DataGeneral
Data General Corporation, Westboro, Massachusetts 01580

Customer Documentation

Managing Modems and UUCP on the DG/UX™ System

069–000698–00

AViiON®
PRODUCT LINE

Managing Modems and UUCP on the DG/UX™ System

069–000698–00

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085–series) and / or Update Notice (078–series) supplied with the software.

Printed in the United States of America
Rev. 00, January 1994
Ordering No. 069–000698

Notice

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE–TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS DOCUMENT OR THE INFORMATION CONTAINED IN IT, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES.

Managing Modems and UUCP on the DG/UX™ System
069–000698–00

Revision History:	Effective with:
Original Release – January 1994	DG/UX System 5.4, Release 3.00

About this manual

This manual describes how to install and set up a Hayes Smartmodem™–compatible modem in the DG/UX™ environment. This manual also describes how to set up the UUCP (UNIX to UNIX CoPy) software, which comes with your DG/UX system, to support a variety of asynchronous modem applications.

UUCP is a collection of programs that support telecommunications between remote DG/UX systems, between DG/UX systems and other UNIX®-based systems running UUCP, and between a DG/UX host and its remote terminal users. UUCP can also connect modem users to systems that are not UNIX-based.

This manual assumes the modem is connected either to an asynchronous or synchronous port on an AViiON® computer, or to an asynchronous port on a terminal.

How this manual is organized

This manual contains seven chapters, five appendixes, and a glossary. An overview of the chapters and appendixes follows:

Chapter 1 Describes typical modem-based hardware configurations and typical modem applications. It also describes what you need to know to install your modem and where to find the information.

Chapter 2 Describes the tools you will need to install a modem to support asynchronous or synchronous communications in an AViiON environment. It also supplies general information about modem-hardware configurations and powering up your modem.

Chapter 3 Explains how to set up the DG/UX system to support an asynchronous port on an AViiON computer that is connected to a Public Data Network through a modem. For those using UUCP software, it also explains how to set up the primary UUCP data files to support asynchronous modem communications with remote systems. This chapter also describes how to establish a direct link with a modem, view the contents of the modem's status, and test the link by connecting to a remote system.

Chapter 4 Describes the UUCP commands, servers, directories, and data files and provides examples of UUCP software configurations to support typical asynchronous modem applications.

Chapter 5 Describes how to run UUCP over TCP/IP.

Chapter 6 Explains how to troubleshoot modems and UUCP.

Chapter 7 Explains how to access on-line information.

Appendix A Describes how DG/UX responds to RS-232C modem signals on its asynchronous and synchronous ports.

Appendix B Describes interface connector pin assignments for AViiON asynchronous or synchronous port; for example, Clear to Send (CTS), Data Terminal Ready (DTR), Data Carrier Detect (DCD), and so on.

Appendix C Contains a table of Data General AViiON modem cables.

Appendix D Describes how to set up a modem to use SLIP.

Appendix E Provides a quick reference of modem and UUCP information.

Related Data General manuals

Within this manual, we refer to the following manuals:

- *Customizing the DG / UX™ System* (093–701101). This manual explains how to set up a newly installed DG/UX system. Because this manual is task-oriented, it supports system administrators at all levels of experience.
- *HPS VMEbus Multiplexer (HPS-6236 / 6237) Technical Manual* (014–001817). This manual is a technical reference for the HPS VMEbus multiplexer.
- *Installing and Operating Your D217, D413, and D463 Display Terminals* (014–002057). This manual tells you how to install, configure, and operate the terminals.
- *Managing and Using PAD on the DG / UX System* (093–701089). This manual describes the Packet Assembler/Disassembler (PAD), a product used in synchronous communications.
- *Managing ONC™ / NFS® and Its Facilities on the DG / UX™ System* (093–701049). This manual describes how to install, manage, and use Open Network Computing software and the Network File System on the DG/UX system.
- *Managing TCP / IP on the DG / UX™ System* (093–701051). This manual describes how to install and manage TCP/IP on an AViiON system or network. The manual includes a chapter on the Serial Line Interface Protocol (SLIP). It also includes some basic terms and introductory material about network administration.
- *Managing the DG / UX™ System* (093–701088). This manual introduces DG/UX system management and describes the various DG/UX system-related tasks that the typical system administrator must address.

- *Programmer's Guide: STREAMS* (093–701106). This manual describes the STREAMS interface facility and how to use it.
- *Setting Up and Installing VMEbus Options in AViiON® Systems* (014–001867). This manual together with the AViiON *Setting Up* manuals describe the communications controllers available on AViiON system, including the pinouts of their port connectors and the part numbers of their supporting cables.
- *Setting Up and Starting AViiON™ 400 Series Stations* (014–001858) and *Setting Up and Starting AViiON™ 3000 and 4000 Series Computer* (014–001870). These and other *Setting Up* manuals identify the asynchronous and synchronous ports that support modems.

Other related documents

This section lists books that are related to this manual but which are not available from Data General Corporation. To order books at a discount from O'Reilly & Associates, Inc., use Data General's *UNIX Books, O'Reilly 20% Off Form* (069-100486). You can get copies of the RFC documents from InterNIC Information Services, P.O. Box 85608, San Diego, CA, 94186–9784, 1-800-444-4345.

- *Getting Started with WorldView 1.1 For X Windows Motif* (P/N 75031–0003/a) by Interleaf, Inc., Prospect Place, 9 Hillside Avenue, Waltham, Massachusetts 02154.
- *Managing UUCP and Usenet* (A Nutshell Handbook), O'Reilly & Associates, Inc., 103–A Morris Street, Sebastopol, California 95472, USA. This book provides an overview of UUCP software, how it works, and how to administer and maintain it. Although this book covers several releases of UUCP and many descriptions do not apply to the UUCP software on DG/UX systems, it serves well as introduction to UUCP and as a reference.
- *RFC 1055 (A Nonstandard for Transmission of IP Datagrams over Serial Lines)*. This document describes the basis for Data General's Serial Line Interface Protocol (SLIP).
- *RFC 1144 (Compressing TCP/IP Headers for Low-Speed Serial Links)*. This document describes the header compression technique used by Data General's SLIP.
- *UNIX® Networking*, Hayden Books. This book contains practical discussions of several important UNIX networking systems including UUCP, TCP/IP, NFS®, RFS, Streams, and LAN Manager/X.
- *Using UUCP and Usenet* (A Nutshell Handbook), O'Reilly & Associates, Inc. This manual provides an overview of UUCP software and how to use it from the user's perspective.

Reader, please note:

Throughout this manual we use the following format conventions:

COMMAND **required** *required* [*optional*] ...

Where	Means
required	Enter the case-sensitive characters as shown. On some modems, AT commands are case-insensitive.
required	Enter some argument (such as a filename).
{ *required1* *required2* }	Enter one of the arguments. Do not type the braces; they only set off the choices.
[*optional*]	You have the option of entering this argument. Do not type the brackets; they only identify the argument as optional.
...	You may repeat the preceding entry.

Additionally, we use certain symbols in command lines.

Symbol	Means
<Ctrl-M>	Hold the Control (CTRL) key down and press the M key on your terminal keyboard.
↵	Press the New Line, Carriage Return (CR), or Enter key on your terminal keyboard. For AT commands issued to modems, you must type Ctrl-M or press a key that emits Ctrl-M.
#, $, %	The superuser, Bourne or Korn, and C shell prompts.

Finally, in examples we use:

This typeface to show your entry.
This typeface to show system queries and responses.
This typeface to show file contents.

Contacting Data General

Data General wants to assist you in any way it can to help you use its products. Please feel free to contact the company as outlined below.

Manuals

If you require additional manuals, please use the enclosed TIPS order form (United States only) or contact your local Data General sales representative.

Telephone assistance

If you are unable to solve a problem using any manual you received with your system, free telephone assistance is available with your hardware warranty and with most Data General software service options. If you are within the United States or Canada, contact the Data General Customer Support Center (CSC) by calling 1–800–DG–HELPS. Lines are open from 8:00 a.m. to 5:00 p.m., your time, Monday through Friday. The center will put you in touch with a member of Data General's telephone assistance staff who can answer your questions.

For telephone assistance outside the United States or Canada, ask your Data General sales representative for the appropriate telephone number.

Joining our users group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive *FOCUS* monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1–800–253–3902 or 1–508–443–3330.

End of Preface

Contents

Chapter 1 – Before you start

Chapter 2 – Installing modem hardware

Chapter 3 – Setting up software for asynchronous communications

Chapter 4 – Setting up UUCP to support modem applications

Chapter 5 – Setting up UUCP to run over TCP/IP

Chapter 6 – Troubleshooting

Chapter 7 – Accessing on-line information

Appendix A – How the DG/UX system handles modem signals and connections

Appendix B – Interface connector pin assignments

Appendix C – Modem cable matrix

Appendix D – Setting up a modem to use SLIP

Appendix E – Quick reference: setting up UUCP and modem for basic dial-in service

Glossary

Index

Tables

Table

Figures

Figure

1 Before you start

To install your modem in the DG/UX™ environment in a timely fashion, you will find it helpful to gather the information required to complete the installation tasks before you start. For this reason, this chapter discusses the topics listed below in the following order:

- Identifying the modem application and communications needs.
- Typical modem-based hardware configurations.
- Highlights of what you need to know to complete your modem hardware and software installation, and where to find the information.

Identifying the modem application and communications needs

The role your modem serves is defined by your communications requirements and the type of communications software required to implement the application. In general, modems are configured for one or more of the applications described below.

From a communications software perspective, this manual supplies detailed information only about those applications supported by the UUCP asynchronous communications software that comes with your DG/UX operating system. For detailed information about other communications software packages, see the documentation for the package.

Asynchronous terminal connections to remote host

From a software viewpoint, a remote terminal connection can be as simple as supplying each modem-supported terminal user with a user account and the telephone number of a modem connected to an asynchronous port on the remote DG/UX host. See Figure 1–1.

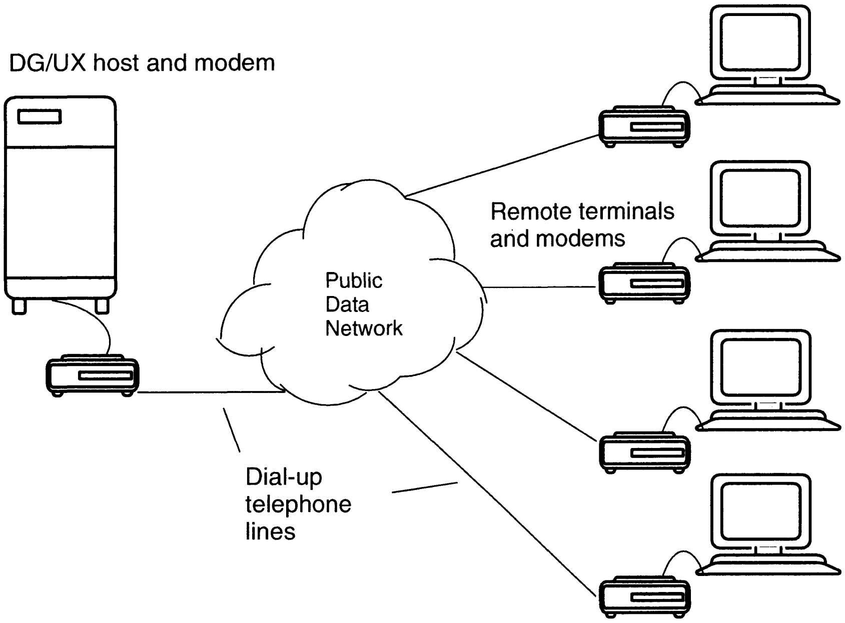


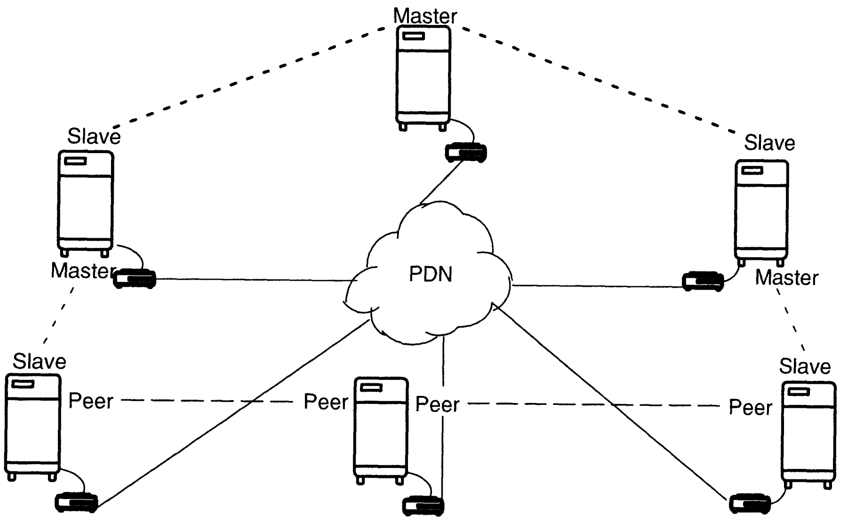
Figure 1–1 Terminal connections to a remote DG/UX host

Asynchronous communications between systems

Using UUCP or other asynchronous communications software packages, you can transfer files between communicating systems. You may also be able to log in remotely or send electronic mail.

With UUCP, you can set up communicating remote systems to operate in either peer-to-peer mode or master/slave mode. In simple terms, those DG/UX systems configured as masters can control the movement of data to and from those systems configured as *slaves*, as well as remotely execute commands and programs, usually at specified intervals of time (*polling*), on slave systems.

As shown in Figure 1–2, you can set up a system using UUCP software to operate in peer mode with some systems and in master or slave mode with others.



——— Dial-up telephone lines connecting hosts, via modem, to the Public Data Network.

- - - - - Operational mode: master/slave

— — — Operational mode: peer-to-peer

Figure 1–2 Using UUCP to operate in peer-to-peer and master/slave modes

Synchronous communications

You can set up synchronous communications between DG/UX systems, or between DG/UX systems and other vendors' platforms running SNA, X.25, or other industry-standard synchronous communications software. The physical installation of a modem in a synchronous communications environment is similar to installation in an asynchronous communications environment.

Typical modem-based hardware configurations

Each modem supports one link in a communications circuit. See Figure 1–3, which illustrates remote terminal modem-based hardware configurations connected to an AViiON® server via dial-up telephone lines. A common use of modems is to connect an AViiON server to remote terminals and personal computers (PCs).

Communicating modems must employ a common data transfer rate (bits per second, or baud rate) during the connection period. Most modems support a range of baud rate settings in hardware, with the highest baud rate being noted in the modem's description. The baud rate of a modem can be set using the Hayes Smartmodem™ command set (the Hayes AT command set); or, when present, baud rate switches on the modem. In addition, some modems can adjust their baud rate to that of the remote modem at the other end of the circuit.

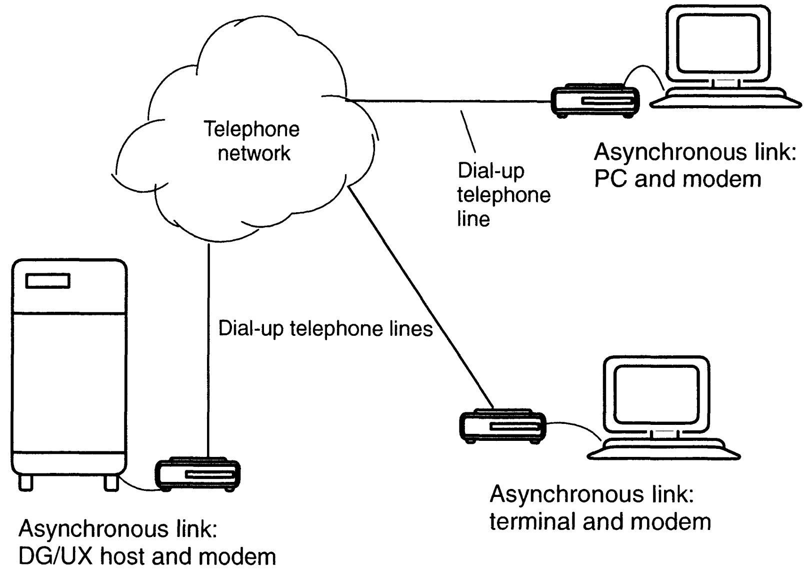

Telephone network
Dial-up telephone line
Asynchronous link: PC and modem
Dial-up telephone lines
Asynchronous link: terminal and modem
Asynchronous link: DG/UX host and modem

Figure 1–3 Typical hardware configurations based on asynchronous modems

Asynchronous data transfers between remote systems connected to a telephone network require only a modem at each end of the communications circuit to complete a connection between the asynchronous ports of two remote systems.

