN ADDRESS of the Entryway.

When the host is not the GPIB controller, the controller should issue the TALK ADDRESS of the host before ATN is released. When the host is the controller, it should enter the TALK-ONLY state and release ATN. The host can then transfer a frame across the GPIB to the Entryway.

EDLS/488 assumes that the first two bytes transferred are the frame length field. EDLS/488 reads the length fields and attempts to input the corresponding number of bytes. Frames with lengths less than 6 bytes or greater than 1520 bytes are discarded; 6 and 1520 are the lower and upper limits, respectively, of a frame. Any frame transfer is aborted by asserting IFC (Interface Clear) or UNLISTEN.

Entryway-to-Host Frames. When EDLS/488 receives a frame from the net to transfer to the host, EDLS/488 requests service from the host/controller by asserting SRQ (Service Request) on the GPIB. When devices other than the host and Entryway are attached to the GPIB, or when the host is not the controller, it may be necessary for the host/controller to conduct a serial poll to verify that the Entryway is requesting service. You must use a serial poll. You cannot use a parallel poll to verify an Entryway service request becuase the parallel poll response bit is used as the ready-for-data indicator.

Once the host/controller establishes that the Entryway is requesting service, the host/controller should assert ATN. The host/controller should then UNTALK any devices that might have data gated onto the GPIB, UNLISTEN and devices for which the frame is not intended, and issue the TALK address of the Entryway.

When the host is not the GPIB controller, the LISTEN ADDRESS of the host should be issued. When the host is the GPIB controller, it should enter a LISTEN ONLY state before releasing ATN.

The Entryway transfers the frame across the GPIB, beginning with the length field. Note that the two bytes of the length field are included in the frame length.

When the host/controller issues a TALK ADDRESS of the Entryway and no frame is present, a null frame consisting of a length field of 00 02 hex is transferred. The last byte of the frame transferred by EDLS/488 is accompanied by the EOI (End of Identify) signal to indicate completion. Any frame transfer is aborted by asserting IFC or UNTALK. Both IFC and UNTALK return the Entryway to the talker idle state and reassert SRQ.

EDLS/488 Functionality. Table 11-5 describes the subset of the IEEE-488/1978 standard implemented by EDLS/488.

DR11-W ETHERNET DATA LINK SERVICE INTERFACE

The EDLS/DR11 interface provides access to the Ethernet Data Link Service for Digital Equipment Corporation (DEC) computers equipped with DR11-W Unibus DMA controllers, or other machines equipped with a DR11-W interface. EDLS/DR11 also supports the DR11-B interface. All references to DR11-W include DR11-B unless otherwise noted.

This section should help you generate a host I/O driver for the EDLS/DR11 interface to Codex LAN. It assumes that you have some knowledge of both the DEC DR11-W DMA controller and the Ethernet Data Link Service.

Hardware Requirements

The 32-bit Parallel I/O Module and Processor Board combination must be installed in the Entryway to use the Ethernet Data Link Service. Communication between the host and the Entryway occurs over the host's DR11 cables. A DR11 adapter attaches at the rear panel of the Entryway between the I/O ports of the 32-bit Parallel I/O Module and the DR11 cable. Refer to the Codex 4000 Series LAN Hardware Reference Guide for a discussion of the 32-bit Parallel I/O Module.

Table 11-5.
EDLS/488 Functionality

Function	*Description*
SH1	Source handshake.
AH1	Acceptor handshake.
T6	Basic talker and serial poll mode. The lower five bits of the I/O Module sense switches define the GPIB interface address. EDLS/488 supports serial poll mode. The EDLS/488 interface leaves the talk-addressed state when the listen address is issued.
TE0	Extended talker mode is not supported.
L4	Basic listener capability implemented by the EDLS code. The interface exits the listen-addressed state when the talk address is issued.
LE0	Extended address listener mode is not supported.
SR1	Service request function is implemented, with SRQ always signifying the availability of an output buffer for the host.
RL0	Remote/local mode setting is not supported.
PP2	Parallel poll capability possible only with local configuration. The upper three bits of the I/O Module sense switch define the bit used for a parallel poll response; a parallel poll response indicates that the interface is ready for input from the host. A parallel poll is not a substitute for a serial poll that verifies SRQ output.
DC0	Selective device clearing is not supported.
DT0	Selective device triggering is not supported.
C0	Bus controller mode is not supported.

Software Requirements

You must use the Configure program to configure the Entryway software for the Ethernet Data Link Service. The host computer must implement all frame sequencing, acknowledgement, and error recovery logic.

EDLS/DR11 Interface

This section defines the characteristics of the EDLS/DR11 interface. This interface allows the Entryway to operate as a high-speed peripheral device to a DEC system equipped with a DR11-W or DR11-B module. The terms used in the following descriptions are derived from the DEC document, *DR-11W Direct Memory Interface Module User's Guide.*

Data Formatting. The Entryway has an 8-bit data path, while the DEC Unibus has a 16-bit data path. The EDLS/DR11 interface accepts and transmits data while preserving the order of the DEC data strings. Therefore, the bytes of any 16-bit element of the Frame Header must be swapped before transfer to the Entryway.

When an odd number of bytes is transferred, EDLS/DR11 transfers an additional byte to complete the last word. The extra byte has no effect on the Frame Length field because EDLS/DR11 discards this additional byte before transmitting the frame.

Entryway Status Indications. DR11-W has three status input lines: STATUS A, B, and C. EDLS/DR11 assigns them the following meanings:

Status	*Description*
A	Error. A host request for an invalid operation sets the A status; the completion of a valid operation or a RESET resets the status.
B	Frame Ready. When a frame for the host is queued in the Entryway, the B status is set; when the host has read or flushed all frames, it is reset.

Status	*Description*
C	Network Interface Ready. The C status is set when the Entryway can accept a frame from the host; it is reset when no output buffers are available.

ELDS/DR11 loads an additional status byte on the lower half of the interface Databus upon completion of a command or software reset. This byte is accessed by the host through the Input Data Register of the DR11-W. Bit 0 of this byte (Reset Acknowledge) is used to regain synchronization after the RESET command or an INIT pulse aborts a command. This bit is set when a RESET command terminates or when software intialization occurs, and reset when any other command terminates. All other bits of this byte are undefined.

Interrupt Requests. EDLS/DR11 initiates an interrupt request by asserting the ATN signal to the DR11-W. The ATN line is pushed high for roughly three microseconds for the interrupt request. Interrupt requests are issued to indicate the completion of a transfer or control function requested by the host.

Interface Control. Commands from the host are presented to EDLS/DR11 by placing a 3-bit function code on the lines FNCT 1, FNCT 2, and FNCT 3. A 200-nanosecond active pulse is asserted on the GO line. Table 11-6 describes the function bits and the associated commands. An undefined command results in an error state termination. Any valid command clears a previous error state.

Table 11-6.
Function-Bit Commands for EDLS/DR11

Function-Bit 3	2	1	Command Description
0	0	0	RESET. Halts any data transfer in progress across the interface, taking the interface out of the active idling state if it was in that state. Sets the Reset Acknowledge bit.
0	0	1	PREVIEW FRAME. Transfers to the host up to 32 bytes from the beginning of the frame.
0	1	0	OUTPUT FRAME. Allows transfer of a frame from the host into the Entryway when the Entryway is ready for data. The command generates an error condition when the network interface is not ready, or the frame has an invalid length field.
0	1	1	READ FRAME. Transfers the remainder of the referenced frame into the host and clears the internal buffer when preceded by the PREVIEW FRAME command. When not preceded by a PREVIEW FRAME command, READ FRAME transfers the entire frame. When no frame is available, the command generates an error condition.
1	0	0	FLUSH FRAME. Discards the frame at the head of the host-bound queue, and clears the internal buffer. This command can be preceded by a PREVIEW FRAME command, but it is not required. An error results when no frame is available at the interface.
1	1	0	IDLE. IDLE terminates with an interrupt request when a frame is available for the host, or when an interface busy condition clears. The command terminates immediately if either event occurs after the last interrupt request but before the next IDLE command. The RESET command overrides the IDLE command when you want to leave the idle state to perform a data transfer operation.

Using EDLS with the EDLS/DR11 Interface

This section describes the command sequences, data formats, and signaling conventions required when using the DR11-W interface for host-Entryway communication. The Entryway's DR11-W driver for the host is a drop-in assembly language module that controls the host interface. The Entryway operates as a high-speed peripheral device to a DEC system.

Interface Command. The host issues commands to the Entryway interface by placing a command code on the FNCT lines and issuing a GO pulse. The host loads the Control and Status Register (CSR) of the DR11-W with a value containing the proper FNCT code with GO held false. The host then rewrites the register, setting the GO bit along with the FNCT code. The command codes are listed in Table 11-6. The rewriting of the register allows the FNCT control lines to synchronize with the GO pulse.

Interface commands are terminated by an interrupt request when the Entryway interface pulses ATN. ATN causes READY to go true on the DR11-W. An interrupt occurs when the IE bit of the CSR is set.

Initialization and Idling. Put the EDLS/DR11 interface in a known state as part of driver initialization. Accomplish this either by pulsing INIT on the DR11-W or issuing a RESET command. The RESET command can be issued while a previous command is pending completion. RESET is the only command that should be issued when command completion is pending.

While the EDLS/DR11 interface driver is waiting for an event on the net, it should issue an IDLE command to the Entryway, and then, wait for completion. For example, an event could be the arrival of a message or the clearing of a flow control state.

A driver forcibly terminates the IDLE command by issuing a RESET command and waiting for completion. The Input Data Register (IDR) is then sampled and the 0 bit tested. When this bit is set, RESET has been completed and the IDLE command has been aborted. When this bit is reset, the IDLE command is terminated before execution of RESET. When this is the case, the host should wait for RESET to complete the sampling of the

IDR. The interface is then ready to accept data transfer commands.

Host-to-Entryway Frames. Before sending a frame to a port or a process on the network, the host should first check the network interface to see whether it is ready for data. STATUS C in the CSR is set when the network interface is ready. The host loads the Word Count Register (WCR) with a value equal to the frame word count plus an additional word. The additional word count protects against loss of interface sychronization.

Next, the host loads the Bus Address Register (BAR) with the low-order 16 bits of the Unibus address of the frame buffer, and issues the OUTPUT FRAME command to the interface. The Entryway terminates the OUTPUT FRAME command when the transfer is complete.

Error termination occurs when the specified frame length exceeds 1520 bytes, or when the frame is output and the network interface is not ready. A word count register overflow indicates loss of synchronization between the DR11-W and Entryway. To clear this, the host issues a RESET command.

Following a successful transfer, the host examines the STATUS C bit of the CSR to see whether the interface is ready for another frame.

Entryway-to-Host Frames. EDLS/DR11 supports two mechanisms for frame input. The first mechanism inputs the frame in two parts: first the header, then the rest of the frame. The second mechanism inputs the entire frame all at once. Table 11-6 shows the frame input commands for EDLS/DR11.

With the first mechanism, the driver sets the STATUS B bit in the CSR to indicate that a frame is available at the interface. The driver loads the WCR with the length of the preview portion of the frame, in 16-bit words, plus one additional word. The BAR is loaded with the address of a preview buffer, and a PREVIEW FRAME command is issued. The Entryway terminates this command when the preview portion of the frame is transferred to the preview buffer.

The driver examines the bytes returned and determines whether the frame is a Data Frame or a Command Frame. When the frame is a Command Frame, the driver does not need to issue a

READ FRAME command because the PREVIEW FRAME command returns the entire frame.

When the frame is a Data Frame, the host software determines its origin, destination address mode, and destination E-type. When the frame is of no interest to the host, the host issues a FLUSH FRAME command to discard the remainder of the frame.

When the host driver wishes to read the data field, it loads the WCR with the word count of the data field length plus one additional word. When the data buffer is not contiguous to the header buffer, the host also loads the BAR with the data buffer addresss. The READ FRAME command causes the transfer of the data field. When a Data Frame is received with no data field following the header, it is still necessary to issue a FLUSH FRAME or READ FRAME command to free the Entryway's frame buffer.

The second mechanism is simpler. It is used in installations where the following conditions are met:

- All frames are of interest to the same low-level host process.
- All arriving frames are less than or equal to the maximum size of the input buffer.
- There is no need to separate headers from the rest of the frame.

With these conditions met, program the WCR with the maximum buffer length plug one additional word, and program the BAR for the address for buffer. Issue the READ FRAME command to start the transfer. The Entryway signals termination when the transfer is complete. The frame length field is used to derive the number of bytes actually transferred.

Error termination occurs in both of the above mechanisms under the following conditions:

- No frame exists at the interface when the PREVIEW FRAME or READ FRAME commands are issued.
- A WCR overflow condition indicates loss of synchronization.

Note that when a WCR overflow occurs, an extra, incorrect word will have been transferred across the interface. In your program design you should ensure that the memory word following each header or input buffer can be modified without damaging the host system.

A WCR overflow results when the second frame input mechanism transfers a frame whose lengthis greater than the maximum buffer size of the host. The host must issue a FLUSH FRAME command following the RESET to remove the oversized frame from the queue.

RESET HANDSHAKE PROTOCOL

As a result of setting the Receiver Control Mode and E-type parameters, the host initializes the Entryway to a known state. Whenever a host or its Entryway is initialized, the host and Entryway must establish or re-establish this state. The host must implement the following algorithm for this protocol.

```
PowerUP:
take host restart actions:

xmt_flag := true;

LOOP
    if xmt_flag then
      send 'clear' frame
      start rexmt timer
      xmt_flag := false
    fi;

    recv frame f
    CASE f OF
      'cleared': EXIT
      'entryway_reset'           : send 'entryway_reset_seen' frame
       otherwise                 : discard
    ENDCASE

    If rexmt timer expired then
      xmt_flag := true
    fi

REPEAT
Normal:
LOOP

    ....
    recv frame f
    CASE f OF
      ....
      'entryway_reset':           send 'entryway_reset_seen' frame
                                  take entryway_reset actions;

      ....
```

The following algorithm is the one the Entryway uses to implement its part of the protocol. The Entryway algorithm is given only to make the host algorithm easier to understand in the discussion below. The host driver only needs to implement the host algorithm; the Ethernet Data Link Service contains the Entryway algorithm.

```
PowerUP:
take entryway restart actions;

xmt_flag := true;

LOOP
    if xmt_flag then
      send 'entryway_reset' frame;
      start rexmt timer
      xmt_flag := false
    fi;

    recv frame f

CASE f OF
      'entryway_reset_seen'          : EXIT
      'clear'                        : take entryway clear actions;

      otherwise                      : discard
    ENDCASE

    If rexmt timer expired then
      xmt_flag := true
    fi

REPEAT

Normal:

LOOP
    ....
    recv frame f
    CASE f OF
      ....
      'clear'  : take entryway clear actions;
                 send 'cleared' frame
      ....
```

Notice that each algorithm has two sections: PowerUp and Normal. The PowerUp loops show what should happen when the host or the Entryway is reset; the Normal loops show what happens when the host or Entryway is running when the other is reset.

When a host goes through a power-up sequence, it should register the E-type it wishes to use, and establish the Receiver Control Mode it wants. Before it issues these commands, it should issue the Clear command and wait for the Entryway response to make sure that the Entryway has no E-type or Receiver Control Mode currently registered for that host.

The host sets the transmit flag (xmt_flag) to false and waits for either a Cleared or an Entryway_Reset command. Set the response timer for about 10 seconds. This value will probably prevent the host from mistaking a slow response for no response.

Once the host receives a Cleared command, it can proceed with normal operation. However, if the Entryway is going through a power-up sequence at the same time as the host, the host receives an Entryway_Reset command and must respond with an Entryway_Reset_Seen command before starting normal operation. If the host does not receive either the Cleared or the Entryway_Reset command before the timer expires, it retransmits the Clear command, and waits again for a response.

The Normal section of the host algorithm is necessary when the Entryway must be reset after the initial power-up of the Entryway and the host. When this occurs, the host receives the Entryway_Reset command and must respond with the Entryway_Reset_Seen command. The host should then send the Clear command, and reregister for E-type and Receiver Control mode.

The Entryway algorithm is essentially the same as the host algorithm. When going through a power-up sequence, the Entryway must inform the host that it has been reset for acknowledgement of this message. When the host is already operational, it acknowledges the message and reregisters its E-type and receiver control. When the host is also going through a power-up sequence, the Entryway receives a Clear command and responds with a Cleared command.

Chapter 12
Datagram Service

Contents

OVERVIEW

The Codex LAN Datagram Service provides a simple host-to-host communication protocol via the Network Layer for Codex Local Area Networks (LAN), including those interconnected by Codex LAN bridges. The Datagram Service allows a user program running on a host computer to send information to other user programs running on any or all hosts on the local network or internet without having to establish a virtual circuit connection. This service is intended for user programs that run on a host.

The Datagram Service software runs inside a Codex LAN Entryway. Each Entryway running the Datagram Service runs its own copy of the software. There are two versions of the Datagram Service software: Single-port Datagram Service (SDS) and Multiport Datagram Service. (MDS). Each version supports a subset of the various host-Entryway communications interfaces that the Datagram Service supports. Although the two versions of the software support different communications interfaces, the Datagram Service allows hosts using SDS to communicate with hosts using MDS and vice versa.

The Datagram Service carries data via the Network Layer for user programs running on attached hosts. In addition to carrying data for a host program, the Datagram Service responds to requests from a local host program for information necessary to manage the data transfer operation, such as the host's network address.

The data that the Datagram Service carries across the network for its host clients is formatted into units called datagrams. The Datagram Service assures best-effort delivery of datagrams for its clients, but cannot guarantee delivery. The Datagram Service also cannot guarantee that a series of datagrams will be delivered in the order sent, or that occasionally a datagram will not contain corrupted data. This service is meant for user applications that can either tolerate some errors, or employ their own error recovery protocols.

EXAMPLE OF DATA FLOW

Figure 12-1 illlustrates data flow between two host computers (Host A and Host B), both connected to Entryways running the Datagram Service software. Although a message (indicated by colored lines) travels from the user program on one host computer to the user program on the other, a program directly exchanges data only with the local Datagram Service. The local Datagram Service handles all necessary network communications.

Before the user program on Host A can send data to the user program on Host B using the Datagram Service, the program on Host A should perform certain administrative operations. The figure shows only two of these operations, the host inquiry, WHO AM I?, and the Datagram Service response, YOU ARE, to illustrate the exchange of control information between a user program and the local Datagram Service. All of the available operations are described later in this chapter.

In the figure, the user program running on Host A requests its identification from the local Datagram Service. After receiving the response, the user program sends a message to the user program running on Host B.

The message first goes to the Datagram Service running on the Entryway attached to Host A. Notice that the Datagram Service does not send an acknowledgement to the host program when it receives the message. It is up to the user program to detect and handle any errors that occur.

When the Datagram Service running on the Entryway attached to Host A receives the message, it formats and sends the message to the Datagram Service running on the Entryway attached to Host B. Finally, the Datagram Service on the Entryway attached to Host B passes the message to the user program running on Host B.

After receiving the message from Host A, the program running on Host B sends a message to the program running on Host A. This message first goes to the local Datagram Service, then over the network to the Datagram Service running on the Entryway attached to Host A, and finally, to the user program running on Host A.

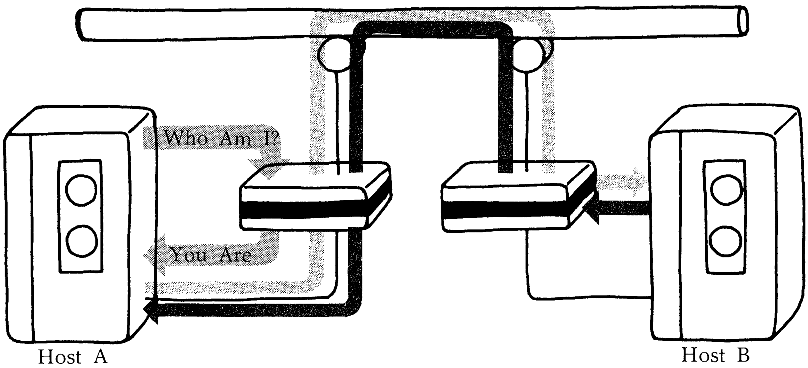


Figure 12-1. Flow of Data and Control Information

As you can see, the Datagram Service on each Entryway is a middle man for a user program running on an attached host. It communicates with the other Entryways on the network running the Datagram Service in order to send and receive the datagrams the hosts wish to exchange. The local Datagram Service also provides the administrative information and services the user program needs to manage the host-to-host data transfer.

COMMUNICATIONS INTERFACES SUPPORTED

A variety of standard communications interfaces are provided for the Datagram Service in order to support the different host-Entryway communications interfaces. The communications interfaces supported include two serial asynchronous interfaces, an IEEE-488 high-speed interface, an 8-bit parallel interface, and a 16-bit Digital Equipment Corporation (DEC) DR11-W parallel interface. Each host attached to an Entryway requires a particular interface. In order to use the Datagram Service, an Entryway must be configured with the Codex LAN I/O module that provides the physical interface that the host requires.

The Codex LAN IEEE-4888 and DR11-W/B interfaces use the single-port version of the Datagram Service software (SDS). SDS is a high-speed block-oriented service. A block-oriented driver handles data in block units rather than byte by byte.

The Codex LAN Six Serial and Serial/Parallel I/O modules use the multiport version of the Datagram Service software (MDS). MDS constructs a block structure on top of a stream-oriented interface. A stream-oriented driver handles data in asynchronous byte streams.

USING THE DATAGRAM SERVICE

Before a host program can use the Datagram Service, the attached Entryway must be configured to run the service. You define the characteristics of an Entryway with the Configure program.

When you use the Datagram Service, you must provide the host driver interface for the hardware. This driver formats, sends, and receives information for host programs that use the Datagram Service.

The communications interface between a host and an Entryway determines the method for transferring data and control information between a user program and the Datagram Service. You must be familiar with this interface in order to write the required driver. Although this manual does provide information about using the supported communications interfaces with the Datagram Service, it does not provide a detailed explanation of each interface.

A program using the Datagram Service must format the information that it sends to the service. The information formats that the Datagram Service recognizes are described next.