Synchronous data transfers between remote systems connected to a packet switching network (such as one supporting the X.25 protocol) may require the addition of a PAD (Packet Assembler/Disassembler) unit or other software between asynchronous terminals and their modem. See Figure 1–4. For a diagram of the layers of DG/UX software that run on synchronous modems, see the **modem**(5) on-line man page.

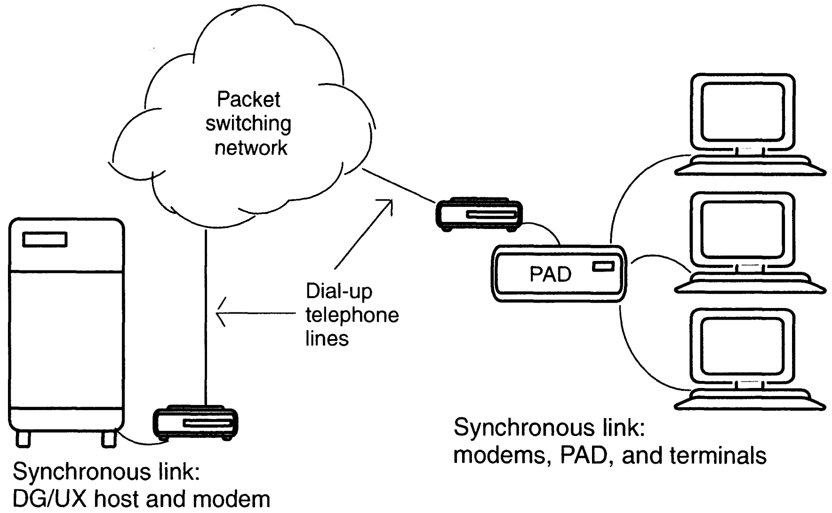

Packet switching network
PAD
Dial-up telephone lines
Synchronous link: modems, PAD, and terminals
Synchronous link: DG/UX host and modem

Figure 1–4 Typical hardware configurations based on synchronous modems

Asynchronous, interactive communications between a remote terminal and a computer system connected to a telephone network require only a modem to connect the remote terminal to the network. However, adding a telephone to the modem configuration lets you use a common telephone line for either voice or data service.

What you need to know and where to find it

This manual is intended to supplement, not replace, the information supplied in your modem and communications software documentation. This section highlights particular types of information you will need to complete your modem installation and the documentation that will supply it. We recommend that you get acquainted with the contents of this manual and the sections of the supporting documentation described below before you start the installation.

Identifying ports that support modems

Your AViiON computer's *Setting Up* or *Installing* manual and the *VMEBus Options* manual identify the asynchronous and synchronous ports on your AViiON® computer that support modems. They also list the model and part numbers of cables that connect these ports to your modem by the standard used (for example, RS-232C).

Determining a port's terminal number

If you are installing a modem on an asynchronous port of an AViiON computer running the DG/UX system, you must know the port's terminal number. The terminal number is indicated by an entry in /**dev** that begins with **tty**. The value of this number depends on how your system is set up.

If you do not know the terminal number, (for example, **tty01**) of the asynchronous port you have selected to support your modem, either ask your system or network manager for the port number or see the "Terminal Line Controllers" section of the Planning Worksheets appendix in *Customizing the DG / UX™ System.*

Determining modem-specific information

To perform the basic installation tasks associated with your modem hardware, you need to know the following information:

- To determine the location and setup of external configuration switches, when present, that select your modem's communications mode (such as asynchronous or synchronous), see "Setting the modem's configuration switches" in Chapter 2 of this manual.

- To determine the set of Hayes AT commands supported by the modem, see your modem owner's manual.
- To determine the meanings of any indicator lights, see your modem owner's manual.
- To determine the location of the modem's external connectors, see your modem owner's manual.
- To determine the special features, if any, offered by your modem and instructions for setting them up using the AT command set, see your modem owner's manual. Also see the "Communicating with a modem using a direct link" section in Chapter 3 of this manual.

Managing asynchronous communications

If your modem supports asynchronous communications, see the "Terminal and Port Management" section of *Managing the DG / UX™ System*. Also see the DG/UX release notice for any information relating to the installation and/or operation of modems in the DG/UX environment.

Managing asynchronous software other than UUCP

If your modem is supported by asynchronous software other than UUCP, see the documentation and release notice for information affecting modem settings (for example, baud rates and clocking sources).

Managing synchronous communications

If your modem supports synchronous communications, see the "Controller Management" section of the *Managing* manual. If your modem is supported by a synchronous communications software product, for example, X.25 for AViiON systems, see the documentation and release notice for those products for information affecting modem settings (for example, baud rates and clocking sources).

What to do next

After you become acquainted with the contents of this manual, and have collected the information described above, you are ready to install your modem. Follow the instructions in Chapter 2 of this manual.

End of Chapter

2 Installing modem hardware

This chapter describes how to configure and connect a Hayes Smartmodem™ or compatible modem either to a DG/UX™ system on an AViiON® computer or to a stand-alone terminal communicating with a remote DG/UX system via modem.

This chapter covers the following topics:

- Tools and other prerequisites
- Selecting a serial port and cable
- Configuring your modem using push buttons and switches
- Connecting your modem to a power source, a telephone line, and either an asynchronous port on a terminal or an asynchronous or synchronous port on an AViiON computer
- Turning your modem on

Tools and other prerequisites

To install your modem hardware, you need the following items:

- A secure surface on which to place the modem
- A small flat-head screwdriver
- The owner's manual that shipped with your modem
- An RS-232C serial communications port with modem support on the local terminal or the serial communications port on the AViiON computer to which the modem will be connected
- The appropriate RS-232C cable to connect the modem to a terminal or serial communications port on your AViiON computer
- A connection to a telephone line

After unpacking your modem and placing it on a secure surface, see your modem owner's manual and get acquainted with the installation instructions. This includes the locations and types of external connectors, the functions of external switches (if present) and indicator lights. Also determine if your modem can be configured to respond to the Hayes Smartmodem command set (Hayes AT command set).

Selecting a serial port and cable

This section describes how to select a serial port and cable for your modem. Topics are as follows:

- Selecting an asynchronous port on an AViiON computer
- Selecting a synchronous port on an AViiON computer
- Selecting an asynchronous port on a terminal
- Selecting the correct modem cable

Selecting an asynchronous port on an AViiON computer

All AViiON computers have at least one RS-232C asynchronous port with modem control signals such as Request to Send, Clear to Send, Data Set Ready, Signal Ground, Data Carrier Detect, Data Terminal Ready, Transmit Data, and Receive Data.

In addition, AViiON computers with VME slots accommodate a variety of asynchronous terminal controllers, each supporting multiple RS-232C ports with modem control.

Selecting a synchronous port on an AViiON computer

Some AViiON computers have RS-232C synchronous ports. Each synchronous port supports a modem.

Selecting an asynchronous port on a terminal

Data General D217, D413, and D463 terminals have two RS-232C asynchronous ports: host port and auxiliary port. Although the auxiliary port is intended for connecting to a second host computer or add-on device, such as a modem or printer, either port can accommodate a modem. See Figure 2–1.

If you are using another terminal, ensure that its RS-232C port supports modem control. See the user's manual for your terminal.

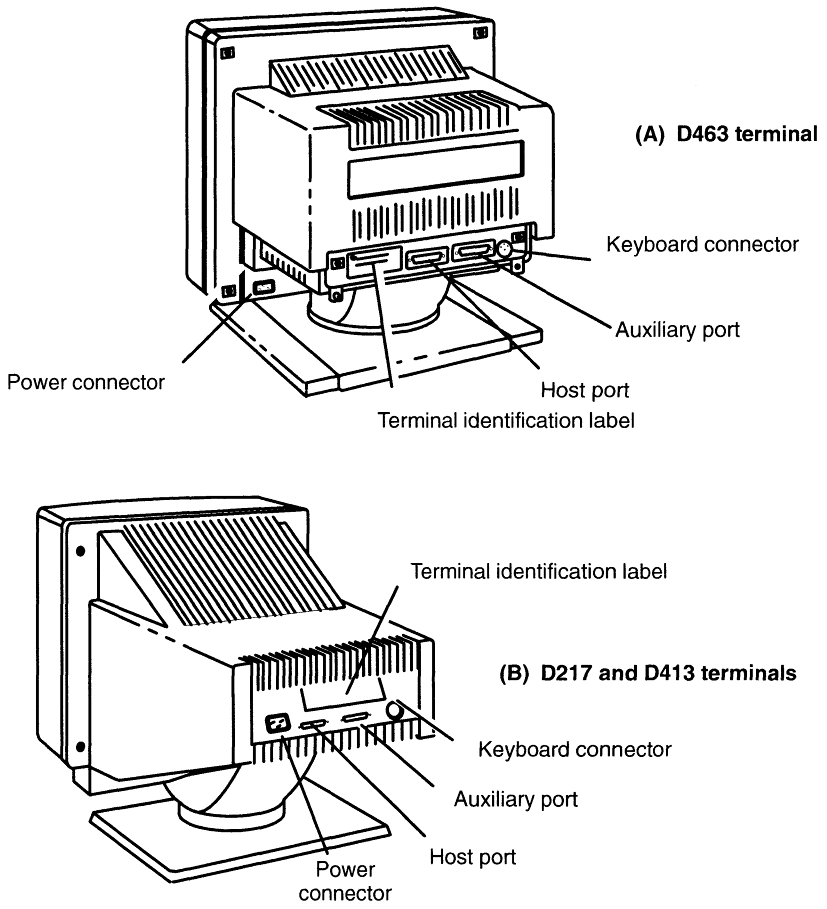


Figure 2–1 D217, D413, and D463 terminal ports and connectors

Selecting the correct modem cable

If you are connecting a modem to an AViiON computer or graphics workstation, see your AViiON installation and VMEbus options manuals. These manuals identify the model and part numbers of modem cables compatible with the serial ports.

If you are connecting a modem to a Data General D217, D413, or D463 terminal, please see *Installing and Operating Your D217, D413, and D463 Display Terminals.* This manual lists the Data General part number(s) for RS-232C modem cables suitable for use with these terminals.

If you are using a terminal from a different vendor, see the owner's or user's guide for your terminal.

Configuring your modem using push buttons and DIP switches

This section describes how to set external push buttons and switches present on some modems. The description uses a Microcom QX/4232hs error-correcting modem (Data General model 18901, part number 119-2087) as an example. If you are installing a different modem, you may find this section helpful but you should also see your owner's manual for information on your modem's configuration switches.

This section covers the following topics:

- Setting the push buttons
- Setting the DIP switches

If your modem doesn't have external configuration switches, skip this section and go directly to the "Connecting a modem to its resources" section, appearing later in this chapter.

In Chapter 3, you will learn how to complete the configuration of your modem, using the Hayes AT command set.

Setting the push buttons

The section describes the modes controlled by push buttons, each of which can be pushed in or out. Table 2–1 shows the meaning of the in and out positions.

Table 2–1 Modes set by modem push buttons

Button	Out	In
T/D	data	talk
O/A	originate	answer
A/S	asynchronous	synchronous

Selecting talk or data mode

To transmit data, choose data mode. For voice communication, choose talk mode. Talk mode is for environments where, for example, you must speak to a person at the remote site you are calling to gain access to the remote computer. You can override the talk setting by issuing a dial command such as **ATDP** or **ATDT**.

Selecting originate or answer mode

To call out, choose originate mode. To receive calls, choose answer mode. This choice is meaningful only if the modem to receive the call is not in autoanswer mode. In such case, the O/A selection is used in conjuction with talk mode. If you are establishing voice communication before transferring data, one modem must be in originate mode and the other in answer mode.

Selecting asynchronous or synchronous communications

Whether you choose asynchronous or synchronous data transfer will depend on the hardware and software that you are using. Since asynchronous communications is more commonly used, the examples later in this manual assume asynchronous communications. For more information about asynchronous versus synchronous data transfer, see the next section.

Setting the DIP switches

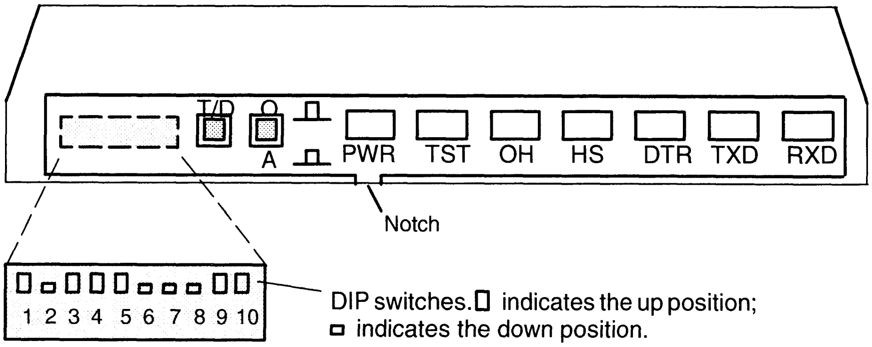
Dual In-line Parallel (DIP) switch settings select characteristics such as AT mode, asynchronous or synchronous mode, and no-return-to-factory-default configuration on modem reset. This section describes how to set the front and rear panel DIP switches for asynchronous communications on the DG/UX system.

If you are installing a modem for synchronous communications, see your modem owner's manual and your synchronous communications documentation for additional information about setting the modem's front and rear panel switches.

Setting the front panel DIP switches

The front panel of the sample Microcom modem shown in Figure 2–2 has two push buttons, T/D and O/A, and ten DIP switches. Some Microcom modems may have an A/S button instead of an O/A button.

If you have a Microcom modem on a DG/UX system, set the switches as shown unless the modem has an O/A switch and you want synchronous communications; in that case, set Switch 10 down.



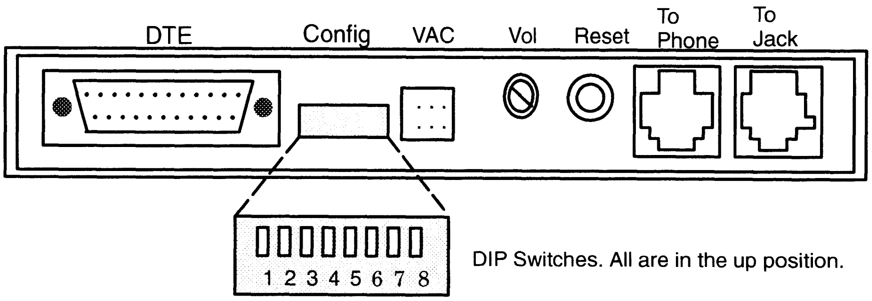
DIP switches. □ indicates the up position; ▫ indicates the down position.

Note: The DIP switches are located behind the faceplate. To remove the faceplate, disconnect the power source, then insert the point of a small screwdriver or paper clip in the notch, and pull out the faceplate.

Figure 2–2 Microcom front panel switches

Setting the rear panel DIP switches

The rear panel of the sample modem shown in Figure 2–3 has eight DIP switches used to set the configuration. It also contains a volume control, a reset push button, an asynchronous (DTE) connector, two RJ11 telephone line connectors (one to the telephone and one to the wall jack), and a VAC (volts alternating current) connector.



DIP Switches. All are in the up position.

Figure 2–3 Microcom rear panel switches

If you are running the Serial Line Internet Protocol (SLIP), set the rear panel switches as shown in Appendix D.

Connecting a modem to its resources

Figure 2–4 shows how to connect your modem to a power source, a local communications port, a telephone line, and to a telephone. After connecting your modem, you are ready to turn it on.

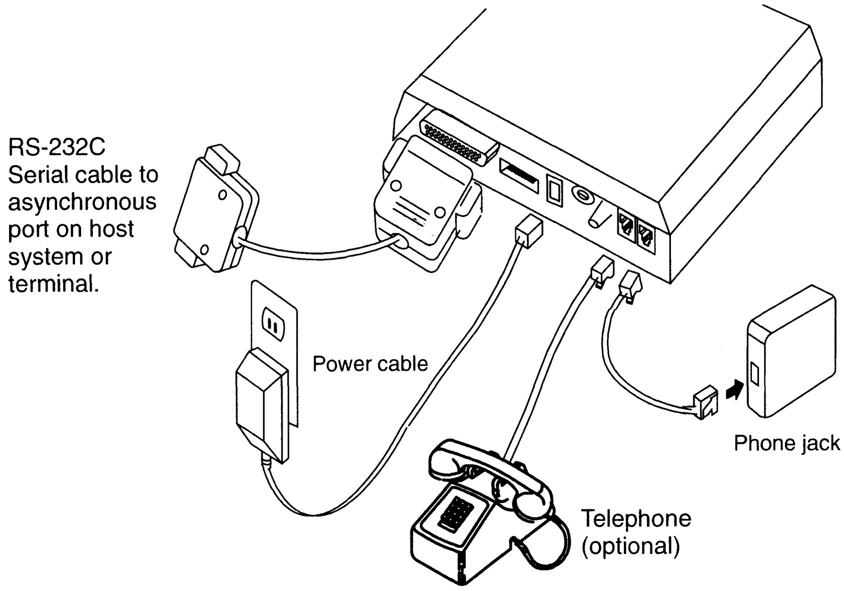

RS-232C
Serial cable to asynchronous port on host system or terminal.
Power cable
Phone jack
Telephone (optional)

Figure 2–4 Typical modem connection diagram

Turning your modem on

See your modem owner's manual for information about your modem's indicator lights. When you turn on the modem, make sure that it is operating according to owner's manual description before proceeding further.