FRAME FORMATS AND CONTENTS

The data and information exchanged by a user program and the Datagram Service are sent in structures called frames. A frame that contains a datagram is called a Data Frame, and a frame that contains control information is called a Command Frame. Depending on the type of frame, a frame consists of a header followed by addressing information and data, or a header followed by command parameters.

This section defines the formats of the two frame types and explains the commands that the host and the Datagram Service use to pass the necessary control information to each other.

This section also explains how to address Data Frames and how host programs determine what type of data is in a frame. To send a Data Frame, a host program needs to supply the Datagram Service with its 10-byte network address and the 10-byte network address of the destination host program. The Datagram Service provides the source address for a host program. However, a host program sending a Data Frame must know through administrative means the address of any destination host program.

The address portion of a Data Frame includes a field called in Internet Packet Type (IP-type). The IP-type characterizes the type of data in the datagram.

Before a host program can receive Data Frames, it must register a range of IP-types with the Datagram Service. The Datagram Service delivers a Data Frame to a host program only if the Destination IP-type of the frame is within the IP-type range or ranges that the host program has registered. The section on the Command Frame explains how to register IP-types.

Frame Header

Both Data Frames and Command Frames begin with a 6-byte Frame Header. The Frame Header provides information, such as the length of the frame, the type of frame and the purpose of the frame. Figure 12-2 shows the format of this header, and Table 12-1 explains each of the header fields. Note that the most significant byte of all multibyte numeric fields is transmitted first.

To evaluate a checksum, first convert each byte to an 8-bit binary quantity. Then, find BYTE1 XOR BYTE2. In the example in the table, BYTE1 is 00 and BYTE2 is 1E. Next, find RESULT1 XOR BYTE3, where RESULT1 is the result of the previous operation. Continue computing the exclusive OR of the previous result and the next byte until you reach the last byte of the equation. The result of the exclusive OR of the last byte and the previous result is the value of the checksum.

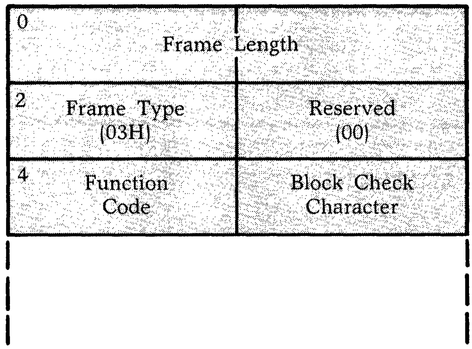


Figure 12-2. Datagram Service Frame Header

Table 12-1.
Frame Header Field Description

Bytes	***Description***
0-1	Length of the frame, including the header, in hexadecimal. The most significant byte of the value is Byte 0; Byte 1 is the lease significant byte of the value.
2	Identifies the client receiving the frame. This byte must contain 03 (hex) for all Datagram Service frames.
3	Reserved by the system; must be 00 (hex).
4	Specifies the purpose of a frame. The code 04 (hex), for example, indicates that a host is requesting its network address. Table 12-2 lists all the codes and meanings.
5	The results of a bytewise exclusive OR (XOR) of the first five bytes of the Frame Header. For example, the checksum of the header of a frame 30 bytes long (1E hex) with a Function Code of 01 is 1C, computed as follows: 00 XOR 1E XOR 03 XOR 00 XOR 01 = 1C

The host must compute the checksum and insert it in Byte 5 when sending a frame to the Datagram Service. The Datagram Service also computes the checksum of each frame it receives from a host, and verifies that the exclusive OR of the six bytes of the header is 0. When this is the case, the checksum is correct. If the checksum is not correct, the Datagram Service discards the frame. When a host receives a frame from the Datagram Service, it should also verify that the checksum is correct, and discard any incorrect frames.

Table 12-2 lists all the Datagram Service Function Codes and their meanings. The two Here Is Data codes identify a frame as a Data Frame. The remaining codes are the commands the host and Entryway use to manage the exchange of Data Frames. The functions are explained in more detail in the following sections on the Data Frame and the Command Frame.

Table 12-2.
Datagram Service Function Codes

Code	*Function*	*Description*
00	Here is Data (Host to Entryway)	Indicates that the host is passing data to the Entryway to send to another host.
01	Here Is Data (Entryway to Host)	Indicates that the Entryway is passing data to the host from another host.
02	NOP Request (Host to Entryway)	A host can send this command to find out whether its Entryway is operational.
03	NOP Response (Entryway to Host)	The Entryway response to the NOP request that informs a host that the Entryway is operational.

Table 12-2 (continued).
Datagram Service Function Codes

Code	*Function*	*Description*
04	Who Am I (Host to Entryway)	Requests the host's 10-byte network address.
05	You Are (Entryway to Host)	Gives a host its 10-byte network address.
06	Get IP-Type Response (Entryway to Host)	A host request to register an IP-type range.
07	Get IP-Type Response (Entryway to Host)	Registers an IP-type range for a host, or indicates that the operation failed.
08	Return IP-Type (Host to Entryway)	A host request to free a registered IP-type range.
09	Return IP-Type Response (Entryway to Host)	Returns the value of the range the host wanted to free, and indicates whether the operation succeeded or failed.
0A	Entryway Reset (Entryway to Host)	Informs host that Entryway was reset.
0B	Entryway Reset Seen (Host to Entryway)	Host acknowledgement of message that Entryway was reset.
0C	Clear Entryway (Host to Entryway)	A host request to the Entryway to clear all registered IP-type ranges.
0D	Entryway Cleared (Entryway to Host)	Informs host that Entryway has cleared all registered IP-type ranges.

Data Frame

A Data Frame consists of the Frame Header, addressing information, and user data. The addressing information allows a host program to address a datagram to any other such program on the local network, or on a remote network.

As mentioned previously, the addressing information includes a field called an IP-type. Each Data Frame has two of these fields: a Source IP-type and a Destination IP-type. The first byte of this field contains one of the valid IP-type range numbers. The second byte is a field available to the user for datagram demultiplexing.

The second byte of an IP-type can contain any value from 00 to FF. The Datagram Service ignores this byte when sending or receiving Data Frames. However, not that some Datagram Service Commands require the second byte of an IP-type field to be zero.

Figure 12-3 shows the format of a Data Frame. Table 12-3 explains each of the fields.

The Network ID is a 2-byte quantity identifying the local network to which the host's Entryway is attached. The meanings of the possible values for this field are as follows:

- 00 00 (hex) always indicates the local network of the source host
- FFFF (hex) indicates all networks
- The range 1-7FFF (hex) indicates a specific network
- The range 8000-FFFE (hex) is reserved by the system

Network IDs are assigned by each installation's Network Administrator.

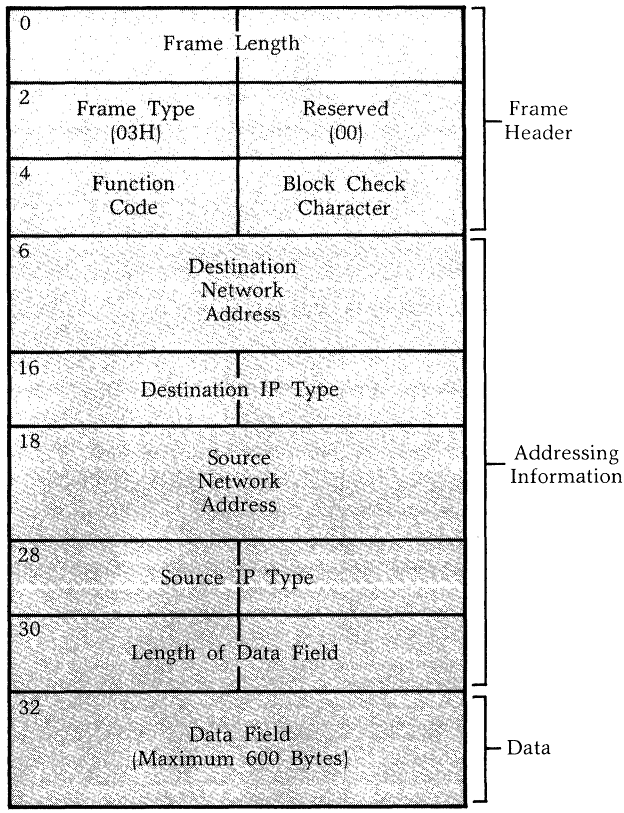


Figure 12-3. Data Frame Format

Table 12-3.
Data Frame Field Descriptions

Field (bytes)	*Description*
Length (0-1)	The length of the frame, in binary, as a byte count including the header.
Frame Type (2)	This field must be 03 (hex) for the Datagram Service.
Reserved Byte (3)	Reserved by the system; must be 00 (hex).
Function Code (4)	Defines the contents of the frame as data. This byte contains the value 00 (hex) if the host is sending the data to the Entryway; the byte contains the value 01 (hex) if the Entryway is sending data to the host.
Block Check Character (5)	The block check character is the 8-bit result of a bytewise exclusive OR (XOR) of the first five bytes of the header.
Destination Network Address (6-15)	The 10-byte address of the destination host consisting of: Bytes — Contents 0-1 — Reserved; must be 00 (hex) 2-3 — Network ID (defined below) 4-9 — Host ID (defined below)
Destination IP-Type (16-17)	The Internet Packet Type (IP-type) of a frame identifies the type of data in a datagram.
Source Network Address (18-27)	The 10-byte address of the source host (see Destination Network Address above).

Table 12-3 (continued).
Data Frame Field Descriptions

Field (bytes)	*Description*
Source IP-Type (28-29)	The source Internet Packet type (IP-type).
Byte Length of Data Field (30-31)	Length of the data being sent or received.
Data Field (32-end)	Any data format is allowed in this field. Minimum length of data = 0; maximum length = 600 bytes.

The Host ID, the Ethernet-compatible identification number of the host, consists of:

Bytes	*Contents*
4-6	The manufacturer identifier. For Codex LAN equipment, these bytes must be 00 DD 00 for host-to-host addressing, and 01 DD 00 for multicast addressing (see below).
7-8	the Entryway node number, in hexadecimal. The most significant byte is first; the least significant byte is second. For example, Entryway #55 is 00 37 hex.
9	The host port and board number within an Entryway. The upper four bits are the port number. Single-port (SDS) is 1, multiport (MDS) can range from 1 to 6. The lower four bits are the board number (Board A=0, D=3).

The byte numbers above refer to the displacements within the address fields shown in Table 12-3.

Data Frame Addressing. The Datagram Service uses the three forms of addressing listed below.

- ***Host-to-Host.*** The destination address specifies a single host.

- ***Broadcast.*** The destination address specifies all hosts. Set all six bytes of the host ID to FF (hex) to send a broadcast message on the specified local network. If your network has a bridge connection to one or more networks and you want to send a message to all hosts on all networks, also set both bytes of the network ID to FF (hex).

  It is possible to combine specific and broadcast values in a destination address. In fact, any of the destination address fields (network ID, Entryway, board, or port) can contain a broadcast value or a specific value. This capability allows you, for example, to send a message to all ports on Board B of Entryway 213, all ports on Board A in all Entryways, or all ports on all boards of Entryway 57, etc. Set all bytes of an address field to FF (hex) to specify a broadcast value for a field.

- ***Multicast.*** The destination address specifies a specific group of clients rather than a subset of Entryways, boards, or ports. To send a multicast message, set the fixed bytes of the host identification to 01 DD 00, and set the port bits to FF. Although all hosts receive a multicast message, each host can use the remaining bits of the host ID to determine whether or not it is included in the group of intended receivers.

Addressing Examples. The following examples show how to use the various addressing modes. For brevity, only the 10-byte Destination Network Address field of the header and the 2-byte Destination IP-type field are given (in hexadecimal notation). The twelve bytes given in each example are as follows:

- Two reserved bytes, which must be set to 00 (hex)
- 2-byte destination network address
- 6-byte host ID
- 2-byte destination IP-type

Examples:

- The host sends a datagram to Entryway #18, Board B, Port 2 on the same network:

Send:

00	00	00	00	00	DD	00	00	12	21	0C	15

00 00	= Reserved bytes
00 00	= Destination network
00 DD 00	= Fixed bytes of host ID
00 12	= Entryway #18
21	= Destination port/board
	2 = Port 2
	1 = Board B
0C 15	= Destination IP-type

- The host sends a broadcast message to every Entryway, board, and port on network 1:

Send:

00	00	00	01	FF	FF	FF	FF	FF	FF	0C	25

00 00	= Reserved bytes
00 01	= Destination network
FF FF FF	= Fixed bytes of host ID
FF FF	= Destination Entryway (every Entryway on the network)
FF	= Every port and board
0C 25	= Destination IP-type

- The host sends a message to all ports and boards on Entryway #20 of network 58 (3A hex):

Send: | 00 | 00 | 00 | 3A | 00 | DD | 00 | 00 | 14 | FF | 0D | 11 |

00 00	= Reserved bytes
00 3A	= Destination network
00 DD 00	= Fixed bytes of host ID
00 14	= Entryway #20
FF	= Every port and board
0D11	= Destination IP-type

- The host sends a message to all ports on every Board A on all networks:

Send: | 00 | 00 | FF | FF | FF | FF | FF | FF | FF | F0 | 0D | 22 |

00 00	= Reserved bytes
FF FF	= Destination network (every network connected by bridge to the local network)
FF FF FF	= Fixed bytes of host ID
FF FF	= Destination Entryway (every Entryway on each network)
F0	= Every port, Board A only
0D22	= Destination IP-type

- The host sends a multicast message to user-defined Group #2 on network 12 (0C hex):

Send:

00	00	00	0C	01	DD	00	00	02	FF	0C	10

00 00	= Reserved bytes
00 0C	= Destination network
01 DD 00	= Fixed bytes of host ID (multicast bit set)
00 02	= User-defined desgnation for Group #2
FF	= Every port and board
0C 10	= Destination IP-type

This message is sent, with the destination address preserved, to every Entryway on the local network. The receiving hosts decide whether or not to accept it, based on the multicast group to which they belong.

The following chart shows a complete Datagram Service datagram sent from a user program running on a host attached to Entryway 18 of network 10 at Board D, Port 1. The message the program is sending is HELLO. The program the datagram is addressed to runs on a host that is attached to Entryway 20 of network 12 at Board A, Port 1.

Byte	*Contents*	*Description*
0	00	Length of header and data
1	25	(37 decimal)
2	03	Frame Type
3	00	Reserved by system; must be 00
4	00	Indicates data is sent from host to Entryway
5	26	Block check character
6	00	Reserved bytes of Destination
7	00	Network Address

Byte	*Contents*	*Description*
8	00	Destination network ID (0C hex, 12 decimal)
9	0C	
10	00	Defined bytes of destination
11	DD	host ID
12	00	
13	00	Destination Entryway (14 hex,
14	14	20 decimal)
15	10	Port 1, Board A
16	0C	Destination IP-type
17	22	
18	00	Reserved bytes of Source
19	00	Network Address
20	00	Source network ID (0A hex,
21	0A	10 decimal)
22	00	Defined bytes of source host ID
23	DD	
24	00	
25	00	Source Entryway (12 hex,
26	12	18 decimal)
27	14	Port 1, Board D
28	0C	Source IP-type
29	00	
30	00	Length of data field
31	05	
32	48	H
33	45	E
34	4C	L — 5 bytes of data in ASCII
35	4C	L
36	4F	O

Offset		
0	Frame Length	
2	Frame Type (03H)	Reserved (00)
4	Function Code	Block Check Character
6	Parameters (When Necessary)	

Figure 12-4. Command Frame Format

Command Frame

A Command Frame consists of the 6-byte Frame Header, defined earlier, and command parameters, such as IP-type. Figure 12-4 shows the format of the Command Frame.

Datagram Service Commands

Datagram Service commands convey the control information needed for the orderly operation of the Datagram Service. The commands are divided into pairs that consist of a request and a response. These command pairs allow a host program to get information from the Datagram Service and allow the Datagram Service to alert a host program when certain events occur.

The order in which the pairs are presented follows the logical order of use. The NOP and NOP Response commands are first, the host identification commands are second, the commands that get and return IP-types are third, and the four commands necessary to the Reset Handshake Protocols end the section.

The definitions of the commands that follow specify all parameters necessary for each command, their length, and their order within the frame. If a command does not require any parameters, the Command Frame consists only of the Frame Header.

NOP, NOP Response. A host program sends the NOP request to determine if its Entryway is operational. The NOP Response is confirmation that the Entryway is operational.

Function Codes:	NOP	02 (hex)
	NOP Response	03 (hex)
Parameters:	none.	
Usage Notes:	When the Entryway sends the NOP response, the host program knows that it can use the Datagram Service; if the host receives no response to the NOP request, it knows hat something is wrong with the Entryway.	

Who Am I? Requests the 10-byte Codex LAN address of the host.

Function Code:	Who Am I	04 (hex)
Parameters:	none.	
Usage Notes:	Unless you know the 10-byte Codex LAN address of the host by administrative means, you must use this command to find out this address. Note that the address includes the Network ID assigned to your segment of the internet if your local network is part of an internet.	

You Are. Returns the 10-byte Codex LAN address of the host.

Function Code:	You Are	05 (hex)
Parameters:	***Bytes***	***Description***
	6-15	The 10-byte Codex LAN address of the host. Refer to the section on addressing for details of the address format.
Usage Notes:	The 10-byte address that this command returns includes the network ID assigned to the local network by the Network Administrator. The assigned network ID must be used when the source and destination hosts of a Data Frame are attached to Entryways connected to different segments of an internet.	

Get IP-Type. Requests an IP-type range for a host.

Function Code:	Get IP-type	06 (hex)
Parameters:	***Bytes***	***Description***
	6-7	Requested IP-type range.
Usage Notes:	The command Get IP-type registers the requesting host for the requested IP-type range, if possible. Once a host is registered, the Entryway sends the host all frames addressed to the host that have a destination IP-type within the registered range. The IP-type is a 2-byte quantity. When requesting an IP-type range, the first byte of the IP-type parameter should be the high-order byte of the range and second byte should be 00. For example, a parameter of 0C00 request all IP-types in the range 0C00 through 0CFF. The valid ranges for this quantity are as follows:	

01xx	Reserved for Codex LAN protocols
02xx	Reserved for Codex LAN protocols
•	
•	
•	
0Bxx	Reserved for Codex LAN protocols
0Cxx	Available for Datagram Service users
0Dxx	Available for Datagram Service users

When the host program registers a range of IP-types using the Get IP-type command, only the first byte is significant. Although the second byte is not significant, it must be zero. Note that an Entryway running the Multiport Datagram Service registers IP-types separately for each host attached to the processor board.

Although it is not required, host programs should request IP-type ranges from the ranges reserved for Datagram Service users. As long as no client of the Network Layer within the Entryway processor has registered an IP-type range, a host program can register IP-type ranges reserved for Codex LAN protocols.

The IP-type FF00 (hex) is a special value used with the Get IP-type command. If a host program registers this IP-type, it will receive datagrams with IP-types in the range 01xx to 0Dxx that are not registered for use by other Codex LAN services withing the Entryway processor. Note that certain packets broadcast by Codex LAN services, such as download requests, may also be delivered to a program that registers for this IP-type. This is most likely to occur when the Datagram Service is running on a board other than Board A.

Get IP-Type Response. Acknowledges registration of an IP-type range.

Function Code:	Get IP-type Response	07 (hex)
Parameters:	***Bytes***	***Description***
	6-7	Request IP-type range.
Usage Notes:	When the operation is successful, the frame contains the requested IP-type range. When the operation is not successful, bytes 6 and 7 contain 0000. Note that when a host is already registered for the requested IP-type range, this command returns the IP-type range and not the error indicator.	

Return IP-Type. Frees the indicated IP-type.

Function Code:	Return IP-type	08 (hex)
Parameters:	***Bytes***	***Description***
	6-7	IP-type range being returned. The second byte must be 00. A value of 00 00 requests that the Entryway free all ranges assigned to the host.
Usage Notes:	The Return IP-type command tells the Entryway that an attached host no longer wishes to receive frames with Destination IP-types within a certain range. Sending 00 00 in the IP-type field of the frame asks the Entryway to free all ranges assigned to the host.	

Return IP-Type Response. Acknowledges which IP-type range was returned and indicates the success or failure of the operation.

Function Code:	Return IP-type Response		09 (hex)
Parameters:	***Bytes***	***Description***	
	6-7	Returned IP-type range.	
	8	Return Code; 00 indicates failure, and FF (hex) indicates success.	
	9	Reserved; must be 00 (hex).	
Usage Notes:	When the IP-type range is free prior to the Return IP-type request, the return code indicates a successful operation. This avoids host confusion that would occur if the response to the original request were lost.		

Entryway_Reset, Entryway_Reset_Seen, Clear_Entryway, Entryway_Cleared. These commands are part of the Reset Handshake protocol that a host and its Entryway use when one of the two comes up after a reset operation. The Reset Handshake Protocol ensures the integrity of the list of IP-types registered to the host.

Function Codes:	Entryway_Reset	0A (hex)
	Entryway_Reset_Seen	0B (hex)
	Clear_Entryway	0C (hex)
	Entryway_Cleared	0D (hex)
Parameters:	none.	
Usage Notes:	The host and Entryway algorithms for the Reset Handshake protocol are described at the end of this chapter. These four commands are part of these two algorithms.	

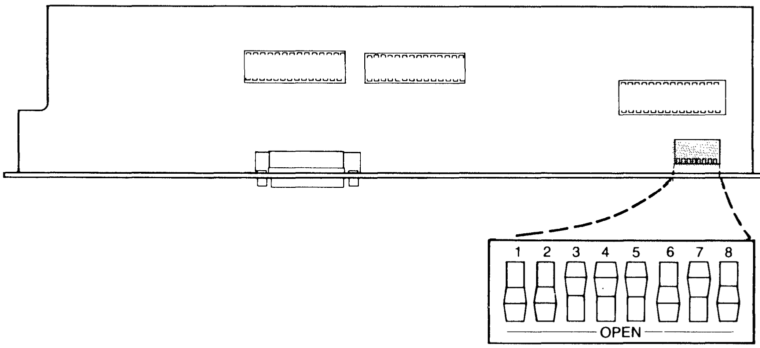


Figure 12-5. IEEE-488 I/O Module Address Sense Switch Location

IEEE-488 DATAGRAM SERVICE INTERFACE

This section provides information on sending and receiving Datagram Service frames for users with programmable IEEE-488 interfaces. You should have some knowledge of the IEEE-488 General Purpose Instrumentation Bus (GPIB) before programming an application that uses this interface to the Datagram Service.