If your modem has a power pack, see your modem owner's manual for instructions on charging the power pack. Frequently, the power pack must charge for 24 hours before use.

Where to go next

Your modem hardware is now installed, and you are ready to set up the DG/UX system and the appropriate communications software to support your modem.

If you are installing a modem that will use asynchronous communications software, such as UUCP, continue on to the next chapter.

If you are installing a modem for synchronous communications, first see the section that describes communicating with a modem using a direct link in Chapter 3; then see the documentation for your synchronous communications software.

End of Chapter

3 Setting up software for asynchronous communications

Although you may plan to use a communications software package other than UUCP, you still must set up the DG/UX system to support the modem. In addition, you may find useful the information about communicating with a local modem using a direct link.

This chapter covers the following topics:

- A general description of UUCP
- Choosing the type of connection
- Specifying the type of connection
- How to set up the DG/UX system to support a modem
- How to initially set up the UUCP software
- How to communicate with your modem and test its initial setup using a direct link

What is UUCP?

UUCP is a set of programs and data files that allow you to transfer files and to execute remote commands between UNIX systems. In this section, the name UUCP refers to the entire system of programs and databases; the name **uucp** refers to the **uucp**(1) command itself.

UUCP is one of three DG/UX base system software packages that can provide network service on your system. Each package has its own menu and operations in **sysadm**'s Networking menu. The other two are TCP/IP (see *Managing TCP/IP on the DG/UX™ System*) and ONC/NFS (see *Managing ONC™/NFS® and Its Facilities on the DG/UX™ System*).

The UUCP software loads as part of the DG/UX package. Like TCP/IP, it functions using several server programs and a number of database files.

UUCP typically uses telephone (modem) connections and direct (port-to-port) connections as its network rather than Ethernet and Internet networks. However, UUCP can be used with TCP/IP.

Choosing the type of connection

Before your system can communicate with a remote system, you must set up a two-way communication connection between the systems. This section describes the two kinds of UUCP connections: the direct connection and the dial-up connection.

When to use a direct connection

The direct connection communication method requires a direct connection from a port on a local host to a port on the remote host. A direct line is advantageous when communication is required with the remote host on a regular basis. The link is always available and access time is short. The disadvantage of the direct link is that the port cannot be used for anything else.

Direct connections are beneficial only when:

- It is not possible to link the hosts together through a local area network (LAN).
- Two hosts transfer large amounts of data on a regular basis.
- Two hosts are connected by no more than several hundred feet of cable.

The distance between two directly linked hosts is dependent on the environment in which the cable is run. The standard for RS-232 connections is 50 feet (15.25 meters) or less with transmission rates as high as 19200 bits per second. As the cable length is increased, noise on the lines may become a problem, which means that the transmission rate must be decreased or limited-distance modems be placed on each end of the line.

Do not use more than 1000 feet (305 meters) of cable to connect the two hosts or communications will be unreliable. This link should operate comfortably at 9600 bits per second in a noise-free environment.

When to use a dial-up connection

In this case, the host that is going to make the connection would call the remote host using an Automatic Calling Unit (ACU). The remote host answers via its own ACU and makes the connection. With this arrangement, the ports are not dedicated to only one host. A dial-up link also requires more hardware (such as the ACU) than the direct connection. Transmission rates are limited to the capacity of the ACUs.

Specifying the type of connection

To specify the type of connection, create an entry in **/etc/uucp/Devices** and an entry in **/etc/uucp/Systems**.

When you have determined which communication links best suit your needs, you will need to set up port services for those lines. To set up serial lines and port services, see the Port menu in **sysadm**'s Device menu. *Managing the DG / UX™ System* discusses the Port menu.

Specifying a direct connection

To specify a direct connection, put `Direct` in the first field of the **Devices** entry and in the third field of the **Systems** entry.

Specifying a dial-up connection

To specify a dial-up connection, put `ACU` in the first field of the **Devices** entry and in the third field of the **Systems** entry.

For each caller type that appears in the fifth field of a **Devices** entry, you must create an entry in **/etc/uucp/Dialers**. In **/etc/uucp/Dialers.proto**, you'll find a description for setting up various kinds of modems.

If your modem name is not listed in the **Dialers** file, you will need to edit this file and create a chat script with your modem name as a label. The chat script is the sequence of commands a modem uses for dialing out. For information on your modem's language and command syntax, refer to your modem's documentation .

Setting up the DG/UX system for a modem

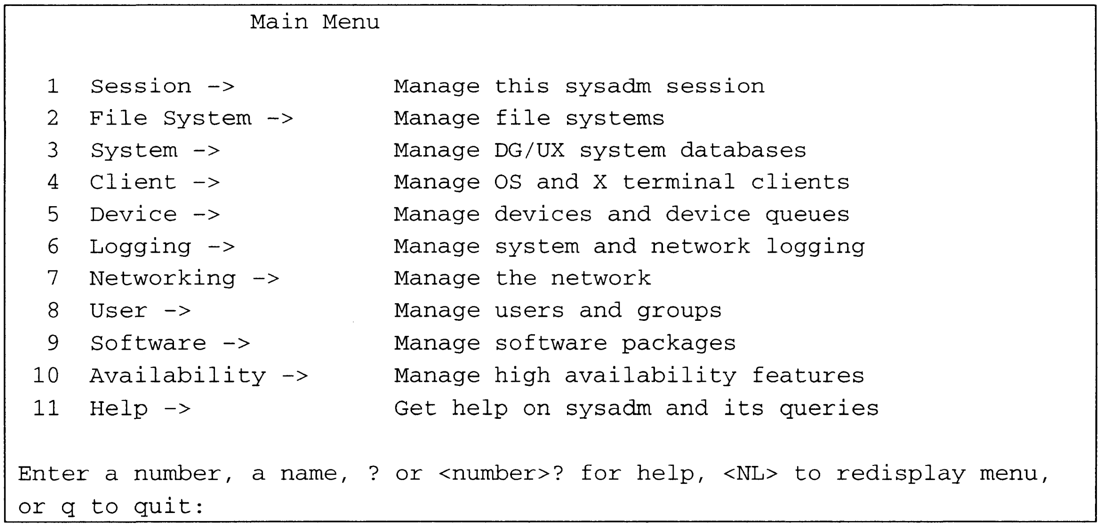
This section explains how to set up a modem to operate in the DG/UX environment. It assumes that you have completed hardware installation of the modem. This section describes the following tasks:

- Starting the DG/UX **sysadm** utility.
- Determining a port's asynchronous (**tty**) line number.
- Using the DG/UX **sysadm** utility to add your modem to the designated asynchronous port on your AViiON computer.
- Using the DG/UX **sysadm** utility to modify the port service for the designated asynchronous port to support your modem.
- Making UUCP the owner of the modem's asynchronous port.

The examples that follow are based on the ASCII version of **sysadm**. If you are using a graphics workstation, invoke the ASCII version by typing **asysadm** in an **xterm** or **mterm** window. When the **sysadm** Main Menu appears, the number and order of some choices may vary from the example, based on the software packages you have installed.

Starting sysadm

1. While operating at run level 3 (**who –r** displays the current run level), log in to the DG/UX system as **sysadm**. (For an explanation of run levels, see the **init**(1M) man page.)
2. Invoke the **sysadm** Main Menu by typing **sysadm** at the shell prompt. **Sysadm** displays the menu shown in Figure 3–1.



```
                 Main Menu

  1  Session ->            Manage this sysadm session
  2  File System ->        Manage file systems
  3  System ->             Manage DG/UX system databases
  4  Client ->             Manage OS and X terminal clients
  5  Device ->             Manage devices and device queues
  6  Logging ->            Manage system and network logging
  7  Networking ->         Manage the network
  8  User ->               Manage users and groups
  9  Software ->           Manage software packages
 10  Availability ->       Manage high availability features
 11  Help ->               Get help on sysadm and its queries

Enter a number, a name, ? or <number>? for help, <NL> to redisplay menu,
or q to quit:
```

Figure 3–1 Sysadm main menu (ASCII version)

3. As **sysadm** prompts you for responses in the interactive dialogue described below, read the comments immediately following each "*prompt*:**response**" pair shown in this manual (for example, `Tty Device(s):` **tty*nn***) before entering your response. If you need more information, enter a question mark (?) at any **sysadm** prompt. If no comments follow the *prompt*:**response** pair, enter verbatim the response indicated in **bold**.

Determining a port's tty line number

The DG/UX system automatically assigns a **tty** line number to each attached port in the hardware configuration when the system boots. A **tty** line number takes the form:

tty*nn*

where:
nn is a sequentially assigned number. For example, **tty00** refers to the first port, **tty01** the second port, and so on. A file with the name of the **tty** line number is created in the **/dev** directory each time the system boots.

If you have terminals attached to multiple, different terminal line controllers, you must determine each terminal's **tty** line number. For information on determining **tty** line numbers, refer to *Customizing the DG/UX™ System.*

Adding a modem to the port

Modems also can be connected to terminal line controllers. In this section, the term *port* is used to refer to terminals and modems. The section applies only to modems that are connected to intelligent line controllers not to **duart** ports (see the **duart**(7) manual page).

To add one or more modems, you must use the Port Services Menu rather than the Terminal Menu. The **tty** definition label for a modem typically begins with **M** and is followed by the modem speed. For example, a **tty** definition label for a 1200-baud modem line is **M1200**. The **tty** labels are located in the first field in each line of the **/etc/ttydefs** file. A **ttymon** port monitor can monitor both modem and terminal lines.

1. Add a modem to the port as follows. From the **sysadm** Main Menu, execute the following path through the submenus:

   Device → Port → Port Monitor → Add

2. Accept the default port monitor type, and enter a name for the port monitor. For example:

```
Port monitor type: [ttymon] ↵
Port monitor tag: ttymon3 ↵
```

3. Enter the defaults for the remaining queries:

```
Command to start port monitor:[/usr/lib/saf/ttymon]↵
Version number: [1] ↵
Initial run state: [STARTED] ↵
Start state: [ENABLED] ↵
Restart count: (0-10) [3] ↵
File name of configuration script: ↵
```

4. Enter a comment to document the use of this port monitor. For example:

```
Comment: Port monitor to control MIS modems ↵
```

5. Confirm the operation. For example:

```
OK to perform operation? [yes] ↵
Adding default port monitor, ttymon1 ...
Port monitor ttymon3 has been added
```

Creating a port service for a modem

After you have created a port monitor, you must assign to it a port service for each modem port you want it to monitor for incoming calls.

Follow this path through **sysadm** to create a port service for a modem:

Device → Port → Port Service → Add

Sysadm guides you through a series of prompts. At each prompt, press Enter (↵) to accept the displayed default, or enter a new value.

1. The first prompt asks for the name of the controlling port monitor to which you are assigning services. Choose a **ttymon** name. Entering **?** lists the currently defined port monitors. For additional information about port monitors, see *Managing the DG/UX™ System.*

```
Controlling port monitor for service: [tcp (listen)]ttymon3↵
```

2. Supply a unique descriptive tag that identifies the service. The tag can consist of up to 14 alphanumeric characters.

```
Port service tag: tty06modem ↵
```

3. Supply the user login name of the owner of the service process. If you are not using UUCP but are just using the modem for dial-in, you will usually accept the **root** default. If you are using UUCP on this modem, you should enter the user name **uucp**.

   ```
   Service Userid: [root] ↵
   ```

4. The next prompt asks whether you want to create a **utmp**(4) entry for the service whenever it is invoked. If the service has a **utmp** entry, information about it will be available to commands such as **who**(1), **write**(1), and **login**(1). For normal login services you will always choose the default (yes).

   ```
   Create utmp entry? [yes] ↵
   ```

5. Supply the absolute pathname of the file that contains the port service configuration script, if any.

   ```
   File name of configuration script: ↵
   ```

   If you specify a script name when the service is added, the contents of this file are copied to the port service configuration script file. The port service reads configuration files only when it starts. Changes to a script file made while the service is running will not be in effect until you restart the service. For more information on configuration scripts, see the **doconfig**(3N) manual page.

6. You are presented the option to supply a description of the port service.

   ```
   Comment: Tty modem port. ↵
   ```

   What you enter here is displayed when you select an operation that lists services assigned to the port monitor.

7. Specify the initial state of the port service when added. For a modem, accept the *ENABLED* default.

   ```
   Initial state: [ENABLED] ↵
   ```

8. You are prompted for the port service version number. Accept the default value, *1*.

   ```
   Version number: [1] ↵
   ```

9. Specify absolute pathname of the ports device file (for example, **/dev/tty06**) that is associated with the service.

   ```
   Path name of terminal device: /dev/tty06 ↵
   ```

10. You are prompted for the label from the **/etc/ttydefs** file to use for setting the initial termio (terminal I/O) settings. The **ttydefs** labels appear as the first field in each line of the **ttydefs** file. For modem lines, use the label that starts with M and is followed by the modem speed. For a list of choices, enter a question mark (?). For a 2400-baud modem line, use M2400.

    ```
    TTY Definition Label: [9600] M2400 ↵
    ```

11. Specify the absolute pathname of the service to be invoked following a successful modem connection. In most cases, you will choose the default, **/usr/bin/login**.

    ```
    Service command: [/usr/bin/login -s login -a device -d %d] ↵
    ```

12. Select whether or not to force a hangup before initializing the line. You usually can select the default.

    ```
    Hangup? [yes] ↵
    ```

13. Specify whether the **ttymon** port monitor should invoke the service without displaying a prompt as soon as it receives a carrier indication. Answer yes only if you are sure of the baud rate and that no prompting is necessary; otherwise choose the default.

    ```
    Connect on Carrier? [no] ↵
    ```

14. Specify whether to use the port is used for calling out and for receiving incoming calls. Answer yes to make a port bidirectional, allowing outside users to connect to the port and, when the port is free, allowing programs such as **uucico**, **cu**, and **ct** to use it for dialing out.

    ```
    Bidirectional? [no] yes ↵
    ```

15. The next prompt asks for the number of new-line characters for which **ttymon** waits before sending out the login prompt. If your modem sends out one or more lines of information before it is ready to receive data, enter the number of such lines. If 0, **ttymon** waits for any character before sending the prompt. If you select the default, *none*, **ttymon** displays a prompt without waiting for characters to be typed.

    ```
    Wait-read value: [none] ↵
    ```

16. Specify the time interval in seconds that the port monitor is open and inactive before hanging up.

    If the port monitor detects no typed characters over the line for the specified number of seconds, it terminates the connection. To specify that the line should never time out, specify 0 seconds. Values range from 0 to 600.

```
Timeout: (0-600) [0] 30 ↵
```

17. Enter the message that the port displays when it establishes a connection and enables the port.

```
Prompt message: [login: ] ↵
```

18. Provide a comma-separated list of STREAMS modules that you want to push. After popping all modules already on the stream, these modules are pushed in the order in which you specify them. You should normally accept the default of no modules to be pushed. For more information, see the *Programmer's Guide: STREAMS*.

```
Modules to be pushed: ↵
```

19. Provide a message to be displayed by the port monitor when a user attempts a connection on a disabled port. If your message includes tabs or more than one line, use "\n" and "\t" to specify the tab and new-line characters.

```
Disabled response message: ↵
```

20. Finally, confirm your entries.

```
OK to perform operation? [yes] ↵
Service tty06modem has been added to port monitor ttymon3.
```

The final message informs you that the named port service *modemservice* was successfully added.

Initially setting up the UUCP software

In this section, we set up the basic UUCP software and edit some UUCP files. We describe these tasks in the following order.

- Setting the administrative UUCP password and logging in.
- Learning about and starting the UUCP shell scripts.
- Learning about and editing the **/etc/uucp** files.

Setting the administrative UUCP password and logging in

UUCP comes with two login accounts:

nuucp is the administrative UUCP login name.

uucp is the login name for remote systems to transact UUCP business by modem.

Before proceeding further, set a password for the administrative login, **nuucp**. Do this while logged in as superuser (**sysadm** or **root**). For example:

```
# passwd nuucp ↲
New password: umgr ↲
Re-enter new password: umgr ↲
```

Now log in as follows.

```
# su nuucp ↲
Password: umgr ↲
#
```

The system will not echo your password as you type it.