Hardware Requirements

The IEEE-488 I/O Module and Processor Board combination must be installed in the Entryway to use this interface to the Datagram Service. The GPIB address for the Entryway and the parallel poll response bit assigned to the Entryway are both set by an address sense switch located on the I/O Module. Refer to Figure 12-5 for the location of this switch. ■ **Note:** The sense switch is located as shown in Figure 12-5. Do not confuse this with the processor board sense switch.

The address sense switch is an 8-bit DIP switch. The bit positions on the switch are zero when placed to OFF, and one when placed to ON. Bit positions 1 through 5 on the address sense switch correspond to the GPIB bus addresses 0 through 31. Bit positions 6, 7, and 8 on the address sense switch define the parallel poll response bit (0-7) that the Entryway will assert when the controller conducts a parallel poll.

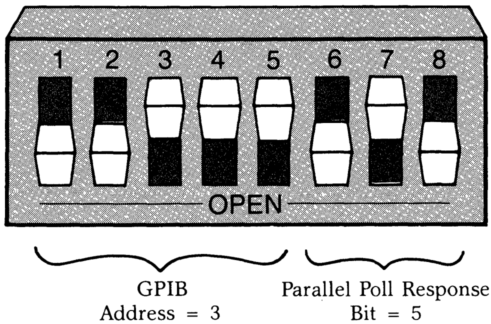


Figure 12-6. Address Sense Switch Setting Example

Figure 12-6 shows an example address sense switch setting. Switch 1 on the DIP switch sets the least significant bit of the Bus Address. Switch 6 sets the least significant bit of the parallel poll response bit.

Software Requirements

Use the CONFIGURE program to configure the Entryway for the Datagram Service software. The host computer must implement all frame construction, sequencing, acknowledgement, and error recovery logic.

Using the Datagram Service with the IEEE-488 Interface

SDS.488 is a software module that allows the Entryway to act as an IEEE-488 peripheral to the host. For the purposes of the information in this section, it is assumed that the host program will use an IEEE-488 driver that provides some degree of standard control over an IEEE-488 I/O port.

Host-to-Entryway Frames. When the host has a frame to send, it should conduct a parallel poll on the IEEE-488 interface to verify that the Entryway is operational and has an input buffer available. When the parallel poll response bit is present, the Entryway is ready for data.

Once the host determines that the IEEE-488 interface is ready, the host or controller should first assert ATN (Attention), and then UNTALK any devices that might have data gated onto the GPIB. The host/controller should then UNLISTEN any devices for whom the frame is not intended. The host/controller then issues the LISTEN ADDRESS of the Entryway.

When the host is not the GPIB controller, the controller should issue the TALK ADDRESS of the host before ATN is released. When the host is the controller, it should enter the TALK-ONLY state and release ATN. The host can then transfer a frame across the GPIB to the Entryway.

SDS/488 assumes that the first two bytes transferred are the frame length field. SDS/488 reads the length field and attempts to input the corresponding number of bytes. Frames with lengths less than 6 bytes or greater than 632 bytes, are discarded; 6 and 632 are the lower and upper limits, respectively, of a frame. Any frame transfer is aborted by asserting IFC (Interface Clear) or UNLISTEN.

Entryway-to-Host Frames. When SDS/488 receives a frame from the net to transfer to the host, SDS/488 requests service from the host/controller by asserting SRQ (Service Request) on the GPIB. If devices other than the host and Entryway are attached to the GPIB, or if the host is not the controller, it may be necessary for the host/controller to conduct a serial poll to verify that the Entryway is requesting service. A parallel poll cannot be used, because the parallel poll response bit is used as the ready-for-data indicator.

Once the host/controller establishes that the Entryway is requesting service, the host/controller should assert ATN. The host/controller should then UNTALK any devices that might have data gated onto the GPIB, UNLISTEN any devices for whom the frame is not intended, and issue the TALK address of the Entryway.

When the host is not the GPIB controller, the LISTEN ADDRESS of the host should be issued. When the host is the GPIB controller, it should enter a LISTEN ONLY state before releasing ATN.

The Entryway transfers the frame across the GPIB, beginning with the length field. Note that the two bytes of the length field are included in the frame length.

If the host/controller issues a TALK ADDRESS of the Entryway when no frame is present, then a null frame consisting of a length field of 00 02 hex is transferred. The last byte of the frame transferred by SDS/488 is accompanied by the EOI (End of Identify) signal to indicate completion. Any frame transfer is aborted by asserting IFC or UNTALK. Both IFC and UNTALK return the Entryway to the talker idle state and reassert SRQ.

SDS/488 Functionality. Table 12-4 describes the subset of the IEEE-488/1978 standard implemented by SDS/488.

DR11-W DATAGRAM SERVICE INTERFACE

SDS/DR11 interface provides access to the Codex LAN Datagram Service for Digital Equipment Corporation (DEC) computers equipped with DR11-W Unibus DMA controllers, or other machines equipped with a DR11-W interface. SDS/DR11 also supports the DR11-B interface. All references to DR11-W include DR11-B unless otherwise noted.

This section should help you generate a host system I/O driver for SDS/DR11 interface to Codex LAN. It assumes that you have some knowledge of both the DEC DR11-W DMA controller and the single-port version of the Codex LAN Datagram Service.

Hardware Requirements

The 32-bit Parallel I/O Module and Processor Board combination must be installed in the Entryway to use the DR11-W Datagram Service. Communication between the host and Entryway occurs over the host's DR11 cables. A DR11 Interface Card attaches at the rear panel of the Entryway between the I/O ports of the 32-Bit Parallel I/O Module and DR11 cable. Refer to the Codex 4000 Series LAN Hardware Reference Guide for details.

Software Requirements

Use the CONFIGURE program to configure the Entryway software for the Datagram Service. The host computer must implement all frame sequencing, acknowledgement, and error recovery logic.

SDS/DR11 Interface

This section defines the characteristics of SDS/DR11 interface. This interface allows the Entryway to operate as a high-speed peripheral device to a DEC system equipped with a DR11-W or

Table 12-4.
SDS/488 Functionality

Function	*Description*
SH1	Source handshake.
AH1	Acceptor handshake.
T6	Basic talker and serial poll mode. The lower 5 bits of the I/O Module sense switches define the GPIB interface address. SDS/488 supports serial poll mode. SDS/488 interface leaves the talk-addressed state when the listen address is issued.
TE0	Extended talker mode is not supported.
L4	Basic listener capability implemented by SDS code. The interface exits the listen-addressed state when the talk address is issued.
LE0	Extended address listener mode is not supported.
SR1	Service request function is implemented, with SRQ always signifying the availability of an output buffer for the host.
RL0	Remote/local mode setting is not supported.
PP2	Parallel poll capability with local configuration only. The upper three bits of the I/O Module sense switch define the bit used for parallel poll responses. A parallel poll response indicates that the interface is ready for input from the host. The parallel poll does not serve as a substitute for a serial poll that verifies SRQ output.
DC0	Selective device clearing is not supported.
DT0	Selective device triggering is not supported.
C0	Bus controller mode is not supported.

DR11-B module. The terms used in the following descriptions are derived from the DEC document, *DR11-W Direct Memory Interface Module User's Guide.*

Data Formatting. The Entryway has an 8-bit data path, while the DEC Unibus has a 16-bit data path. SDS/DR11 interface accepts and transmits data while preserving the order of the DEC data strings. This means that any 16-bit elements of the Frame Header must have their bytes swapped before transfer to the Entryway.

When an odd number of bytes must be transferred, SDS/DR11 transfers an additional byte to complete the last word. The additional byte has no effect on the length field of the Frame Header. SDS/DR11 discards this additional byte before transmitting the frame.

Entryway Status Indications. DR11-W has three status input lines: STATUS A, B, and C. SDS/DR11 assigns them the following meanings:

Status	*Description*
A	Error. This status is set when an invalid operation is requested by the host, and is cleared upon completion of a valid function or by RESET.
B	Frame Ready. This status is set when a datagram or command for the host is queued in the Entryway. It is reset when the host has either read or flushed all frames.
C	Network Interface Ready. This status is set when the Entryway is ready to accept a datagram or a command from the host. It is reset when no output buffers are available.

SDR/DR11 loads an additional status byte on the lower half of the interace data bus upon completion of a command or software reset. This byte is accessed by the host through the Input Data Register of the DR11-W. Bit 0 of this byte (Reset Acknowledge) is used to regain synchronization after the RESET command or the INIT pulse aborts a command. This bit is set when a RESET command terminates or when software initialization occurs, and reset when any other command terminates. All other bits of this byte are undefined.

Interrupt Requests. SDS/DR11 initiates an interrupt request by asserting the ATN signal to the DR11-W. The ATN line is pulsed high for roughly 3 microseconds for the interrupt request. Interrupt requests are issued to indicate the completion of a transfer or control function requested by the host.

Interface Control. Commands from the host are presented to SDS/DR11 by placing a 3-bit function code on the lines FNCT 1, 2, and 3. A 200-nanosecond active pulse is asserted on the GO line. Table 12-6 describes the function bits and the associated commands.

An undefined command results in an error state termination. Any valid command clears a previous error state.

Using SDS/DR11 Interface

This section describes the command sequences, data formats, and signaling conventions required for a host to exchange frames with an Entryway through the DR11-W interface. The DR11-W driver for the host is a drop-in assembly language module that controls the host interface. The Entryway operates as a high-speed peripheral device to a DEC system.

Interface Commands. The host issues commands to the Entryway interface by placing a command code on the FNCT lines and issuing a GO pulse. the host loads the Control and Status Register (CSR) of the DR11-W with a value containing the proper FNCT code with GO held false. The host then rewrites the register, setting the GO bit along with the FNCT code. The command codes are listed in Table 12-6. The rewriting of the register allows the FNCT lines time to de-skew before the GO pulse reaches the Entryway interface.

Interface commands are terminated by an interrupt request when the Entryway interface pulses ATN. This causes READY to go true on the DR11-W. An interrupt occurs when the IE bit of the CSR is set.

Table 12-6.
Function-Bit Commands for SDS/DR11

Function-Bits 3	2	1	*Command Description*
0	0	0	RESET. Halts any data transfer in progress across the interface, and takes the interface out of the active idling state if it was in that state. Sets the Reset Acknowledge bit.
0	0	1	PREVIEW FRAME. Transfers up to 32 bytes from the beginning of the frame for preview by the host. When the frame is a Data Frame, 32 bytes (6-byte Frame Header plus 26 bytes of addressing information) are transferred. When the available frame is a Command Frame, the entire frame is transferred. Generates an error condition when no frame is available.
0	1	0	OUTPUT FRAME. Allows transfer of a frame from the host into the Entryway when the Entryway is ready for data. Generates an error condition when the network interface is not ready, or when the frame has an invalid length field.
0	1	1	READ FRAME. When preceded by a PREVIEW FRAME command, the command transfers the remainder of the referenced frame into the host and clears the internal buffer. When this is not the case, the command transfers the entire frame. An error results when no frame is available.
1	0	0	FLUSH FRAME. Discards the frame at the head of the host-bound queue and clears the internal buffer. This command can be preceded by a PREVIEW FRAME command, but it is not required. An error results when no frame is available at the interface.