Starting the UUCP shell scripts

UUCP provides shell scripts that regularly poll remote systems, transfer files, cleans up undeliverable jobs, and cleans up log files. These scripts are as follows:

uudemon.poll
When scheduled, reads the **Poll** file (**/etc/uucp/Poll**). When any system listed in the **Poll** file is scheduled to be polled, places a work file (**C.***filename*) in the **/var/spool/uucp/***node* directory, where *node* is the name of the system to be polled.

uudemon.hour
Calls the **uusched** server to search the spool directories for work files of the form **C.***filename* that have not been processed, and schedules these files for transfer to a remote machine. Calls the **uuxqt** server to search the spool directories for execute files of the form **X.***filename* that have been transferred to your computer and not processed at the time of transfer.

uudemon.admin
Runs the **uustat** command with –p and –q options. The –q option reports on the status of the files that are queued: work files (**C.***filename*), data files (**D.***filename*), and execute files (**X.***filename*). The –p option prints process information for processes listed in **/var/spool/locks**.
Sends resulting status information to the **nuucp** administrative login using the **mail** facility.

uudemon.cleanup Retrieves log files from the **/var/spool/uucp/.Log** directory of individual machines, then merges them and places them in the **/var/spool/uucp/.Old** directory with other old log information.
Removes the following from the spool files: work files, seven days old or older; data files, seven days old or older; and execute files two days old or older.
Returns mail that cannot be delivered to the sender.
Mails a summary of the status information gathered during the current day to the **nuucp** administrative login.

For information about the **uustat** command and the **uusched** and **uuxqt** servers, see Chapter 4.

To start the UUCP shell scripts, at each prompt ($) type the following:

```
$ uudemon.poll ↵
$ uudemon.hour ↵
$ uudemon.admin ↵
$ uudemon.cleanup ↵
```

The UUCP shell scripts are now running, and you are ready to set up the UUCP software. For information about starting these shell scripts automatically when you reboot your system, see Chapter 4.

Setting up remote communications service

Setting up UUCP software to support modem communications consists largely of editing the data files located in the **/etc/uucp** directory. The primary data files are **Devices**, **Systems**, **Poll**, **Dialcodes**, **Dialers**, and **Permissions**.

These and other data files in the **/etc/uucp** directory are described fully in Chapter 4. Most data files contain descriptions of the file task, sample entries that may or may not support your modem, and examples of how to edit the file for your modem and its applications. These files are located in **/etc/uucp** rather than **/usr/lib/uucp** so that clients with shared **/usr** can be configured individually.

Adding devices for remote access

The **/etc/uucp/Devices** file contains entries for dial-out devices used to connect with remote systems. These entries include the location and line speed of automatic call units (modems).

First, use the **sysadm** utility to add a device to the UUCP Devices table; and second, using an editor, such as **vi**, to add a direct link for your modem entry. The direct link lets you talk to the modem without going through UUCP. With this entry in the **Devices** file, you can open a line to the modem, and then issue AT commands to dial out, configure the modem, or read modem status information.

Proceed as follows:

1. Log out of **nuucp** login and log in as **sysadm**.

   In this section, you may move between the **nuucp** and the **sysadm** login IDs a few times. We recommend that you edit those files in the **/etc/uucp** directory that require the use of an editor, such as **vi**, while logged in as **nuucp**, since **nuucp** is the owner of the **/uucp** directory. However, because **nuucp** does not have superuser privileges, you must log in as **sysadm** (or **root**) while editing files using the **sysadm** facility.

2. Invoke the **sysadm** Main Menu by typing **sysadm** (or **asysadm**) at the shell prompt.

3. Execute the following path through the **sysadm** submenus to add a device to the UUCP **Devices** table:

   Networking → UUCP → Devices → Add

   Sysadm responds as follows.

   ```
   Running subcommand 'adddevice' from menu 'uucpmgmt',
   UUCP Management
   TTY Number? nn ↵
   ```

   Enter the same **tty** number for this modem device that you previously entered in step 3 of the “Setting Up the DG/UX System” section. For example, if you previously entered **tty01**, you would enter **01** here. This entry must not pre-exist in the UUCP **Devices** file.

   ```
   Local Modem Type? [hayes] ↵
   ```

   If you are using a modem other than `hayes`, type **?** in response to the above prompt. The system responds by listing the modems supported by UUCP. If your modem is listed, enter the modem type verbatim. If your modem is not listed, enter a modem type that identifies your modem to the system.

   ```
   Device entry for ttynn has been added.
   Do you want to create another device entry? [yes] n↵
   ```

   We responded with an **n** (no) to this question, since it is usually easier to configure one modem device at a time.

4. Exit (`quit`) from the **sysadm** utility, log out of **sysadm**, and log in as **nuucp**.
5. Using an editor such as **vi**, open the **Devices** file:

```
# vi /etc/uucp/Devices ↵
```

6. Move the cursor to the end of the `ACU tty`*nn* descriptions shown in the sample below, where *nn* is the number of the **tty** line you are configuring.

IMPORTANT: Each time you use **sysadm** to add a `hayes` modem to the UUCP device table, the system automatically adds the following descriptions to the end of the **/etc/uucp/Devices** file for the designated **tty** line.

```
ACU ttynn,M - 300 hayes
ACU ttynn,M - 1200 hayes
ACU ttynn,M - 2400 hayes
ACU ttynn,M - 4800 hayes
ACU ttynn,M - 9600 hayes
```

7. Add the following text directly below the ACU tty*nn* line descriptions for the appropriate line (`tty`*nn*):

```
Direct ttynn,M - baud-rate direct
```

See the example below. If you are setting up a 2400-baud modem on **tty01**, your edited **Devices** file will resemble the following sample.

```
ACU tty01,M - 300 hayes
ACU tty01,M - 1200 hayes
ACU tty01,M - 2400 hayes
ACU tty01,M - 4800 hayes
ACU tty01,M - 9600 hayes
Direct tty01,M - 2400 direct
```

8. Save and close the file.

Establishing links to remote computers

The **/etc/uucp/Systems** file contains information needed by the UUCP software and the **cu** program to establish a link to a remote computer. It contains information such as the name of the remote computer, when the computer can be reached, the telephone number of its modem, and the login name and password expected by the remote system.

To add to the **Systems** file a system with which you want to communicate, follow these steps:

1. Log in as **sysadm** and invoke the **sysadm** Main Menu by typing **sysadm** (or **asysadm**) at the superuser (#) prompt.
2. From the **sysadm** Main Menu, execute the following path through the submenus to add a UUCP system.

   Networking → UUCP → Systems → Add

   Enter the name of the UUCP host system with which you want to communicate:

   ```
   Running subcommand 'addsystem' from menu 'uucpmgmt', UUCP Management
   System Name? NEsales ↵
   ```

   Enter the speed of the modem you are using.

   ```
   Remote Modem Type? [hayes] ↵
   Modem Speed? [1200] 2400 ↵
   ```

   Enter the telephone number you are calling. For example:

   ```
   Phone Number? 16175551234 ↵
   ```

   Obtain the login name and password (see below) for the host UUCP system you are adding from that system's system or network manager.

   ```
   Login Name? uucp ↵

   Password? uucp ↵
   System entry for NEsales has been added.
   Do you want to create another system entry? [yes] n↵
   ```

3. Exit from the **sysadm** utility.
4. Using an editor, such as **vi**, at the superuser prompt open the **/etc/uucp/Systems** file as follows.

   ```
   # vi /etc/uucp/Systems ↵
   ```

   Your **Systems** file should now resemble the following example (Figure 3–2).

```
#$What: <@(#) Systems.proto,v 4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may,
#        call(e.g., MoTuTh0800-1700).
#        Use "Any" for any day.
#        Use "Never" for machines that poll you,
#        but that you never call directly.
# Type            device type
# Class  transfer speed
# Phone  phone number (for autodialers) or token
#        (for data switches)
# Login  login sequence is composed of fields and
#        subfields in the format "[expect send]
#        ...". The expect field may have subfields
#        in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer
NEsales Any ACU 2400 16175551234 in:--in: uucp word uucp:
# sysadm NEsales modem_name=hayes switch_id= login_name=uucp
passwd=uucp
```

Figure 3–2 Sample Systems file

Notice that the second field in the entry for NEsales, is Any. When the **sysadm** utility adds a system to the **Systems** file, it automatically assigns a default Time value of `Any`. `Any` indicates that your system can call the NEsales system on any day, at any time.

To change the value of the Time field for NEsales in the **Systems** file, use an editor, such as **vi**. For example, to allow your system to call or poll NESales only on Mondays and Thursdays, you would change `Any` to `MoTh`:

```
NEsales MoTh ACU 2400 16175551234 in:--in: uucp word uucp:
```

The `Time` entry in the **Systems** file also affects a system entry in the **Poll** file. See below.

Scheduling routine transfers

The **/etc/uucp/Poll** file identifies the times of day specific systems will be polled by your system. The days a particular system can be polled are identified by the Time value for that system, as entered in the **Systems** file.

Because polling is associated with routine transfers of data between systems at specified time intervals, it is frequently set up in a **crontab** file. For more information about setting up **crontab** files for polling, see Chapter 4 of this manual and *Managing the DG/UX™ System.*

Just as you set up the System file to call a system only on certain days, you set up the **Poll** file to call at certain times on those days. Figure 3–3 shows a **Poll** file set up to poll the NESales system three times a day, at 10:00, 16:00, and 22:00 hours.

Once polling times are defined for a UUCP system in the **Poll** file, polling is initiated on a regular basis, regardless of the availability of data to be sent to or retrieved from the designated system.

```
#     $What: <@(#) Poll.proto,v  4.1.1.3> $
#This file (Poll) contains a list of
#"system <tab> hour1 hour2 hour3 ..." lines for polling remote
systems.
#See examples below.
#
# Lines starting with # are ignored.
# NOTE a tab must follow the machine name
#raven    2  6  10

#quail    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22            23
NEsales   10  16  22

#
```

Figure 3–3 Sample Poll file

Substituting abbreviations for dialing sequences

The **/etc/uucp/Dialcodes** file contains abbreviations of dialing sequences that entries in the **/etc/uucp/Systems** file can use in the phone number field.

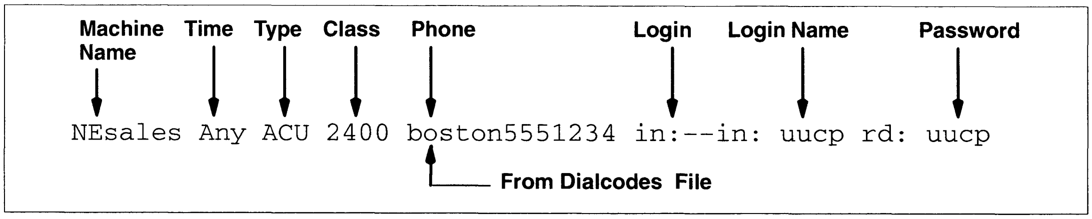
Use an editor, such as **vi**, to edit them in the **/etc/uucp/Dialcodes** file to include the dialing sequences you are using. The **Dialcodes** file explains what to do and supplies examples. See Figure 3–4.

```
#
# The Dialcodes file contains dial-code abbreviations
# that can be used in the 'Phone' field of the Systems
# file. Each entry has the format:
#
# abbr    dial-seq
#
# where 'abbr' is the abbreviation used in the Systems
# file 'Phone' field, and 'dial-seq' is the dial sequence
# that is passed to the dialer when that particular Systems
# file entry is accessed.
#
# Example 1:      chicago 1312
# Example 2:      chicago 9=1312
boston 9=1617
```

Figure 3–4 Sample Dialcodes file

In Example 1 of the **Dialcodes** file shown above, `chicago` is the abbreviation for the dialing sequence 1 plus the area code. Example 2 in the file shown above tells the modem to dial 9 and then wait to receive a secondary dial tone (=) before dialing the rest of the number. The actual entry creates an abbreviation named **boston.**

Looking at the associated field in the **/etc/uucp/Systems** file, you would see our sample addition to the **Dialcodes** file (**`boston 9=1617`**) as an entry such as the following in the **Systems** file.



```
NEsales Any ACU 2400 boston5551234 in:--in: uucp rd: uucp
```

Giving instructions for dialing other modems

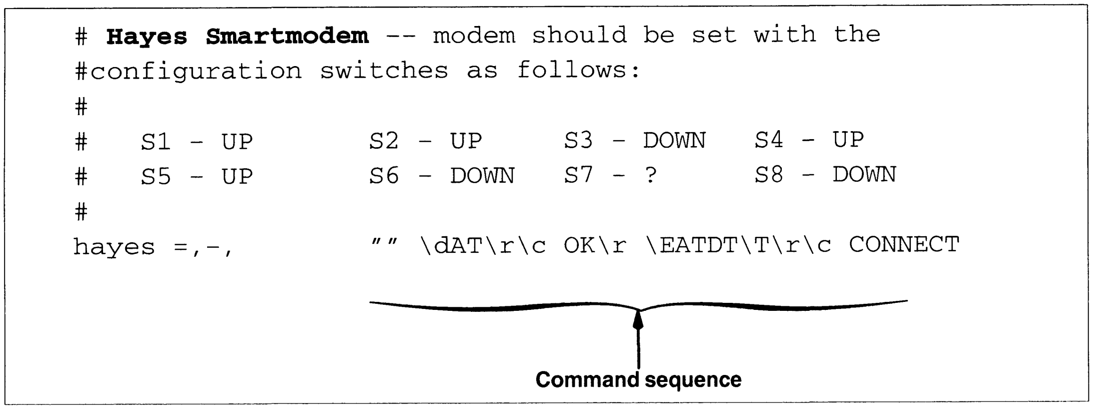
The **/etc/uucp/Dialers** file contains sequences of commands required to connect to remote computers or terminals. This file contains sample chat scripts. A chat script comprises modem command sequences that the **uucico** program sends to a modem when trying to connect to a remote system. The **uucico** program is the server that handles connection tasks and transfers files.

The **Dialers** file includes instructions and examples for editing the file to support your application. Assume that you have a Hayes Smartmodem; use an editor, such as **vi**, to modify the command sequence for your modem entry in **Dialers** file as follows:

1. Open the **Dialers** file:

```
# vi /etc/uucp/Dialers ↵
```

2. Locate the **Hayes Smartmodem** description shown in the example below.



```
# Hayes Smartmodem -- modem should be set with the
#configuration switches as follows:
#
#    S1 - UP       S2 - UP     S3 - DOWN   S4 - UP
#    S5 - UP       S6 - DOWN   S7 - ?      S8 - DOWN
#
hayes =,-,         "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
```

Command sequence

Figure 3–5 Excerpt from Dialers file

The command sequence \d\AT\r\c delays about two seconds, then issues an AT command and a carriage return but no new line.

3. Change the command sequence to \d\EAT\r\c so that the modem will wait for an echo before issuing the next character and will send a **c** after the AT and carriage return.

```
hayes =,-, "" \d\EAT\rc\c OK\r \EATDT\T\r\c CONNECT
```

This modification instructs the UUCP software to wait for any character sent to the modem to be echoed back before sending the next character.

IMPORTANT: Ignore the instructions pertaining to external configuration switch settings (S1, S2, and so on) in the **Dialers** file. Set your modem's switches as described in Chapter 2 of this manual, or as indicated in your modem owner's manual.

Controlling remote access to file systems

The **/etc/uucp/Permissions** file lets you control access to your file system by remote computers and their users with respect to login, file access, and command execution. The **Permissions** file supports two major categories of entries: LOGNAME and MACHINE.

LOGNAME specifies the permissions that take effect when a remote computer logs in to system. Each login ID used by a remote computer must appear in only one LOGNAME entry. Each such entry must be followed by an equals sign (=) and a value.

MACHINE specifies the permissions that take effect when your computer logs in to a remote computer. Each MACHINE entry must be followed by an equals sign (=) and a value.

Setting permissions for access

Table 3–1 describes the options available with LOGNAME and MACHINE entries, indicates the option class, and describes possible values. An equals sign and a value must follow each option.

Table 3–1 Permissions options

Option	Class	Value	Description
REQUEST	LOGNAME and MACHINE	Yes	Remote computer can transfer files from your computer.
		No (default)	Remote computer cannot transfer files from your computer.
SENDFILES	LOGNAME	Yes	Remote user can transfer data previously queued for the user's system.
		No	Remote user cannot transfer queued data.
		Call (default)	Files queued in your computer for a remote machine will be sent only when your computer calls the designated remote computer. Note: When both host and remote computers use the Call value, this option does not work.
READ WRITE	LOGNAME and MACHINE	*Directory names*	Remote computer can read from or write to specified directories. Default for both is **/var/spool/uucppublic**.
NOREAD NOWRITE	LOGNAME and MACHINE	*Directory names*	Specified directories are exceptions to READ and WRITE options. The default is no exceptions.
CALLBACK	LOGNAME	Yes (default = No)	No transactions take place until the calling system is called back by your computer.
COMMANDS	MACHINE	*Commands* (default = **rmail**)	Remote computer can execute specified commands locally. **ALL** is a keyword for all commands.
VALIDATE	LOGNAME	*Host name of remote system*	Used with COMMANDS, this verifies a caller's identity by login/password pair and remote computer name.