Table 12-6 (continued). Function-Bit Commands for SDS/DR11

Function-Bits 3	2	1	*Command Description*
1	1	0	IDLE. Terminates with an interrupt request when a frame is available for the host, or when an interface busy condition clears. If either event occurs after the last interrupt request but before the next IDLE command, the command terminates immediately. The RESET command overrides the IDLE command when you want to leave the idle state to perform a data transfer operation.

Initialization and Idling. Put SDS/DR11 interface in a known state as part of driver initialization. Accomplish this either by pulsing INIT on the DR11-W or issuing a RESET command. The RESET command can be issued while a previous command is pending completion. RESET is the only command that should be issued when command completion is pending.

While SDS/DR11 interface driver is waiting for an event on the net, it should issue an IDLE command to the Entryway and then wait for completion. For example, an event could be the arrival of a message or the clearing of a flow control state.

A driver forcibly terminates the IDLE command by issuing a RESET command and waiting for completion. The Input Data Register (IDR) is then sampled and the 0 bit tested. When this bit is set, RESET has been completed and the IDLE command has been aborted. When this bit is reset, the IDLE command terminated before execution of RESET. When this is the case, the host should wait for RESET to complete the sampling of the IDR. The interface is then ready to accept data transfer commands.

Host-to-Entryway Frames. Before sending a frame to a port or process on the net, the host should first check the network interface to see whether it is ready for data. STATUS C in the CSR is set when the network interface is ready. The host loads the Word Count Register (WCR) with a value equal to the frame word count plus an additional word. The additional word count protects against loss of interface synchronization.

Next, the host loads the Bus Address Register (BAR) with the low-order 16 bits of the Unibus address of the frame buffer, and issues the OUTPUT FRAME command to the interface. The Entryway terminates the OUTPUT FRAME command when the transfer is complete.

Error termination occurs when the specified frame length exceeds 632 bytes, or if the frame is output when the network interface is not ready. A word count register overflow indicates loss of synchronization between the DR11-W and Entryway. To clear this, the host issues a RESET command.

Following a successful transfer, the host examines the STATUS C bit of the CSR to see whether the interface is ready for another frame.

Entryway-to-Host Frames. SDS/DR11 supports two mechanisms for frame input. The first mechanism inputs the frame in two parts, the header first, followed by the rest of the frame. The second mechanism inputs the entire frame all at once. Table 12-6 shows the frame input commands for SDS/DR11.

With the first mechanism, the driver indicates that a frame is available at the interface with the STATUS B bit in the CSR. The driver loads the WCR with the length of the preview portion of the frame (up to 32 bits), in 16-bit words, plus one additional word. The BAR is loaded with the address of a preview buffer, and a PREVIEW FRAME command is issued. The Entryway terminates this command when the preview portion of the frame is transferred to the preview buffer.

The driver examines the bytes returned and determines if the frame is a Data Frame or a Command Frame. When the frame is a Command Frame, the driver does not need to issue a READ FRAME command because the PREVIEW FRAME command returns the entire Command Frame.

When the frame is a Data Frame, the host software determines its origin, destination address mode, destination IP-type, and data field length. When the datagram is of no interest to the host, the host issues a FLUSH FRAME command to discard the remainder of the frame. When the host driver wishes to read the data field, it loads the WCR with the word count of the data field length plus one additional word. When the data buffer is not contiguous to the header buffer, the host also loads the BAR with the data buffer address. The READ FRAME command causes the transfer of the data field. When a datagram is received with no data field following the header, it is still necessary to issue a FLUSH FRAME or READ FRAME command to free the Entryway's frame buffer.

The second mechanism is simpler. It is used in installations where the following conditions are met:

- All frames are of interest to the same low-level host process.
- All arriving frames are less than or equal to the maximum size of the input buffer.
- There is no need to separate headers from the rest of the frame.

With these conditions met, program the WCR with the maximum buffer length plus one additional word, and program the BAR for the address of the buffer. Issue the READ FRAME command to start the transfer. The Entryway signals termination when the transfer is complete. The frame length field is used to derive the number of bytes actually transferred.

Error termination occurs in both of the above mechanisms under the following conditions:

- No frame exists at the interface when the PREVIEW FRAME or READ FRAME commands are issued.
- A WCR overflow condition indicates loss of synchronization.

Note that when a WCR overflow occurs, an extra, erroneous word will have been transferred across the interface. In your program design you should ensure that the memory word following each header or input buffer can be modified without damaging the host system.

A WCR overflow results when the second frame input mechanism (described above) transfers a frame whose length is greater than the maximum buffer size of the host. The host must issue a FLUSH FRAME command following the RESET to remove the oversized frame from the queue.

RS-232/8-BIT PARALLEL DATAGRAM SERVICE INTERFACE

This section provides information that will help users with either an RS-232 or an 8-Bit Parallel interface to exchange frames with the Datagram Service. Before writing an application that uses the Datagram Service, you should have some knowledge of the interface that your host computer uses to communicate with the Datagram Service. Refer to the frame format description in this chapter.

Hardware Requirements

The Serial/Parallel I/O Module or the Six Serial Port I/O Module and Processor Board combination must be installed in the Entryway to use these interfaces. Note that these modules use the Multiport Datagram Service (MDS).

Software Requirements

You must use the CONFIGURE program to configure an Entryway for the Datagram Service software. The host computer must implement all frame construction, sequencing, acknowledgement, and error recovery logic.

Using the RS-232/8-Bit Parallel Interface

As mentioned above, the RS-232 and 8-Bit Parallel Interfaces use the Multiport Datagram Service. The Multiport Datagram Service is supported by the stream-oriented driver that is also used with the Virtual Circuit Service.

Although the unit of data (frame) passed by this driver is the same as the unit of data passed by the driver for the single-port interfaces, the stream-oriented drivers transfer data as a stream instead of as distinct units. To enable a host or the Datagram Service to recognize frame boundaries, a frame must be preceded by an ASCII 〈STX〉 character and followed by an ASCII 〈ETX〉 character. These characters assure Entryway-host synchronization.

When the host driver or the Datagram Service detects an ⟨STX⟩ character, it expects the next two bytes in the stream to be the length of a frame. The driver/Datagram Service then reads the number of characters specified in the length field. After reading the number of characters specified in the length field, the driver/Datagram Service expects the next character to be an ⟨ETX⟩. When the ⟨ETX⟩ character is not the character following the number of bytes specified in the length field, the Datagram Service discards the frame because it assumes that either there was a transmission error, or that the ⟨STX⟩ was not a legitimate start of frame indicator. A host driver should also discard any such frames.

Note that the host driver must use this method to isolate a frame from the data stream. It is not possible to use the ⟨STX⟩ and ⟨ETX⟩ characters as simple delimiters. Because it is possible that one of the eight-bit sequences in the frame is an ⟨ETX⟩ character, checking each character after the start of frame indicator for the ⟨ETX⟩ character is not a valid method for detecting the end of the frame.

Because any binary value may appear withing a frame, both the host driver and the Entryway driver must be insensitive to all binary character codes. When you use CONFIGURE to establish the characteristics of an Entryway that runs MDS, specify the following parameters in order to make the Entryway insensitive to binary character codes.

- No Disconnect Sequence
- No data link escape character
- No character flow control

Note that you can use RS-232 line signal flow control if necessary.

A host using the multiport software is responsible for preceding any frame it sends to its Entryway with the ⟨STX⟩ character, and for sending an ⟨ETX⟩ character after the last byte of the frame. The Entryway adds the ⟨STX⟩ and ⟨ETX⟩ characters to any frame it sends to an attached host that uses the multiport software.

RESET HANDSHAKE PROTOCOL

As a result of registering IP-types, the host builds up a certain state in the Entryway. If an Entryway is reset, it loses this state. The Reset Handshake Protocol solves this problem, and re-establishes host synchronization with the Entryway. The algorithm the host implements for this protocol is displayed below.

```
Power UP:
take host restart actions:

xmt_flag := true;

LOOP
    if xmt_flag then
      send "clear" frame
      start rexmt timer
      xmt_flag := false
    fi;

    recv frame f
    CASE f OF
      "cleared": EXIT
      "entryway_reset" : send "entryway_reset_seen" frame
       otherwise        : discard
    ENDCASE

    If rexmt timer expired then
      xmt_flag := true
    fi

REPEAT
Normal:
LOOP

    ....
    recv frame f
    CASE f OF
      ....
      "entryway_reset"  : send "entryway_reset_seen" frame
                          take entryway_reset actions;
      ....
```

The Datagram Service reserves four Function Codes for commands that help a host program synchronize with the Entryway state when the host first powers up, or when either the host or the Entryway is reset. These four commands are Entryway_Cleared, Entryway_Reset, Clear_Entryway, and Entryway_Reset_Seen. Thse commands are part of the host and Entryway algorithms for the Reset Handshake Protocol.

The algorithm that the Entryway uses to implement its part of the protocol is displayed below. The Entryway algorithm is given only to make the host algorithm easier to understand in the discussion below. The host driver only needs to implement the host algorithm; the Datagram Service contains the Entryway algorithm.

```
PowerUP:
take entryway restart actions;

xmt__flag := true;

LOOP
    if xmt__flag then
      send "entryway__reset" frame;
      start rexmttimer
      xmt__flag := false
    fi;

    recv frame f

CASE f OF
      "entryway__reset__seen"     : EXIT
      "clear__entryway"           : send "entryway__cleared" frame
      otherwise                   : discard
    ENDCASE

    If rexmt timer expired then
      xmt__flag := true
    fi

REPEAT

Normal:

LOOP
    ....
    recv frame f
    CASE f OF
      ....
      "clear__entryway"   : take clear__entryway actions;
                          send "entryway__cleared" frame
      ....
```

Notice that each algorithm displayed has two sections: Power-Up and Normal. The PowerUp loops show what should happen when the host or the Entryway is reset; the Normal sections show what happens if the host or Entryway is reset while the other is already running.

When a host goes through a power-up sequence, it must register the IP-types it wishes to use. Before registering IP-types, however, it should issue the Clear__Entryway command to make sure that no IP-types are currently registered for that host.

The host then sets the transmit flag (xmt__flag) to false and waits for either an Entryway__Cleared or an Entryway__Reset command. Set the response timer for about 10 seconds. This value will probably prevent the host from mistaking a slow response for no response.

Once the host receives an Entryway__Cleared command, it can proceed with normal opertion. However, if the Entryway is going through a power-up sequence at the same time as the host, the host receives an Entryway__Reset command and must respond with an Entryway_Reset_Seen command before starting normal operation. If the host does not receive one of these commands before the timer expires, it retransmits the Clear_Entryway command, and waits again for a response.

The Normal section of the host algorithm is necessary when the Entryway must be reset after the initial power-up of the Entryway and the host. When this occurs, the host must send the Entryway_Reset_Seen command, and then reregister its IP-types.

The Entryway algorithm is essentially the same as the host algorithm. When going through a power-up sequence, the Entryway must inform the host that it has been reset and wait for acknowledgement of this message. When the host is already operational, it acknowledges the message and reregisters its IP-types; when the host is also going through a power-up sequence, the Entryway receives a Clear_Entryway command and responds with an Entryway_Cleared command.