The **Permissions** file in Figure 3–6 shows example options for login name **uucp** on machine NESales. **uucp** can read or write in all directories in the / file system except **/etc**, can run commands **rmail**, **mail**, and **lp**, and can send and receive files.

The DG/UX system reads **Permissions** strings as a single line of text terminated by a new-line character. To continue an entry on the next line, use a reverse solidus (\) at the end of the first line. See the last two lines in Figure 3–6.

```
#
# Copyright (C) Data General Corporation, 1984 - 1990
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
# __PassStamp__
#
# $What: <@(#) Permissions.proto,v  4.1.1.4> $

# This entry for public login.
# It provides the default permissions.
# See Managing the DG/UX System for more information.
LOGNAME=nuucp

LOGNAME=uucp MACHINE=NEsales READ=/ NOREAD=/etc WRITE=/ \
NOWRITE=/etc COMMANDS=rmail:mail:lp SENDFILES=yes REQUEST=yes
```

Figure 3–6 Sample Permissions file

Your basic UUCP software is now set up. A good way to check that you have set the files up correctly is to get a description of the files by using the **uucheck** command:

```
/usr/lib/uucp/uucheck -v ↵
```

Communicating with a modem using a direct link

This section describes the basic operating states of a modem and how to establish a direct link with a modem from a host system or remote terminal. It discusses the following topics:

- Putting your modem into command state
- Establishing a direct link to a modem
- Setting the modem's initial operating characteristics
- Viewing the modem's status
- Modifying the contents of the modem's internal registers
- Testing the modem's initial setup
- Exiting from a direct link to a modem

Putting your modem into command state

Modems normally operate in one of two states: idle or active. (They can also operate in a test or diagnostic state.) When turned on and idle, modems are said to be in the command state and can accept AT commands. In this mode data is not sent to the remote modem. When active—transmitting or receiving data—modems are said to be in the connect state, also referred to by some modem vendors as data mode. In this mode the modem will not act on AT commands but treats them as data and sends them to the remote modem.

To implement the instructions in this section, the modem must be in the command state and properly connected to its host computer or terminal. Usually, powering your modem on puts it in the command state.

Establishing a direct link to a modem

In this section, you will establish a direct link to a modem from its host computer system; and second, from a remote terminal.

From a host system

To establish a direct link from your host computer to the modem, type the following command, where *nn* is the number of the modem's **tty** line; for example **01**:

```
# cu -d -l ttynn↵
```

The modem responds by sending a message similar to the following to your console:

```
After establishing $ cu -d -l /dev/tty01
altconn called
Device Type Direct wanted
Trying device entry 'tty01' from '/etc/uucp/Devices'.
processdev: calling setdevcfg(cu, Direct)
opening with O_NDELAY set
fd_mklock: ok
clear O_NDELAY
fixline(4, 2400)
gdial(direct) called
Trying caller script 'direct' from '/etc/uucp/Dialers'.
getto ret 4
ICANON,ECHO are already off
device status for fd=4
F_GETFL=2,iflag='12005',oflag='0',cflag='2273',lflag='0',line='0'
cc[0]='0',[1]='0',[2]='0',[3]='0',[4]='1',[5]='0',[6]='0',[7]='0
call _mode(1)
Turn off iconv/oconv
Connected
transmit started
_receive started
```

If you do not see similar output or if the modem does not respond when you issue an AT command, check the default value of the modem's results codes in your modem's owner's manual. Results codes must be on for the modem to be able to respond. If they are off, turn them on by typing: **ATQ0** ↵

Next, go to the section "Setting Up Your Modem's Initial Operating Characteristics".

If, at any time, you want to return to the DG/UX environment, go to the section "Exiting a Direct Link" for instructions.

From a remote terminal

Before starting, set up the terminal to operate with the following characteristics:

- 8–bit data
- No parity
- The baud rate of your modem; for example, 2400 baud

To establish a direct link from your terminal to the modem, type the following:

```
AT ↵
```

The modem will respond *OK*.

If the modem does not respond, see your modem's owner's manual. Some modems require that you establish a direct link by first typing a character, such as 4, repeatedly until the modem sends an *OK* response. If this does not result in an *OK* response, check the default value of the modem's results codes in your modem's owner's manual. Results codes must be on for the modem to be able to respond. If they are off, turn them on by typing the following on your terminal's keyboard:

```
ATQ0 ↵
```

Setting up the modem's initial operating characteristics

After establishing a direct link with your modem, enter the following AT command string to invoke the modem characteristics listed below in Table 3–2. Check your modem documentation to ensure that the commands listed here are the same your modem uses, as variations in syntax and operation sometimes exist.

```
AT&FS0=1&d2&c1\d1&w ↵
```

IMPORTANT: Be sure to enter the above command string on one line. Otherwise, you may lose your connection to the modem when the Carrier Detect parameter is modified. If you are including AT commands that are not part of the above command string, your command string can be extended beyond one line by simplying typing a reverse solidus character (\) at the end the line being extended. For example:

```
AT&FS0=1&d2&c1\d1&w \ ↵
.......................... \ ↵
.......................... ↵
```

Table 3–2 AT commands

AT command	Description
AT&F	Resets the modem registers to factory default.
s0=1	Sets the modem to answer on one ring.
&d2	When Data Terminal Ready (DTR) from the computer goes low (on to off), the modem hangs up (goes on-hook), enters command state, and accepts AT commands.

Continued

Table 3–2 AT commands

AT command	Description
&c1	The state of Carrier Detect (CD) follows the status of the connection with the remote modem. When there is no connection, CD is off; when a connection exists, CD is on.[1]
\d1	Clear To Send (CTS) follows CD; Data Set Ready (DSR) follows off-hook.
&w	Stores the current configuration in battery-backup memory.

[1] On some modems, when at&c1 is set, CD turns on after the CONNECT message.

Your initial modem characteristics are now set.

Viewing the modem's current status

To view your modem's current status,which is stored in its status registers, type the following command string:

```
AT\S ↵
```

Your modem's response probably will resemble the following display. The modem used for this example is a Microcom, Class MNP/6.

```
IDLE           000:00:00
LAST DIAL      <nnnnnnn>
ID:
MODEM BPS      2400    AT%G0
MODEM FLOW     OFF     AT\G0
MODEM MODE     DIR     AT\N1
AUTO ANS.      ON      ATS0=1
SERIAL BPS     2400    AT
BPS ADJUST     ON      AT\J1
SERIAL FLOW    OFF     AT\Q0
PASS XON/XOFF  OFF     AT\X0

  - STRIKE ANY KEY TO CONTINUE -
```

```
PARITY         8N     AT
BREAK          5      AT\K5
EXIT CHAR      043    ATS2=43
CMD ECHO       ON     ATE1
RESULTS        ON     ATQ0
RESULT TYPE    LONG   ATV1\V0
CONN MNP-      0      AT-M0
DATA ECHO      OFF    AT\E0
INACT TIMER    00     AT\T0
AUTO RETRAIN   ON     AT%E1
COMPRESSION    ON     AT%C1
96 EQUALIZER   A-LO   AT%P2


 - STRIKE ANY KEY TO CONTINUE -


96 SYNC STD    0      AT%M0
96 ECHO SUP    TONE   AT%Z1
MAX BLK SIZE   256    AT\A3
AUTO BUFF      0      AT\C0
AUTO CHAR      000    AT%A0
EMULATING HP   OFF    AT\H0
PAUSE TIME     002    ATS8=2
DTR            2      AT&D2
CARR DET       1      AT&C1
DSR            1      AT\D1
RING IND       1      AT\R1
SPKR CTRL      1      ATM1


 - STRIKE ANY KEY TO CONTINUE -


LEASE LINE     OFF    AT&L0
ASYNC/SYNC     0      AT&M0
CTS/RTS        ON     AT&R0
SIM RING       0      AT:R0
CD DELAY       000    AT:U0
CTS DELAY      000    AT:V0
DSR DELAY      000    AT:X0
DISC DELAY     000    AT%D0
REM CHAR       042    AT*S42
REM ENABLE     OFF    AT*E0
REM SEC        OFF    AT*R0
RDLB ENABLE    ON     AT&T4


 - STRIKE ANY KEY TO CONTINUE -
```

```
DIAL MODE       4       ATX4
PULSE DIAL      US      AT&P0
GUARD TONE      0       AT&G0
ASYNC PROTOCOL  NONE    AT:K0
KERMIT MARK     001     AT:Q1
PAR CHK         OFF     AT-P0
MANUAL DIAL     0       AT:D0
T/D CTRL        0       AT-T0
RUN DIAGS       1       AT$D1
BELL            ON      ATB1

OK
```

For detailed information about each status line displayed by your modem, see your modem's owner's manual.

Modifying the contents of the modem's registers

After viewing the initial settings of your modem's status registers, you may want to modify certain registers to take advantage of any communications enhancements offered by your particular modem. If so, see the AT command set descriptions in your modem's owner's manual that support these enhancements. Then proceed to issue AT commands, beginning each command string with the letters **AT** and ending each command string by pressing Carriage Return (↵).

For example, the above status information indicates that the Auto-Answer characteristic is turned on (`AUTO ANS. ON ATS0=1`) and the modem will answer after one ring. If you want to use this modem line only for *outgoing* calls, turn Auto Answer off by typing the following command string:

ATS=0 ↵

Turning results codes on and off

As indicated in the above display, the modem's results codes are now turned on (`RESULTS ON ATQ0`), enabling you to communicate directly with your modem. This state of the results codes is appropriate for modems that are being used for *outgoing* calls, or those with which you are establishing a direct link. However, if your modem will be used to support *only incoming* calls, turn the results codes off before exiting this direct link by typing the following command: **ATQ1** ↵

Be sure to save your commands in the modem's memory by entering the **AT&W** command.

IMPORTANT: If your modem will be used for both incoming and outgoing calls (bidirectional), it is important to turn result codes off (`ATQ1`) each time an outgoing call terminates, and turn them on (`ATQ0`) each time you want to initiate an outgoing call. This is necessary because the **ttymon** port monitor does not expect results codes from incoming calls on a line that it supports, and, therefore, may inadvertently abort subsequent incoming modem calls.

Testing your modem's initial setup

Before proceeding further, we recommend that you test your modem's setup thus far by using the AT dialing command (**ATD**) to connect to a remote system equipped to handle remote calls using a modem. Note that this method of direct modem communications bypasses the UUCP software on your system; thus, it tests only your DG/UX modem setup. To test your application setup, see Chapter 4.

To begin the test, you will need the telephone number of a remote system's modem that is setup to accept incoming calls via the Public Data Network. Although it is not required to have an account on the remote system (getting the remote system's login banner displayed on your console's screen usually indicates a successful connection), an account there can be helpful to verify that data can be sent to that system.

The following are examples of direct dialing. The **ATD** dial command can be used with one or more modifiers, a few of which are described below. For information about other dial modifiers and how to use them, see your modem's owner's manual.

`ATD 16175551234` This example does not include modifiers. Notice that there are no spaces between the numbers.

`ATDP 16175551234` This example specifies pulse dialing, which is usually the default dialing method.

`ATDT 16175551234` This example specifies touch-tone dialing.

`ATD 9w16175551234` In this example, the **w** modifier inserted between the digits **9** and **1** tells the modem to wait for a second dial tone before dialing the remainder of the number. See your modem's owner's manual for the default maximum wait time and how to modify it, if necessary, for your application.

Now try it with your modem. Once you press Carriage Return (Ctrl-M) to invoke your command sequence, your modem's off-hook indicator (if present) will light and you will hear the dial sounds.

If you do not connect successfully with the remote system, first check to ensure that you dialed the appropriate telephone number, and you included a wait modifier if needed. Then check with the manager of the remote system to ensure that the designated modem is enabled and not busy. If everything checks out, go back and recheck your DG/UX modem setup. Also make sure the modem's results codes are turned on.

Exiting from a direct link with a modem

When you are ready to exit the direct link with your modem and return to the DG/UX environment, first check the state of the modem's results codes and modify their state (on or off), as required to support the modem's usage; that is, to support incoming or outgoing calls.

Now exit the direct link by typing the following:

```
~
```

The system will issue the following disconnect messages, and then display the Bourne-shell prompt ($) of **nuucp.**

```
~[system name]. ↵
call tilda(.)
call _quit(0)
call _bye(0)

Disconnected/
call cleanup(0)
call _mode(0)
Turn on iconv/ocon
$
```

You are now in the DG/UX environment.

If your modem is attached to a remote terminal, you may need to reset your modem in order to exit the link to your modem. For more information, see your modem's owner's manual.

What to do next

The DG/UX system and the basic UUCP software are set up to support your modem. You are now ready to set up specific UUCP software configurations to meet the needs of your modem applications.

End of Chapter

4 Setting up UUCP to support modem applications

This chapter describes how to set up UUCP to support asynchronous modem applications. It also discusses the major UUCP components and gives examples. The chapter contains the following major topics:

- UUCP setup overview
- UUCP commands
- UUCP servers
- UUCP directories and data files
- Examples of UUCP configurations to support modem applications
- Customizing UUCP data files

UUCP setup overview

This section gives an overview of UUCP setup. It contains the following subsections:

- Understanding the UUCP components
- Overall setup steps
- Starting the UUCP shell scripts

Understanding the UUCP components

The UUCP system consists of multiple files and programs. The five main configuration files, located in **/etc/uucp**, are **Systems**, **Poll**, **Devices**, **Dialers**, and **Permissions**. You can connect only to the remote systems listed in **Systems**. You can do this from the command line for an immediate connection, or you can connect and transfer files automatically at the times set in the **Poll** file. The **uudemon.poll** shell script reads the **Poll** file and initiates connections. Files queued for transfer are exchanged via the modems and devices listed in the **Devices** file. The entries in **Devices** need data from **Dialers**. Finally, the **Permissions** file restricts a remote host's ability to request and receive files. The default **Permissions** file is set up to provide the maximum amount of security. You can use the **uucheck –v** command to see exactly what your default permissions are. If you wish to change them, see "Permissions file" under "Customizing UUCP data files" later in this chapter.

Before you can put this system to work, you must have a location with which you wish to set up file transfer connections. This means you will have to contact the system administrator of a remote site and exchange certain information: passwords, system node names, baud rates, and phone numbers. With this information, you are ready to set up the UUCP files.

If you administer UUCP without **sysadm**, use the **nuucp** login ID because it owns the UUCP files and the spooled data files. The other UUCP login ID is **uucp**, which UUCP systems on remote hosts use when they need to transact UUCP business with another machine. Instead of starting a shell like a normal user's login, the **uucp** profile starts the **uucico** program. For more information on administrative logins, see *Managing the DG / UX™ System*.

Overall setup steps

We recommend that you set your UUCP facility up in the following order:

1. Start the **uudemon.poll**, **uudemon.hour**, **uudemon.admin**, and **uudemon.cleanup** shell scripts.
2. Add devices with the operation Networking –> UUCP –> Devices –> Add.
3. Add systems with the operation Networking –> UUCP –> Systems –> Add.
4. Set up lines for dialing in and out with the operation Devices –> Port –> Port Services –> Add.
5. Add poll entries with the operation Networking –> UUCP –> Polling –> Add .
6. Test your connections with the operation Networking –> UUCP –> Test.

Starting the UUCP shell scripts

Your first step in setting up UUCP is to start several important shell scripts:

uudemon.poll Reads the **Poll** file (**/etc/uucp/Poll**) as scheduled.

If any of the systems in the **Poll** file are scheduled to be polled, places a work file (**C.***file_name*) in the directory **/var/spool/uucp/***node-name* (where *node-name* is the name of the system to be polled).

uudemon.hour

Calls the **uusched** program to search the spool directories for work files (**C.***file_name*) that have not been processed, and schedules these files for transfer to a remote machine.

Calls the **uuxqt** server to search the spool directories for execute files (**X.***file_name*) that have been transferred to your host and were not processed at the time they were transferred.

uudemon.admin

Runs the **uustat** command with **–p** and **–q** options. The **–q** option reports on the status of work files (**C.***file_name*), data files (**D.***file_name*), and execute files (**X.***file_name*) that are queued. The **–p** prints status information for processes listed in **/var/spool/locks**.

Sends resulting status information to the **nuucp** administrative login via **mail**(1).

uudemon.cleanup

Takes log files for individual machines from the directory **/var/spool/uucp/.Log**, merges them, and places them in the directory **/var/spool/uucp/.Old** with other old log information.

Removes from the spool directory work files seven days old or older, data files seven days old or older, and execute files two days old or older.

Returns mail that cannot be delivered to the sender.