Chapter 13
Error Messages

Contents

CONFIGURE PROGRAM

This section describes Configure program messages displayed during consistency checking. Consistency checks occur at different points during the Configuration process. These checks verify that your answers to Configure's menus will work together and will not cause an operational failure. The error message descriptions include the probable cause of the error and a suggested action for correcting it.

I/O Module

This menu is inconsistent. Do you want to proceed anyway?

During file consistency checking, Configure finds choices that are incompatible. Type Y to ignore the incompatibility and continue as though there were no problem. Type N to return to the first menu in the particular group of menus where the problem was found, and then correct your answers.

Device on port (number) is incompatible with port.

An illegal device is connected to a port. For example, a serial device is connected to a parallel port, or a device configured for a Six Serial Port I/O module is attached to a Serial/Parallel I/O module. You will have to reconfigure the port. The device and I/O module will not work when this error condition exists.

(Name) driver is required but is not present.

Devices are connected to a port for which an incorrect driver has been selected. For example, you configured a parallel device on the Serial/Parallel I/O module, yet only serial drivers were selected for the board. Reconfigure the board to supply the correct driver(s) for the I/O module. The device will not work on the port without the correct driver.

(Name) driver is present but is not required.

You chose a driver that is not used by any device on the I/O module. For example, you selected the serial asynchronous and eight-bit parallel drivers for a Serial/Parallel I/O module to which no parallel device is connected. This is not a serious error; the Entryway will operate correctly. You need to correct this error only when Configure tells you that the configuration file is too large to fit into the available memory space.

You may not mix aysnc ports using extended RS-232 signaling with parallel ports. Port (number) uses extended RS-232. Parallel port (number) is in use.

You configured a parallel port on the Serial/Parallel I/O Module and chose extended RS-232 signaling. If you select extended signaling for Port 1 or 2, you cannot use Parallel Port 5. If you select extended signaling for Port 3 or 4, you cannot use Parallel Port 6. You must change your configuration. The device and I/O module will not operate correctly when this error condition exists.

Port (number) uses programmable handshaking. Port (number) is configured. You cannot have both.

When you select programmable handshaking, only one parallel port can be used. You must remove or change one of the parallel port configurations.

You have not loaded a parallel driver for this port.

You have configured a parallel port on a Serial/Parallel I/O module, but did not select the eight-bit parallel and serial asynchronous drivers. You must select the correct driver for the Serial/Parallel I/O module if you want to use the parallel ports.

Device Characteristics

Incorrect threshold values. Input buffer size must be greater than the stop-device threshold, which must be greater than the restart device threshold.

This error can occur if you change the stop-device threshold value and skip the restart-device threshold question. To correct it, change these values.

No parity with 8 bits per character.

Parity check is not allowed with eight bits per character on a serial asynchronous port. You must change the number of data bits to a value other than eight, or select 'no parity'.

Virtual Circuits

Command interpreter is required but not present.

You configured a port for a LAN command interpreter, but did not configure the board for one. Reconfigure the board and select a command interpreter.

Command interpreter is present but not required.

You did not configure a port for a command interpreter, yet you selected a command interpreter for the board. This is not a serious error; the Entryway will operate correctly. If Configure tells you that the configuration is too large to fit into available memory space, you will have to correct it.

This board can only support (number) circuits. You have configured (number). You must lower some of your guarantees.

You have guaranteed more virtual circuits than the board can support. You must reduce the number of guaranteed circuits. Refer to Table 6-1 in Chapter 6 for a list of allowed circuits on the Serial/Parallel and Six Serial I/O modules.

Multiple virtual circuits are not available on this board.

You configured multiple virtual circuits for a port but did not select them for the board. You must reconfigure the board for multiple virtual circuits.

Disconnect sequence contains an illegal character. Hold sequence contains an illegal character.

The first character of a sequence appears elsewhere in the sequence. For example, the disconnect sequence "ABAC" is not allowed because the first character "A" is also the third character; the hold sequence "BCDA" is not allowed because the "A" is the same as the first character of the disconnect sequence. You must change the disconnect or hold sequence.

Break signal is configured to be ignored. Causes no disconnect or hold.

If you have chosen to ignore the break key, you cannot use the break key for a disconnect or hold sequence. If you want to use the break key for disconnect or hold, type N when Configure asks: "Do you still want the break key to be ignored?"

Invalid I/O module type for virtual circuits. Reverting to Six Serial I/O module.

You selected virtual circuits and then named a similar board that does not support virtual circuits. Configure displays the default values for the Six Serial I/O module. If this is not the I/O you are configuring, back up and answer the question again.

Entryway Level Checks

Space required for this configuration is too large by (number) bytes.

Your configuration uses more memory than the application processor has available. You should consider any of the following fixes:

- Reallocate device buffer space.
- Remove unnecessary drivers. For example, remove parallel device drivers on a Serial/Parallel I/O module when only serial devices are used.
- Remove Multiple Virtual Circuits.
- Change an extended command interpreter to a standard command interpreter.
- Remove an unnecessary command interpreter.

LC file is inconsistent. Write it anyway?

Your answers to menu choices are inconsistent. This message displays at the end of your configuration session before the LC file is written to the disk. Type Y to save the LC file. You can run Configure again to modify it. If you type N, the LC file is not saved. As a general rule, ''When in doubt, write it out.''

Cannot write LC file: (filename)

For some reason, Configure cannot write the LC file to the disk or diskette. If you are storing your configuration files on a diskette, check to see that it is write-enabled.

Cannot read LC file.

When you choose a similar board, Configure checks that the LC file for the similar board exists, and checks that the choices you selected for it are consistent with what you have already selected for the current board. This message also displays when you name a similar board for which no LC file can be found. You chose ''Modify'' as a response when ''Add new board'' is the correct answer. Correct this by using the Backspace character to back-up and answer the question again. You can also opt to continue in which case Configure will display system default values.

The ''A'' board must be present.

No ''A'' processer board is configured for the Entryway. You must configure all Entryways with an ''A'' board.

You cannot mix Ethernet Data Link Service boards with non-Ethernet boards. (Board) is not an Ethernet board.

When one board in an Entryway supports the Ethernet Data Link Service, all boards in the Entryway must support it. You must reconfigure the Entryway.

DOWNLOAD SERVER

The Download Server displays the following message when it cannot find an LS file for the Entryway which is requesting software:

DLS - Cannot open LS file

ERROR CODES

The messages listed on the following pages can occur during any terminal operating session. They are responses to network-level operations, and do not represent terminal operator messages except in cases where a connection resource was entered incorrectly or the specified device has not been defined to the network software through the Configure program.

Number	*Meaning*
0	Failure. The operation did not complete successfully. The most common cause of failure is a timeout in the network software. Try again.
3	Bad parameter. A bad parameter value was passed internally to the network. This might be an illegal version number or an out-of-bound value for any parameter having a fixed, system-defined set of allowed values. This is a programming error. If this occurs, contact your local Codex representative.
9	Timeout. The remote Entryway failed to respond to a network message within the timeout limit (approximately 10 seconds). This condition occurs either when the processor is temporarily overloaded or when the Entryway is short of memory space. This condition can occur any time communication between Entryways is attempted. Try again.
11	Network resource not available. The named device or service is currently busy and cannot accept a connection. Codex LAN returns this error code only when the specified name is a unique name. Try again later.
58	Connection descriptors exhausted. The local or remote Entryway has no available memory space in which to perform the operation. No corrective action is possible.

Number	*Meaning*
60	Connection cannot be broken. The device to be disconnected is not active, the device is not connected, or an attempt is made to disconnect the command interpreter from a device.
65	Local buffers exhausted. The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
69	Remote buffers exhausted. The remote Entryway has no available memory space in which to perform the operation. No corrective action is possible.
150	Local device data structures exhausted. The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
151	No such resource. Either the specified resource is not configured, or it is inactive.
152	Local process descriptors exhausted. The local Entryway has no available memory space in which to perform the operation. No corrective action is possible.
199	No memory. The internal Entryway memory resources that communicate with the User Board are not available. No corrective action is possible.
201	No response for specified name. The specified name cannot be found on the network. The Entryway with that resource may have failed. The Entryway also may be temporarily overloaded and unable to respond.
202	All ports busy. All ports or services associated with the specified name are busy. This code is returned only when the specified resource is a rotored name. Try again.

Number	*Meaning*
203	No response from address. No Entryway responds to the specified address.
204	No "n"th name. No "n"th aliased name exists in the name table for the specified address. This error is associated with the programming environment. It is not returned by the Command Interpreter.
205	Stream broken. An I/O operation is requested on a stream that has been broken. This error is associated with the programming environment. It is not returned by the Command Interpreter.

Glossary

This glossary defines terms used in data communications and in the 4000 Series LAN Administrator's Guide.

Acknowledge Character (ACK)	A transmission control character transmitted by a station as an affirmative response to the station with which the connection has been made; a transmission control character transmitted by a receiver as an affirmative response to a sender.
ANSI	American National Standards Institute; the coordinating organization for standards in the United States.
Application Board with Network Processor	The first processor board in an Entryway (known as the A Board), which coordinates access between the other processor boards and the Transceiver Interface. This board is capable of supporting any I/O Module.
Application Processor Board (APB)	The second through fourth optional processor board in an Entryway (known as the B, C, and D boards). These boards are capable of supporting any I/O module.
ASCII	American Standard Code for Information Interchange; a standard seven-bit (eight bits with parity) character code used for data communications and data processing.
Asynchronous Data Transmission	From the Greek A (Not), SYN (With), CHRONOUS (Time). Data without an accompanying clock; timing is built into each transmitted character in the form of start and stop bits.

Automatic Baud Rate Detection	A mechanism employed by a port on the Six Serial Port I/O Module to determine the data rate of a sending device.
Baseband	With reference to a LAN, the system whereby digitally encoded information is directly connected to the transmission medium without being modulated. Compare Broadband.
Baud	A measurement of the signaling speed of a data transmission device; equivalent to the maximum number of signaling elements, or symbols, per second that are generated; may be different from bits per second (bps).
Bisync (BSC)	A family of character-oriented binary synchronous communications protocols.
Bit	The smallest unit in a byte of information, expressed in 1's or 0's.
Bps	Bits per second. The basic unit of measure for serial data transmission capacity.
Bridge	A LAN active device which provides a communication path between logically or physically separate networks. See also Gateway.
Broadband	A LAN frequency-division multiplexing scheme which commonly employs CATV technology to carry analog signals on the transmission medium. Compare Baseband.
Broadcast	Describes the class of media for which Ethernet is designed, in which all stations are capable of receiving a signal transmitted by any other station. Also describes the mode of usage of such a medium by the Data Link Layer in which all stations are instructed to receive a given frame.
BSC	See Bisync.

Buffer	A device between the source and destination that temporarily stores data while the destination is busy. Buffers allow transmission devices to accommodate different data rates, to perform error checking, and to retransmit data received in error.
Byte	A group of bits, usually seven or eight, which represents a unit of information.
Carriage Return Character [CR]	A control character which causes the print or display position to move to the first position on the same line.
Cathode Ray Tube (CRT)	An electronic device that can be used to display graphic images and which is commonly used in data processing.
CATV	Cable television (previously Community Antenna Television) technology, commonly employed by broadband LAN's for signal distribution.
Character	Standard bit representation of a symbol, letter, number, or punctuation mark; generally means the same as byte.
Clock	An oscillator-generated signal that provides a timing reference for a transmission link; used to synchronize events such as digital logic or transmission.
Coaxial Cable	A cable consisting of one conductor (usually a small copper tube or wire) surrounded by a shield, normally aluminum tubing or copper braid.
Command Interpreter	A software module which converses with the device driver and Session Level Interface; allows a user to establish and manage virtual circuits.
Connection Resource Name	The name of a device or service on the LAN. Used as a connection address.