Mails a summary of the status information gathered during the current day to the **nuucp** administrative login.

To run the scripts on a regular basis, schedule them as **cron** jobs to be run under the username **nuucp**. Use the **su**(1) command to change your username to **nuucp**. Then use **crontab** to schedule the UUCP jobs. The recommended **cron** jobs for UUCP are in the file **/admin/crontabs/uucp.proto**. For more information on **cron**, see *Managing the DG/UX™ System.*

UUCP commands

Table 4–1 summarizes the UUCP administrative programs. Table 4–2 summarizes the UUCP user programs. The “Arguments” column lists the most frequently used arguments and options.

For detailed information about UUCP commands and a complete listing and description of their arguments and options, see the individual man pages. Also see the man page for **rmail**, which UUCP supports.

Administrative commands

The administrative commands are located in **/usr/lib/uucp**, except for **uulog** and **uuname**, which are located in **/usr/bin**.

Table 4–1 UUCP administrative programs

Program	Arguments	Description
uulog	**–s***sys*	Queries a log file of **uucp** and **uuxqt** (a UUCP server) transactions. With the **–s***sys* option, prints information about file transfers relating to the system identified by *sys*. A log file is created for each remote host with which your host communicates. The log files contain records for each use of **uucp**, **uuto**, and **uux**. This command is not available through **sysadm**.
uuname	**–c** **–l**	Lists the names of systems known to UUCP. With the **–c** option, lists the names of systems known to cu. With the **–l** option, displays the local system’s name. You can do this from the command line or via the List operation on **sysadm**’s UUCP → Systems menu.
uucleanup	See the **uucleanup** man page.	Cleans up the spool directory as follows: informs the requester of send/receive requests for systems that cannot be reached; returns mail that cannot be delivered to the sender; deletes or executes **rnews** or **rnews**-type files, depending on where the news originated. Also notifies users of requests that have been waiting for a specified number of days. This command is normally run from a shell script called **uudemon.cleanup**, which is started by the **cron** facility.

Continued

Table 4–1 UUCP administrative programs

Program	Arguments	Description
uucheck	**–v**	Checks for the presence of directories, programs, and support files required by UUCP. With the **–v** option, displays the current permissions for your system. This command is not available through **sysadm**.
Uutry	**–x***n* *system name*	Tests connections between computers and displays messages on failed and successful sessions. Invokes the **uucico** server to establish a communication link between your host and the remote host you specify. With the **–x***n* option, performs debugging at the level indicated by n. You can also initiate the testing performed by **Uutry** by selecting the “Test” operation listed on the UUCP submenu of **sysadm**.

User commands

The user programs for UUCP are in **/usr/bin**. No special permission is needed to use these programs.

Table 4–2 UUCP user programs

Program	Arguments	Description
ct	*telephone number*	Calls the specified telephone number of a modem connected to remote terminal. When the modem answers, the SAF facility spawns a login process. If more than one telephone number is specified, **ct** tries each in succession until the remote modem answers. The user of the remote terminal may call the host and request that it call the remote terminal back. The host will hang up the initial link to the terminal so that it will be available to answer the call back. This is similar to making a collect call. For **ct** to work, the remote terminal’s modem must be set up with auto-answer enabled (on).
cu	*telephone number*, *system name*	Connects your host to a remote host and allows you to be logged in on both hosts at the same time. You can run commands on either host without dropping the communication link. See the **cu**(1) man page for handling of **cu** data preceded by a tilde (~) character.

Continued

Table 4–2 UUCP user programs

Program	Arguments	Description
uucp	*source files*, *destination file*	Copies files from one host to another. It creates work files and data files, queues the job for transfer, and calls the **uucico** server, which in turn tries to contact the remote host.
uudecode	*filename*	Decodes a file encoded by **uuencode** and recreates the original file.
uuencode	*filename*	Encodes a binary file to be sent via **uucp** or **rmail**.
uupick	**d** **m** [*dir*]	Retrieves the files placed under **/var/spool/uucppublic/receive** when files are transferred to a host using **uuto**. With the **d** option, deletes a file entry; with the **m** [*dir*] option, moves the file entry to the specified directory (current directory is the default directory).
uustat	**–a**, **–q**	Displays the status of requested file transfers (**uucp**, **uuto**, or **uux**). It also lets you control queued transfers. You can specify only one **uustat** option at a time. The **–a** option lists all jobs in the queue; **–q** lists jobs queued for each machine.
uuto	*source files*, *destination*	Copies files from one host to a public spool directory (**/var/spool/uucppublic/receive**) on another host. Unlike **uucp**, which lets you copy a file to any accessible directory on the remote host, **uuto** places the file in an appropriate spool directory and tells the remote user to pick it up with the **uupick** program. The destination field has the form *system***!***username*.
uux	See the **uux**(1) man page.	Creates the work, data, and execute files needed to execute commands on a remote host. The work file contains the same information as work files created by **uucp** and **uuto**. The execute files contain the command string to be executed on the remote host and a list of the data files. The data files are those files required for the command execution. For security reasons, most systems permit only incoming **rmail** commands to be executed.

UUCP servers

Server programs are routines that run as background processes and perform system-wide public functions. The **uucico**, **uuxqt**, and **uusched** servers described below handle UUCP file transfers and command executions.

uucico

The **uucico** server performs the following tasks:

- Selects the device (in this case, a modem; but it can be a local asynchronous port connected directly by cable to the asynchronous port of another system) used for the link
- Establishes the link to the remote host
- Performs the required login sequence and permission checks
- Transfers data and executes files, logs results, and notifies the user by mail (**rmail**) of transfer completions

When the local **uucico** server calls a remote system, it interacts with the **uucico** server on the remote system during the session.

After the required files have been created, the **uucp**, **uuto**, and **uux** programs execute the **uucico** server to contact the remote host. The **uusched** server and **Uutry** program also execute the **uucico** server.

uuxqt

The **uuxqt** server executes remote execution requests by searching the **/var/spool/uucppublic** directory for execute files (**X.***file*) sent from a remote host. When an execute file is found, **uuxqt** opens it to get the list of data files required for the execution. If the files are present and accessible, **uuxqt** checks the **Permissions** file to verify that it has permission to execute the requested command.

The **uudemon.hour** shell script, which is started by the **cron** utility, runs the **uuxqt** server.

uusched

The **uusched** server schedules the queued work in the **/var/spool/uucppublic** directory and randomizes the order in which remote host will be called. Then it starts the **uucico** server.

The **uudemon.hour** shell script, which is started by the **cron** utility, runs the **uusched** server.

UUCP directories and data files

This section lists and describes the directories and data files needed to run UUCP. It also discusses using **sysadm** to modify data files.

UUCP directories

The UUCP directories are as follows:

/usr/bin Contains UUCP user programs.

/usr/lib/uucp Contains the executable files for the UUCP system.

/usr/spool/uucp The HOME directory for the **uucp** login.

/etc/uucp Contains the files that make up the UUCP database.

/var/spool/locks Contains lock files for UUCP devices.

/var/spool/uucp Contains directories for administrative purposes and for storing log and status information. The spool directory for queued work that is to be processed by UUCP servers.

/var/spool/uucppublic Stores work that has been sent to your host. The public directory for UUCP.

UUCP data files

The data files for UUCP are in the directory **/etc/uucp**. You can make all changes to the **Devices**, **Systems**, and **Poll** files through **sysadm**'s UUCP menu. You will need to edit the others manually. UUCP has the following data files:

Devices Contains information concerning the location and line speed of the automatic call unit, direct links, and network devices.

Contains entries for the dial-out devices or modems used to connect with remote systems. Each **Devices** entry contains fields for the device location (port pathname in the **/dev** directory) and line speed of the attached automatic calling unit or modem, if used, or information about other types of connecting devices.

Dialers Contains character strings required to negotiate with network devices (automatic calling devices) to establish connections to remote hosts (non-801-type dialers). This file contains some sample chat scripts. A chat script is a series of modem commands and strings that **uucico** sends to a modem to initiate a connection (or attempt to connect) with a remote system.

Poll Defines hosts that are to be polled by your system and when they are polled.

This database, **/etc/uucp/Poll**, contains entries for the remote systems that your system needs to call to initiate UUCP transfers. Polling is necessary for contacting sites that do not have hardware, such as modems, necessary for calling out to other systems. Each Poll entry contains fields for the remote system name and the times when calling should take place.

The Polling menu provides operations for adding, deleting, modifying, and listing Poll entries.

Systems Contains information needed by the **uucico** server and the **cu** program to establish a link to a remote host. It contains information such as the name of the remote host, the name of the connecting device associated with the remote host, when the host can be reached, telephone number, login ID, and password.

This database, **/etc/uucp/Systems**, contains entries for remote hosts that you want to contact using UUCP. Each entry contains fields for the remote system name, the name of the device used to connect to the remote system, the times during which you may reach the remote system, the telephone number of the remote system's modem (if access is via modem), and login information.

Dialcodes Contains dial-code abbreviations that may be used in the phone number field of **Systems** file entries.

Permissions Defines the level of access that is granted to hosts when they attempt to transfer files or remotely execute commands on your host.

Sysfiles Assigns different or multiple files to be used by **uucico** and **cu** as **Systems**, **Devices**, and **Dialers** files.

Using sysadm to modify data files

You can modify the **Devices**, **Poll**, and **Systems** files via the Devices, Polling, and Systems menus under Networking → UUCP. Each of these three menus provides operations for adding, deleting, modifying, and listing entries.

To change other UUCP data files, you can use a text editor such as **vi**.

Examples of UUCP configurations to support modem applications

This section discusses your modem-line configurations. It also describes examples of how to use UUCP commands and set up UUCP files to support the typical modem applications listed below.

Example 1	Setting up remote terminal service
Example 2	Setting up systems for peer-to-peer communications
Example 3	Setting up systems in master / slave roles

Configuring modem lines for optimum service

If your system will support more than one modem, you should first estimate your modem traffic; that is, the number and duration of incoming and outgoing calls the modems will handle on a daily basis, including peak hours. Then determine how best to configure the service (incoming, outgoing, or bidirectional) each modem will supply to achieve optimum utilization of your resources.

When your system's modems will support both incoming and outgoing modem calls, we recommend dedicating some of your modems' terminal lines only to incoming calls; others only to outgoing calls. Of course, the mix of incoming and outgoing lines must match the needs of your modem applications, which, in turn, influence modem traffic patterns. Further, as traffic patterns change, you can easily modify the **ttymon** port service supplied to each terminal line. In fact, if you have a limited number of modems to support your traffic, this might be required to meet the needs of your peak traffic hours for particular applications.

You must select bidirectional service in order to support outgoing modem calls on a particular terminal line. If you choose to use a modem line for bidirectional modem services, we suggest you write a script that will automatically turn result codes on before placing outgoing calls on this line, and will turn them off each time an outgoing call terminates.

As you configure your UUCP applications, we suggest you create and maintain a worksheet similar to that appearing in Figure 4–1 (or make photocopies of Figure 4–1). You can also use this information to determine the hours when polled data transfers can occur with the least impact on other modem services.

Local System's Modems					Remote Systems' Modems		
tty No. & Baud Rate	Modem Tel. No.	Direction			Hostname	Tel. No.	Poll Time
		In	Out	Bi			

Figure 4–1 Sample worksheet: modem configurations

Example 1: Setting up remote terminal service

Figure 4–2 shows a typical remote terminal configuration. In this example, four dedicated modem lines are connected to a DG/UX host system for incoming remote-terminal service and two for outgoing remote-terminal service. For this reason, the host's modems supporting outgoing calls will have their results codes turned on while the host's modems supporting incoming calls will have their results codes turned off.

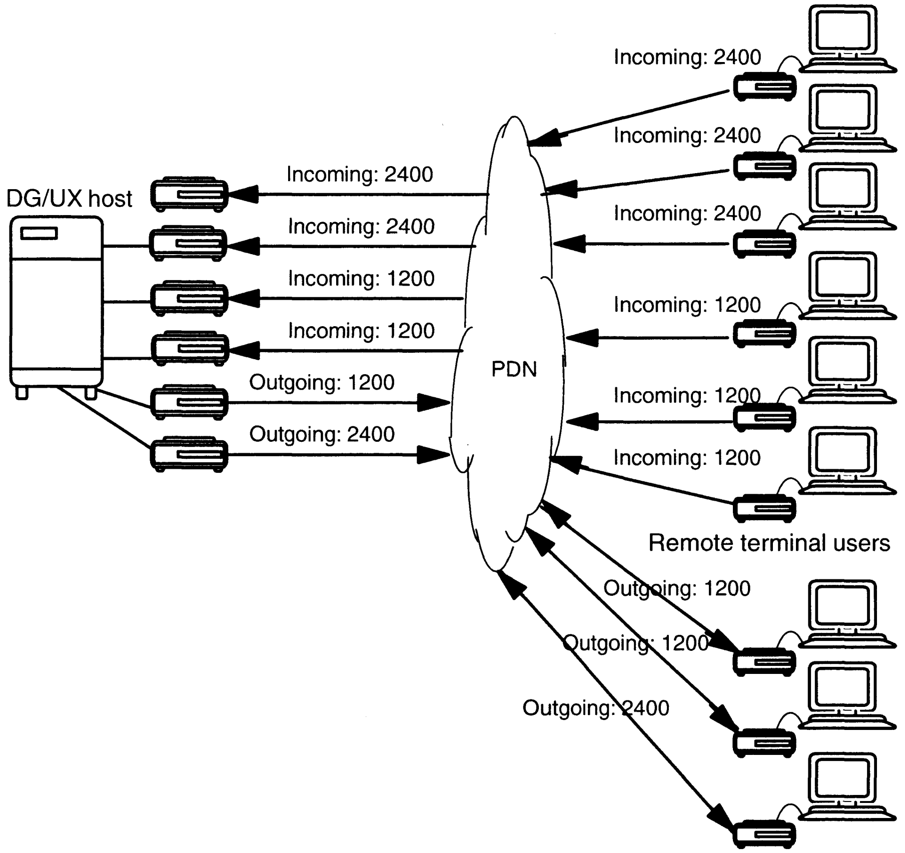


Figure 4–2 Typical remote terminal configuration

For older modems, the baud rate of a remote terminal and modem must match that of the selected incoming or outgoing modem line on the host system. If you are using constant speed interface modems, the only requirement is that the baud rate between a modem and the local computer or terminal be the same.

Setting up UUCP

All modems and their terminal lines must be supported with the appropriate **tty** number and direct modem line entries in the **/etc/uucp/Devices** file. For incoming calls only, the username and password must be known to the host DG/UX system.

Setting up remote modems

If remote terminal service is initiated by a UUCP command executed on the host DG/UX system, the receiving modem's auto-answer feature must be turned on.

Setting up remote terminal users

Before remote terminal users can initiate access to your system via modem, you must supply the following to each of them:

- User account on your system, together with a login name and password.
- List of telephone numbers and baud rates of available incoming and outgoing modem lines.
- Number of new-line characters they must type before a login prompt will be displayed on their terminal's screen.
- Time period the host system will wait for a remote user to complete the login process before dropping the line.

Calling the host from a remote terminal

A user of a remote terminal calls its remote host system by typing on the terminal keyboard the command **ATD** followed by the telephone number of the incoming modem line on the host system. For example:

```
ATD 18001234567↵
```

The baud rate of the modem line selected on the DG/UX host must be the same as that of the remote terminal and modem initiating the call.

When the remote system displays its login banner, the user logs in. Terminating a remote session with the host is the same as terminating a local user session. When the session ends, the DG/UX system drops the line.

After remote terminal users log in to their host system, they can immediately initiate callback service to their terminals. See "Using the **ct** command for callback service" below.

Calling a remote terminal from the host

UUCP supplies two commands for calling remote terminals: **ct** and **cu**.

The **ct** command dials the telephone number of a modem attached to a remote terminal, as shown in the following command string,

ct [**-s** *baud_rate*] *telephone_number*

The **–s** option in the command string above must indicate the baud rate of an existing outgoing modem line. The default baud rate is 1200. See the **ct**(1) man page for information about other **ct** options.

On connection, **ct** displays the host's login prompt. Once the user logs in, the system responds to input from a remote terminal in the same manner as local input. When a remote-terminal user logs out of the DG/UX system, **ct** displays the following prompt: `Reconnect?` If the user responds by typing the letter **n**, **ct** drops the line; otherwise, **ct** again displays the login prompt.

Using the ct command for callback service — After dialing in to a remote host and executing the login process, remote-terminal users can initiate callback to their modems by simply typing the **ct** command string shown above, and then immediately logging off.

The **cu** command dials the telephone number of a modem attached to a remote terminal, as shown in the command string below. Notice that this command string can also be used to dial the telephone number of a remote DG/UX or foreign system.

cu [**-s** *baud_rate*] *telephone_number*

The baud rate of the modem line selected on the DG/UX host must be the same as that of the remote terminal and modem receiving the call. The default baud rate is "Any." Any selects the first available modem **tty** line listed in the **/etc/uucp/Devices** file. See the **cu** man page for more information about **cu** options.

After making the connection, **cu** runs two processes: transmit and receive. The **cu** transmit process reads data from the standard input (calling process) and, except for lines beginning with a tilde (~), passes it to the remote terminal. The **cu** receive process accepts data from the remote terminal and, except for lines beginning with a tilde, passes the data to the standard output.

To terminate a **cu** session, type the following: **~.**

For more information about using the **cu** command and the meanings of data preceded by a tilde, see the **cu** man page.

Example 2: Setting up systems for peer-to-peer communications

Figure 4–3 shows one incoming line and one outgoing line on each of two DG/UX systems to support peer-to-peer communications between them.

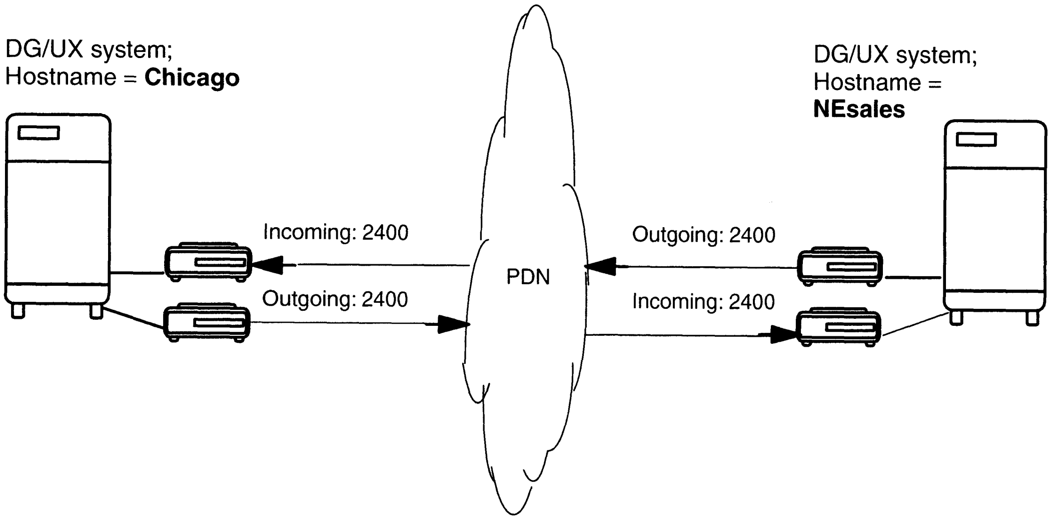


Figure 4–3 Peer-to-peer UUCP communications between remote DG/UX systems

In this example, assume that the modems supporting these asynchronous lines are newly installed and operating at 2400 baud. For this reason, before proceeding further, test the modem lines on both systems, using the outgoing modem line on each system to call the incoming modem line on the other system.

Begin the test on both systems by using the following **cu** command to directly access the local modem configured for outgoing service.

```
cu -d -l ttynn ↵
```

When the local modem responds in command state, issue the following **AT** command to initiate a call to the modem configured for incoming service on the remote system.

```
ATDtelephone_number ↵
```

When both receiving systems respond to the call by displaying a login prompt, you know the modems at both ends of the communications circuit are operating properly.

Adding remote systems to the /etc/uucp files

To set up peer-to-peer communications between two DG/UX systems, called Chicago and NEsales, you must add information in the following **/etc/uucp** files on both systems: **Dialcodes**, **Poll**, **Permissions**, and **Systems.**

Adding dialing sequences on the Chicago host — As shown below, this file contains an entry for dialing the area code for Boston, Massachusetts; namely, 1617. Because the modems on the Chicago host dial directly to an outside line, this example does not include coding for secondary dialing.

```
#  (On Chicago host)
# The Dialcodes file contains dial-code abbrevia-
# tions that can be used in the 'Phone' field of
# the Systems file. Each entry has the format:
#
#                abbr    dial-seq
#
# where 'abbr' is the abbreviation used in the Sys-
# tems file Phone field, and 'dial-seq' is the dial
# sequence that is passed to the dialer when that
# particular Systems file entry is accessed.
#
# Example 1:     chicago   1312
# Example 2:     chicago 9=1312
boston 1617
```

Adding dialing sequences on the NEsales host — As shown below, this file contains an entry for dialing the area code for Chicago, Illinois; namely, 1312. Because the modems on the NESales host must employ secondary dialing to access an outside line, this example includes the dialcode for secondary dialing (**9=**).

```
#   (On NEsales host)
# The Dialcodes file contains dial-code abbrevia-
# tions that can be used in the 'Phone' field of
# the Systems file. Each entry has the format:
#
#               abbr    dial-seq
#
# where 'abbr' is the abbreviation used in the
# Systems file 'Phone' field, and 'dial-seq' is the
# dial sequence that is passed to the dialer when
# that particular Systems file entry is accessed.
#
# Example 1: chicago   1312
# Example 2: chicago 9=1312
chicago  9=1312
```

Specifying polling times on the Chicago host — As shown below, this file contains an entry for the NESales system, indicating that the Chicago system will poll (for example, transfer data to and/or retrieve data from) the NESales system at the following time intervals: 10:00 a.m., 4:00 p.m., and 10:00 p.m., Chicago time; 11:00 a.m, 5:00 p.m., and 11:00 p.m., Boston time. The days on which the polling will be initiated is determined by the `Time` entry for NEsales in the **Systems** file on the Chicago host.

Although polling is initiated at the times indicated in the **Poll** file, the actual time the calling system completes the polling task may vary, based on the availability of a suitable outgoing modem line; that is, one operating at the appropriate baud rate, and the busy or idle state of the incoming modem line on the system to be polled.

```
   (On Chicago host)
# $What: <@(#) Poll.proto,v  4.1.1.3> $
# This file (Poll) contains a list of
# "system <tab> hour1 hour2 hour3 ..." lines for polling remote
# systems. See examples below.
#
# Lines starting with # are ignored.
# NOTE a tab must follow the machine name

#raven 2  6  10
#quail 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
NEsales 10  16  22
```

The Poll file on the NEsales host — As shown below, this file does not contain an entry for polling. For this reason, the NEsales system is not set up to poll the Chicago system.

```
   (On NEsales host)
# $What: <@(#) Poll.proto,v  4.1.1.3> $
# This file (Poll) contains a list of
# "system <tab> hour1 hour2 hour3 ..." lines for polling remote
# systems. See examples below.
#
# Lines starting with # are ignored.
# NOTE a tab must follow the machine name

#raven 2  6  10
#quail 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

Setting permissions on the Chicago host — As shown below, the **Permissions** file on the Chicago system contains two entries.

The first entry lets the NEsales system log in to the Chicago system with the LOGNAME of **NEuucp**. Once logged in using the appropriate password (see the "Setting UUCP passwords..." section below), the NEsales system can send files to and retrieve files from the **/var/spool/uucppublic** directories on the Chicago system.

The second entry not only gives the Chicago system complete access to the NEsales file systems but also lets it execute any and all commands on the NEsales system.

```
   (On Chicago host)
#
# Copyright (C) Data General Corporation, 1984 - 1990
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
# __PassStamp__
#
# $What: <@(#) Permissions.proto,v  4.1.1.4> $

# This entry for public login.
# It provides the default permissions.
# See Managing the DG/UX System for more information.

LOGNAME=nuucp

LOGNAME=NEuucp MACHINE=NEsales READ=var/spool/uucppublic/ \
WRITE=/var/spool/uucppublic COMMANDS=rmail:mail:lp \
SENDFILES=yes REQUEST=yes

MACHINE=NESALES LOGNAME=major READ=/ WRITE=/ COMMANDS=ALL \
SENDFILES=yes REQUEST=yes
```

Setting UUCP passwords on the Chicago host — After identifying the "LOGNAME" used by the NEsales system when calling the Chicago system, you must set a password for it as shown below. You must also provide the "LOGNAME" and its password to the UUCP administrator on the NEsales system.

```
# passwd NEuucp ↵
# New passwd: NEuucp ↵
# Re-enter new password: NEuucp ↵
```

Setting permissions on the NEsales host — As shown below, the **Permissions** file on the NEsales system also contains two entries. Each complies with the entries in the **Permissions** file on the Chicago system. Unlike the **Chicago** system, **NEsales** does not have default permissions for the public login.

```
   (On NEsales host)
#
# Copyright (C) Data General Corporation, 1984 - 1990
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
# __PassStamp__
#
# $What: <@(#) Permissions.proto,v  4.1.1.4> $

# This entry for public login.
# It provides the default permissions.
# See Managing the DG/UX System for more information.

LOGNAME=nuucp   READ=var/spool/uucppublic/ \
WRITE=/var/spool/uucppublic COMMANDS=rmail:mail:lp \
SENDFILES=yes REQUEST=yes

LOGNAME=major MACHINE=chicago READ=/ WRITE=/ COMMANDS=ALL \
SENDFILES=yes REQUEST=yes

MACHINE=chicago READ=var/spool/uucppublic/ \
WRITE=/var/spool/uucppublic COMMANDS=rmail:mail:lp \
SENDFILES=yes REQUEST=yes
```

Setting UUCP passwords on the NEsales host — After identifying the "LOGNAME" used by the Chicago system when calling the NEsales system, you must set a password for it as shown below. You must also provide the password to the UUCP administrator on the Chicago system.

```
# passwd major ↵
# New passwd: daduucp ↵
# Re-enter new password: daduucp ↵
```

Specifying a system the Chicago host can access — This file contains an entry for the NEsales system. As shown below, the `Time` field in the entry has been edited. It now shows that the NESales system can be called Monday through Friday, from 0700 to 2330 (7:00 a.m. to 11:30 p.m., Chicago time; 8:00 a.m. to 12:30 a.m., Boston time)

```
  (On Chicago host)
#$What: <@(#) Systems.proto,v 4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may,
#         call(e.g., MoTuTh0800-1700).
#         Use "Any" for any day.
#         Use "Never" for machines that poll you,
#         but that you never call directly.
# Type            device type
# Class           transfer speed
# Phone           phone number (for autodialers) or token
#         (for data switches)
# Login           login sequence is composed of fields and
#         subfields in the format "[expect send]
#         ...". The expect field may have subfields
#         in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer

NEsales MoTuWeThFr0700-2330 2400 boston5551234 in:--in: NEuucp \
word NEuucp:
# sysadm NEsales modem_name=hayes switch_id= login_name=NEuucp
passwd=NEuucp
```

Specifying systems the NEsales host can access — This file contains an entry for the Chicago system. As shown below, the `Time` field in the entry has not been edited. It shows that the Chicago system can be called at any time.

```
   (On NEsales host)
#$What: <@(#) Systems.proto,v  4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may,
#        call(e.g., MoTuTh0800-1700).
#        Use "Any" for any day.
#        Use "Never" for machines that poll you,
#        but that you never call directly.
# Type            device type
# Class           transfer speed
# Phone           phone number (for autodialers) or token
#        (for data switches)
# Login           login sequence is composed of fields and
#        subfields in the format "[expect send]
#        ...". The expect field may have subfields
#        in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer
Chicago Any 2400 chicago3951156 in:--in: major word: daduucp
# sysadm chicago modem_name=hayes switch_id= login_name=major
passwd=daduucp
```

Peer-to-peer communications can now be initiated between the Chicago and NEsales systems used in this example. For more information on setting up transfers using UUCP commands and their servers, see Chapter 5 of and the man pages for UUCP-related commands and servers listed in Chapter 6 of this manual.

Example 3: Setting up systems in master/slave roles

Figure 4–4 shows a slave system (IllinoisMfg) added to the peer-to-peer communications scenario used in Example 3.

In this example, Chicago is a master of Illinois Mfg., which is the slave. In this UUCP relationship, Chicago is the active member of the communications link; that is, it initiates file transfers to and from the Illinois Mfg. system, and executes commands on the Illinois Mfg. system. Illinois Mfg. is the passive member; that is, it does not initiate UUCP communications of any kind with the Chicago system.

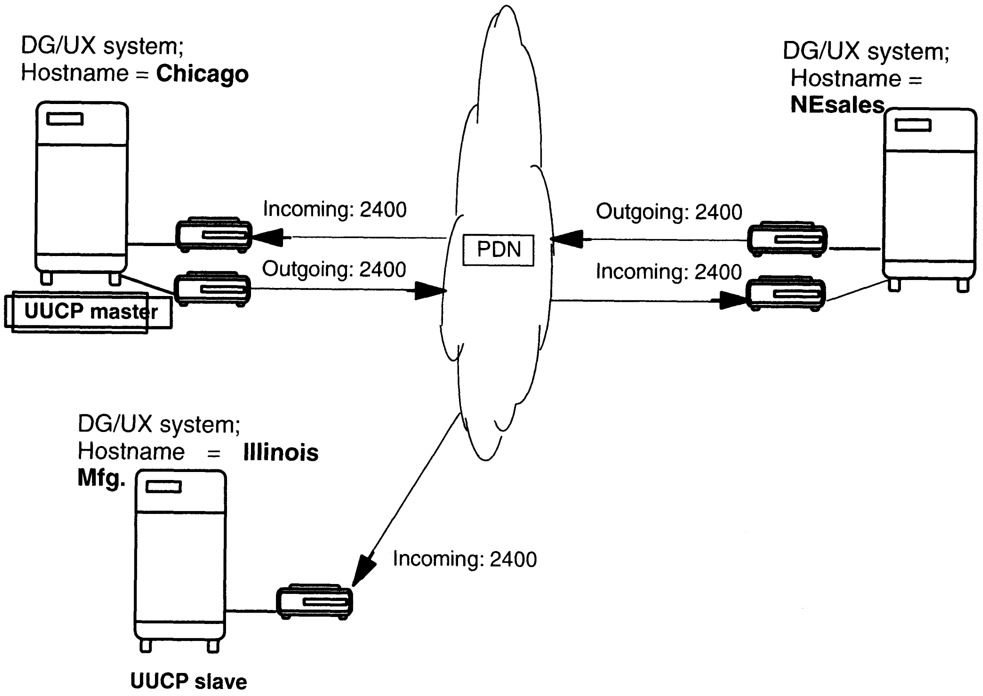


Figure 4–4 Master/slave UUCP communications between remote DG/UX systems

Editing the /etc/uucp files

To set up master/slave communications between two DG/UX systems called Chicago and Illinois Mfg., edit the following **/etc/uucp** files on both systems: **Dialcodes**, **Poll, Permissions,** and **Systems.**

The Dialcodes file on the Chicago host — As shown below, this file contains an entry for dialing the area code for Springfield, Illinois; namely, 1217, which is the location of the Illinois Mfg. system.

```
#  (On Chicago host)
# The Dialcodes file contains dial-code abbreviations
# that can be used in the 'Phone' field of the Systems
# file. Each entry has the format:
#
#        abbr     dial-seq
#
# where 'abbr' is the abbreviation used in the Systems
# file 'Phone' field, and 'dial-seq' is the dial sequence
# that is passed to the dialer when that particular Systems
# file entry is accessed.
#
# Example 1:      chicago     1312
# Example 2:      chicago 9=1312
boston 1617
springfield 1217
```

The **Dialcodes** file on the Illinois Mfg. system does not require editing, since this system does not initiate UUCP communications with other systems.

The Poll file on the Chicago host — As shown below, this file now contains an entry for the Illinois Mfg. system that indicates Chicago will initiate polling on the Illinois Mfg. system at 9:00 a.m., 12 noon, 3:00 p.m., and 6:00 p.m. In this case, both systems are located in the same time zone.

```
   (On Chicago host)
# $What: <@(#) Poll.proto,v  4.1.1.3> $
#This file (Poll) contains a list of
#"system <tab> hour1 hour2 hour3 ..." lines for polling remote
# systems. See examples below.
#
# Lines starting with # are ignored.
# NOTE a tab must follow the machine name

#raven 2  6  10
#quail 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
NEsales 10  16  22
Illinois Mfg.    9  12  15  18
```

The **Poll** file on the Illinois Mfg. system does not require editing, since this system does not initiate UUCP communications with other systems.

The Permissions file on the Chicago host — As shown below, the **Permissions** file on the Chicago system now contains three entries.

Like the second entry, the Illinois Mfg. entry gives the Chicago system complete access to the Illinois Mfg.'s file systems but also lets it execute any and all commands on the Illinois Mfg. system.

```
   (On Chicago host)
#
# Copyright (C) Data General Corporation, 1984 - 1990
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
# __PassStamp__
#
# $What: <@(#) Permissions.proto,v  4.1.1.4> $

# This entry for public login.
# It provides the default permissions.
# See Managing the DG/UX System for more information.