Control Character	A character whose occurrence in a particular context initiates, modifies, or stops a certain operation. Control characters may be recorded for use in a subsequent action. A control character is not a graphic character, but may have graphic representations in some instances.
CRT	See Cathode Ray Tube.
Datagram Service	A software module which allows an intelligent device to send and receive Codex LAN internetwork packets through either a single high-speed parallel interface (single port Datagram Service, providing IEEE-488 and DR-11 interfaces) or multiple lower-speed interfaces (Multiport Datagram Service, providing RS-232 asynchronous interfaces).
Data Link	The physical means of connecting one location to another for the purpose of transmitting and receiving data.
Data Link Escape Character	Signifies to the network software that the control characters it precedes should be passed across a circuit as data.
Data Rate	The average number of bits, characters, or blocks per unit of time passing between corresponding equipment in a data transmission system.
DCE	Data Circuit Terminating Equipment; in a communications network, equipment that is either part of the network, a network node, or equipment at which a network circuit terminates. With the RS-232 ports, a modem is generally a DCE.
Dedicated Line	A dedicated circuit; a nonswitched channel. See Leased Line.

Device Driver	A software module which allows the network to access many different devices without requiring changes to the majority of the network code.
Dial-Up Line	A temporary circuit established via the use of the switched telephone network.
Directory	Disktalk directories are special volumes that act as pointers or indexes to other volumes or subdirectories.
DTE	Data Terminal Equipment; generally end user devices such as terminals and computers, which connect to DCE ports on Entryways or to modems.
Duplex Transmission	Data transmission over a circuit capable of transmitting in both directions at the same time. Synonymous with Full Duplex Transmission.
EBCDIC	Extended Binary Coded Decimal Interchange Code; a set of 256 characters, each represented by 8 bits. Used extensively by IBM Corporation.
Endpoint	The logical or physical entity at the end branch of a network. Endpoints include application processes.
Ethernet Data Link Service (EDLS)	A software module which allows an attached intelligent device to send and receive Ethernet Data Link packets. EDLS assures best-effort delivery of packets but does not guarantee delivery, sequencing, or uncorrupted data. Most intelligent devices using this service provide their own error recovery protocols.
Extended Command Interpreter	A software module which provides the same capabilities as the standard Command Interpreter but also provides the ability to examine, interconnect, and disconnect other virtual circuits.

Field	A position within a message frame, e.g., the frame length field, function code field, and so on.
File Transfer Service	A software module which allows the transfer of user programs between host computers; used in LAN to transfer load modules to the Download Server for subsequent loading into Entryways.
Flow Control	Controls the rate of data transfer between endpoints in a data network; line signals or flow control characters stop and start the flow of data.
Frame	The basic data transmission unit between data-link entities in a LAN; includes a frame header, addressing information, and data.
Frame Check Sequence	An encoded value appended to each frame by the Data Link Layer to allow detection of transmission errors in the physical channel.
Full Duplex Transmission	See Duplex Transmission.
Gateway	A node common to two or more networks through which data flows from network to network. The gateway may reformat the data as necessary and may also participate in error and flow control protocols. Used to connect LANs employing different protocols and to connect LANs to public data networks.
Half Duplex Transmission	Data transmission over a circuit capable of transmitting in either direction, but only in one direction at a time.
Handshaking	A predefined exchange of signals or control characters between two devices or nodes. Sets up the conditions for data transfer and transmission.
HDLC	See High Level Data Link Control.

High Level Data Link Control	(HDLC) The protocol defined by ISO in 1976 for bit-oriented, frame-delimited data communications.
High Speed Serial I/O Module	Provides RS-232, RS-449, and V.35 interfaces under DMA control for speeds up to 700 kbps.
Host Computer	An intelligent processor which provides supporting services to users. Generally self-sufficient, i.e., requiring no supervision from other processors.
IEEE-488 I/O Module	A module which is attached to an application processor board; provides an IEEE-488 interface. This module is also used with an NSM to provide a CP/M environment for network management.
Interface	A shared boundary; a physical point of demarcation between two devices, where the electrical signals, connectors, timing, and handshaking are defined; the procedure, codes, and protocols that enable two entities to interact for the meaningful exchange of information.
LAN Manager	The 4810 PC XT LAN Manager is a specially equipped IBM PC XT which configures, downloads, and stores the network software.
Leased Line	A dedicated circuit, typically supplied by the telephone compnay, that permanently connects two or more user locations; generally voice-grade in capacity and in range of frequencies supported; typically analog, although it sometimes refers to digital channels; also private lines.
Linefeed Character ⟨LF⟩	A control character that causes the print or display to move to the corresponding position on the next line.

Local Area Network (LAN)	A communications network whose dimensions are typically less than 5 kilometers. Transmissions on a LAN are generally digital, carrying data among stations at rates usually above one Megabit per second.
Local Station	In data communications, a device which is connected to a controlling unit by cables, rather than by data links.
Logical Drive	A disk drive on a PC in the Disktalk network that is really a volume on the hard disk of a Network Server.
Mark	A signaling condition equal to a binary 1.
Message	A unit of data transmitted as a logical entity within an information network.
Modem	Modulator/demodulator. Modulates and demodulates signals transmitted over a data communications facility. Provides an interface between a terminal or processor and the communications facility.
Multicast	An addressing mode in which a given frame is targeted to a group of logically related stations.
Network Management Facility (NMF)	The combination of an Entryway and an NSM which together provide the ability to configure, download, and manage the LAN.
Network Server	An IBM PC XT on which the Disktalk/Printtalk software has been installed. Provides access to its hard disk for other workstations.
Network Storage Module (NSM)	A unit that provides disk drive storage for the NOSS software and configuration files.

Node	An endpoint of any branch of a network, or a junction common to two or more branches; a point where one or more functional units interconnect transmission lines. LAN Entryways are nodes.
Packet	A string of characters that includes a source address, destination address, and data. Different data communications systems use different-sized packets.
Parallel Transmission	The simultaneous transmission of all bits in a character or of multiple-bit data blocks.
Parity bit	An additional noninformation bit appended to a group of bits, typically to a 7- or 8-bit byte, which indicates whether the number of ones in the group of bits is an odd or even number; a basic mechanism for error checking.
Pathname	A listing of the directory, subdirectory, and volume name of a file on the Disktalk system.
Physical address	The unique address value associated with a given station on the network. This address is unique from all other physical addresses.
Port	A point of access into a computer, a network, or other electronic device; the physical or electrical interface through which one gains access; the interface between a process and a communications or transmission facility.
Private line	A leased line; a nonswitched circuit.
Protocol	The set of rules governing the format and relative timing of information exchanges.
Queue	A line or list formed by items in a system waiting for service; in Printtalk, a volume which contains the list of files to be printed.
Remote Job Entry (RJE)	The submission of data processing jobs via a data link.

Remote Station	Any device that is attached to a controlling unit by a long distance data link such as a phone line. Compare with local station.
Resource	A network component (hardware or software) which can serve a user requirement.
RJE	See Remote Job Entry.
Rotored Name	A connection resource name shared by several devices on the LAN. Compare with Unique Name.
RS-232-C	An EIA-specified physical interface, with associated electrical signaling, between data circuit-terminating equipment (DCE) and data terminal equipment (DTE); the most commonly employed interface between computers and modems.
RS-449	A newer standard than RS-232-C, also used for serial communications; will eventually replace RS-232-C interface.
SDLC	See Synchronous Data Link Control.
Serial/Parallel Port I/O Module	A module attached to an application processor board which provides TTL parallel, RS-232 asynchronous and RS-232 synchronous communication. This module has four RS-232 ports and two TTL parallel ports.
Serial Port	An I/O port that transmits data one bit at a time. Contrasted to a Parallel Port, which transmits multiple (usually eight) bits simultaneously. RS-232 is a common serial signaling protocol.
Serial Transmission	The sequential transmission of bits constituting an entity of data over a data circuit. Compare Parallel Transmission.
Server name	On the Disktalk system, the name assigned to the hard disk on a Network Server.

Session	1) The period of time during which a user at a terminal can communicate with an interactive system; usually, the elapsed time from when a terminal user logs onto the system until he/she logs off. 2) The period of time during which devices or programs can communicate with each other.
Six Serial Port I/O Module	A module which attaches to an application processor board to provide asynchronous communication through six serial ports.
Space	A signal condition that equals a binary 0. Compare Mark.
Special Character	A graphic character in a character set which is not a letter, not a digit, and not a space character.
Sync Character	A character (two or more in bisync) sent from a transmitting station in order to synchronize the clocks of the transmitter and receiver.
Synchronous Data Link Control (SDLC)	IBM's bit-oriented data communications protocol. Similar to the unbalanced normal class of HDLC. See High-Level Data Link Control.
32-Bit Parallel I/O Module	This module provides a general-purpose 32-bit parallel interface which can be managed in various ways.
Transceiver	The portion of the Physical Layer implementation that connects directly to the coaxial cable and provides both the electronics which send and receive the encoded signals on the cable and the required electrical isolation.
Transmission	The dispatching of a signal, message, or other form of intelligence by wire, radio, telegraphy, telephony, facsimile, or other means (ISO); a series of characters, messages or blocks, including control information and user data; the signaling of data over communications channels.

Unique Name	A connection resource name associated with one device on the LAN. Compare Rotored Name.
Virtual	Conceptual or appearing to exist, rather than actually existing (e.g., virtual storage in computer systems).
Virtual Circuit	A communications path established by computerized switching rather than by a physical link. The virtual circuit exists only when it is carrying data.
Virtual Circuit Service	A software module which allows devices to communicate via virtual circuits in a reliable manner and which allows either intelligent or nonintelligent devices to make use of the network.
Volume	On a Disktalk Server hard disk, a contiguous area of storage space.
Working Directory	On the Disktalk system, the default directory containing the user's volumes and subdirectories.
XON/XOFF	Flow control characters used to start and stop data transmission between a sending and receiving device.

Index

A

B

C

F

H

I

L

M

N

P

Q

R

S

T

U

V

4000 Series LAN
Administrator's Guide
Part No. 08148
May 1985

USER/READER COMMENTS

In order to improve the quality and utility of its manuals, Codex Corporation needs the active cooperation and participation of its user readership. Your comments as a user/reader will be greatly appreciated and reviewed for incorporation in the next revision of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability.

____

____

____

Did you find errors in this manual? If so, specify by page.

____

____

____

How can this manual be improved?

____

____

____

Other comments?

____

____

____

Please state your name

______________________________ Date ________

Position ________________________ Department ____

Organization ____________________ Street ________

City ________________ State ____ Zip or County ____

— — — — — — — — DO NOT TEAR — FOLD HERE AND STAPLE — — — — — — —

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 39783 MANSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

codex

MOTOROLA

Technical Publications 3-50
20 Cabot Boulevard
Mansfield, MA 02048