LOGNAME=nuucp

LOGNAME=NEuucp MACHINE=NEsales READ=var/spool/uucppublic/ \
WRITE=/var/spool/uucppublic COMMANDS=rmail:mail:lp \
SENDFILES=yes REQUEST=yes

MACHINE=NESALES LOGNAME=major READ=/ WRITE=/ COMMANDS=ALL \
SENDFILES=yes REQUEST=yes

MACHINE=Illinois Mfg. LOGNAME=major READ=/ WRITE=/ COMMANDS=ALL \
SENDFILES=yes REQUEST=yes
```

The Permissions file on the Illinois Mfg. host — As shown below, the **Permissions** file on the Illinois Mfg. system contains only one entry, which complies with the corresponding entry in the **Permissions** file on the Chicago system.

```
    (On Illinois Mfg. host)
#
# Copyright (C) Data General Corporation, 1984 - 1990
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
# __PassStamp__
#
# $What: <@(#) Permissions.proto,v  4.1.1.4> $

# This entry for public login.
# It provides the default permissions.
# See Managing the DG/UX System for more information.
LOGNAME=nuucp

LOGNAME=major MACHINE=chicago READ=/ WRITE=/ COMMANDS=ALL \
SENDFILES=yes REQUEST=yes
```

After identifying the "LOGNAME" used by the Chicago system when calling the Illinois Mfg. system, you must set a password for it on the Illinois Mfg. system, and provide the password to the UUCP administrator on the Chicago system.

The Systems file on the Chicago host — This file contains two entries: one for the NEsales system; the other for the Illinois Mfg. system.

```
   (On Chicago host)
#$What: <@(#) Systems.proto,v 4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may
#         call(e.g., MoTuTh0800-1700).
#         Use "Any" for any day.
#         Use "Never" for machines that poll you,
#         but that you never call directly.
# Type            device type
# Class           transfer speed
# Phone           phone number (for autodialers) or token
#         (for data switches)
# Login           login sequence is composed of fields and
#         subfields in the format "[expect send]
#         ...". The expect field may have subfields
#         in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer
NEsales MoTuWeThFr0700-2330 2400 boston5551234 in:--in: major \
word daduucp:
IllinoisMfg MoTuWeThFr0700-2330 2400 chicago5551234 in:--in: \
major word loluucp:
```

The Systems file on the Illinois Mfg. host — This file contains one entry for the Chicago system.

```
   (On Illinois Mfg. host)
#$What: <@(#) Systems.proto,v  4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may,
#        call(e.g., MoTuTh0800-1700).
#        Use "Any" for any day.
#        Use "Never" for machines that poll you,
#        but that you never call directly.
# Type            device type
# Class           transfer speed
# Phone           phone number (for autodialers) or token
#        (for data switches)
# Login           login sequence is composed of fields and
#        subfields in the format "[expect send]
#        ...". The expect field may have subfields
#        in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer
NEsales MoTuWeThFr0700-2330 2400 boston5551234 in:--in: major \
word daduucp:
# sysadm NEsales modem_name=hayes switch_id= login_name=NEuucp
passwd=NEuucp

Chicago Never 2400 chicago3951156 in:--in: major word: daduucp
# sysadm chicago modem_name=hayes switch_id= login_name=major
passwd=daduucp
```

The Systems file on the NEsales host — This file contains an entry for the Chicago system. As shown below, the `Time` field in the entry has not been edited.

```
    (On NEsales host)
#$What: <@(#) Systems.proto,v  4.1.1.5> $
# Entries have this format:
#
# Machine-Name Time Type Class Phone Login
#
# Machine-Name    node name of the remote machine
# Time            day-of-week and time-of-day when you may,
#        call(e.g., MoTuTh0800-1700).
#        Use "Any" for any day.
#        Use "Never" for machines that poll you,
#        but that you never call directly.
# Type            device type
# Class           transfer speed
# Phone           phone number (for autodialers) or token
#        (for data switches)
# Login           login sequence is composed of fields and
#        subfields in the format "[expect send]
#        ...". The expect field may have subfields
#        in the format "expect[-send-expect]".
#
# Example:
# cuuxb Any ACU 1200 chicago8101242 in:--in: nuucp word: panzer
Chicago Any 2400 chicago3951156 in:--in: major word: daduucp
# sysadm chicago modem_name=hayes switch_id= login_name=major
passwd=daduucp
```

Customizing UUCP data files

UUCP data files are owned by **uucp** and must grant the owner read and write access. This section describes the following files in **/etc/uucp** that support UUCP file transfers:

- **Devices**
- **Dialers**
- **Systems**
- **Dialcodes**
- **Permissions**
- **Poll**
- **Sysfiles**
- **Maxuuxqts**
- **Maxuuscheds**
- **remote.unknown**

/usR/lib/Send mail . CF
/usr/Lib/Mail/mailsurr

Devices file

The **Devices** file (**/etc/uucp/Devices**) contains information for all the devices that may be used to establish a link to a remote host; these are devices such as automatic call units, direct links, and network connections.

This file works closely with the **Dialers**, **Systems**, and **Dialcodes** files. Before you make changes in any of these files, you should be familiar with them all. A change to an entry in one file may require a change to a related entry in another file.

Each entry in the **Devices** file has the following format:

```
Type Line Line2 Class DTP
```

where:

Type `Direct` or `ACU` (described earlier in this chapter) or one of the following values:

Switch The name of a LAN switch listed in the **Dialers** file.

System The name of a host, indicating a direct link to that host. The line associated with this entry is for a particular host in the **Systems** file.

For every unique Type-Class pair in the third and fourth fields of the **Systems** file, you must have an entry in the **Devices** file where that pair occupies the first and fourth fields.

Line The device name of the line (modem port) associated with the **Devices** entry. For instance, if the modem for a particular entry is attached to the **/dev/tty11** line, the name entered in this field would be `tty11`.

Line2 This is normally a minus (-). It is the name of the ACU (for example, `801`), if the keyword ACU appears in the *Type* field and the ACU is separate from the modem.

Class If the keyword `ACU` or `Direct` is used in the *Type* field, *Class* may be just the speed of the device. However, it can contain a letter and a speed (for example, `C1200`) if needed to differentiate between classes of dialers (Centrex or Dimension PBX).

Some devices can be used at any speed, so the keyword `Any` may be used in the *Class* field. If `Any` is used, the line will match any speed requested in a **Systems** file entry. If this field is `Any` and the **Systems** file *Class* field is `Any`, the speed defaults to 1200 bps.

DTP Dialer-token pairs. This field has the format:

```
[dialer token[esc]]... dialer [token][esc]
```

where:

dialer The name of an automatic dial modem or a LAN switch, or the keyword direct for a Direct Link device. If you specify a name, it must be defined in the **Dialers** file or must be one of the following, which are compiled into the software: 801

token An argument to pass to the dialer. If you omit this field for the final pair, the token is retrieved from the *Phone* field of the **Systems** file entry.

esc One of the following escape sequences:

`\T` Indicates that the *Phone* field should be translated using the **Dialcodes** file. This escape character is normally placed in the **Dialers** file for each caller script associated with an automatic dial modem (**penril**, **ventel**, and so on). Therefore, the translation will not take place until the caller script is accessed.

`\D` Indicates that the *Phone* field should not be translated using the **Dialcodes** file. If no escape character is specified at the end of a **Devices** entry, the **\D** is assumed (default). A **\D** is also used in the **Dialers** file with entries associated with network switches (develcon and micom).

The *DTP* field can be structured four different ways, depending on the device associated with the entry:

- If an automatic dialing modem is connected directly to a port on your host, the *DTP* field of the associated **Devices** file entry will only have one pair. This pair would normally be the name of the modem. This name is used to match the particular **Devices** file entry with an entry in the **Dialers** file. Therefore, the *dialer* field must match the first field of a **Dialers** file entry as shown below:

  ```
  Devices: ACU tty11 - 1200 ventel

  Dialers: ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
  ```

  Notice that only the *dialer* portion (**ventel**) is present in the *DTP* field of the **Devices** file entry. This means that the token to be passed on to the dialer (in this case the phone number) is taken from the *Phone* field of a **Systems** file entry.

- If a direct link is established to a particular host, the *DTP* field of the associated entry would contain the keyword `direct`. This is true for both types of direct link entries, `Direct` and *System* (refer to discussion on the *Type* field).

- If a host with which you wish to communicate is on the same local network switch as your host, your host must first access the switch and the switch can make the connection to the other host. In this type of entry, there is only one pair. The *dialer* portion is used to match a **Dialers** file entry as shown below:

  ```
  Devices: develcon tty13 - 1200 develcon \D

  Dialers: develcon "" "" \pr\ps\c est:\007 \E\D\e \007
  ```

  As shown, the *token* portion is left blank, which indicates that it is retrieved from the **Systems** file. The **Systems** file entry for this particular host will contain the token in the *Phone* field, which is normally reserved for the phone number of the host (refer to **Systems** file, *Phone* field). This type of *DTP* contains an escape character (**\D**), which ensures that the contents of the *Phone* field will not be interpreted as a valid entry in the **Dialcodes** file.

- If an automatic dialing modem is connected to a switch, your host must first access the switch and the switch will make the connection to the automatic dialing modem. This type of entry requires two dialer-token pairs. The *dialer* portion of each pair (fifth and seventh fields of entry) will be used to match entries in the **Dialers** file as shown below:

  ```
  Devices: ACU tty14 - 1200 develcon vent ventel

  Dialers: develcon "" "" \pr\ps\c est:\007 \E\D\e \007
  Dialers: ventel =&-% "" \r\p\r\c $ <K\eT%%\r>\c ONLINE!
  ```

In the first pair, **develcon** is the dialer and **vent** is the token that is passed to the Develcon switch to tell it which device (**ventel** modem) to connect to your host. This token would be unique for each LAN switch since each switch may be set up differently. Once the **ventel** modem has been connected, the second pair is accessed, where **ventel** is the dialer and the token is retrieved from the **Systems** file.

Dialers file

The **Dialers** file (**/etc/uucp/Dialers**) specifies the initial conversation that must take place on a line before you can use it to transfer data. This conversation is usually a sequence of ASCII strings that are transmitted or expected (called a chat script). A chat script is often used to dial a phone number using an ASCII dialer (such as an automatic dial modem).

As shown earlier, the fifth field in a **Devices** file entry is an index into the **Dialers** file or a special dialer type (801). Here an attempt is made to match the fifth field in the **Devices** file with the first field of each **Dialers** file entry. In addition, each odd numbered **Devices** field (the token field) starting with the seventh position is used as an index into the **Dialers** file. If the match succeeds, the **Dialers** entry is interpreted to perform the dialer negotiations.

Each entry in the **Dialers** file has the following format:

```
dialer substitutions expect-send ...
```

The *dialer* field matches the fifth and additional odd numbered fields in the **Devices** file. The *substitutions* field is a translation string: the first of each pair of characters is mapped to the second character in the pair. This is usually used to translate the equals (=) and minus (–) characters into whatever codes the dialer requires for “wait for dial tone” and “pause.”

The remaining *expect-send* fields are character strings. Below are some character strings distributed with the **Dialers** file.

```
penril =W-P "" \d > K\c : \EP\T OK
penril_old =W-P "" \d > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
hayes_att =,-, "" \dAT\r\c OK\r ATDT\T\r\c CONNECT
rixon =&-% "" \d\r\r\c $ s9\c )-W\r\ds9\c-) s\c : \T\r\c $ 9\c LINE
vadic =K-K "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon "" "" \pr\ps\c est:\007 \E\D\e \007
micom ""  "" \s\c NAME? \D\r\c GO
direct
att2212c   =+-, "" \r\c :--: ato12=y,T\T\r\c red
att4000 =,-, "" \033\r\r\c DEM: \033s0401\c \006 \033s0901\c \
         \006 \033s1001\c \006 \033s1102\c \006 \033dT\T\r\c \006
att2224 =+-,   "" \r\c :--: T\T\r\c red
nls     ""     "" NLPS:000:001:1\N\c
```

Three AT&T modems have entries in the **Dialers** file. The Penril, Micom modem, and Hayes modem scripts have all been confirmed at Data General as have the Micom and Develcon data switches. The other entries have not been tested. If you need to modify the supplied script, refer to your modem documentation.

Table 4–3 shows the meanings of some of the escape characters (those beginning with ****) used in the **Dialers** file.

Table 4–3 Escape characters used in Dialers file

Escape Character	Description
\D	Phone number or token without **Dialcodes** translation.
\E	Enable echo checking (for slow devices).
\T	Phone number or token with **Dialcodes** translation.
\K	Insert a BREAK.
\c	No new line or carriage return.
\d	Delay (approximately 2 seconds).
\e	Disable echo checking.
\n	Send a new-line character.
\p	Pause (approximately 1/4 to 1/2 second).
\r	Carriage return.
nnn	Send octal number *nnn*.

Additional escape characters that may be used are listed in the section discussing the **Systems** file. The Penril entry in the **Dialers** file is executed as follows:

=W-P The phone number argument is translated, replacing any equal sign (**=**) with a **W** (wait for dial tone) and replacing any minus (**-**) with a **P** (pause).

"" Wait for nothing. (In other words, proceed to the next thing.)

\d Delay for 2 seconds.

> Wait for a >.

K\c Send a **K**. Send no terminating new line.

: Wait for a **:**.

\EP\T Enable echo checking. (From this point on, whenever a character is transmitted, it will wait for the character to be received before doing anything else.) Then, send a **P** and the phone number. The **\T** means take the phone number passed as an argument and apply the **Dialcodes** translation and the modem function translation specified by field 2 of this entry.

OK Waiting for the string **OK**.

Systems file

The **Systems** file (**/etc/uucp/Systems**) contains the information needed by the **uucico** server to establish a communication link to a remote host. Each entry in the file represents a host that can be called by your host. In addition, UUCP software can be configured to prevent any host that does not appear in this file from logging in on your host. More than one entry may be present for a particular host. The additional entries represent alternative communication paths that will be tried in sequence.

Using the **Sysfiles** file, you can define several files to be used as **Systems** files. See the description of the **Sysfiles** file later in this chapter for details.

Each entry in the **Systems** file has the following format:

```
Sys Time[;retry] Type[,proto] Class Phone Login
```

where:

Sys The host name of the remote system.

Time A string that indicates the days of the week and times of day when the remote host can be called. The time of day is a range of times within a 24-hour span. If you omit the time of day, any time of day is allowed. The day of week can contain the following:

`Su Mo Tu We Th Fr Sa` for individual days.

`Wk` for any weekday (Monday through Friday).

`Any` for any day.

`Never` Your host will never initiate a call to the remote host. The remote host must initiate the call.

The following example allows calls from 5:00 p.m. to 8:00 a.m. Monday through Friday and any time Saturday and Sunday:

```
Su,Wk1700-0800,Sa
```

retry The minimum time (in minutes) to wait before retrying following a failed attempt. The default is 60 minutes.

Type The device type that should be used to establish the communication link to the remote host.

proto The protocol used to contact the system.

Class The transfer speed of the device used in establishing the communication link. It may contain a letter and speed (for example, `C1200`) to differentiate between classes of dialers (refer to the discussion on the **Devices** file, *Class* field). Some devices can be used at any speed, so the keyword `Any` may be used.

If information is not required for this field, use a minus (–) as a place holder for the field.

Phone The telephone number of the remote host for automatic dialers or LAN switches. The number is made up of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be listed in the **Dialcodes** file.

If your host is connected to a LAN switch, you may access other hosts that are connected to that switch. The **Systems** file entries for these hosts will not have a phone number in the *Phone* field. Instead, this field will contain the token that must be passed on to the switch so it will know which host your host wishes to communicate with. (This is usually just the system name.) The associated **Devices** file entry should have a `\D` at the end of the entry to ensure that this field is not translated using the **Dialcodes** file.

Login Login information of the format:

expect send

where *expect* is the string that is received, and *send* is the string that is sent when the *expect* string is received.

The *expect* field may be made up of subfields of the form:

expect[-*send* -*expect*] . . .

where the *send* is sent if the prior *expect* is not successfully read and the *expect* following the *send* is the next expected string.

For example, with **login--login**, UUCP will expect **login**. If UUCP gets **login**, it will go on to the next field. If it does not get **login**, it will send a null string followed by a new line, then look for **login** again. If no characters are initially expected from the remote host, the characters **""** (null string) should be used in the first *expect* field. Note that all *send* fields will be sent followed by a new line unless the *send* string is terminated with a **\c**.

When assembling a send/expect sequence, it is good practice not to specify the first letter of the **login:** or **password:** strings that you want UUCP to expect. The reason for this is that systems in general are inconsistent as regards the case of these first letters—some systems prompt with **login:** while others with **Login:**. Even on a given system, **login:** may appear all lowercase while **Password:** appears with a capital **P**. To avoid problems with capitalization, it is best to specify shortened forms such as **ogin:** and **sword:.**

Here is an example of a **Systems** file entry that uses an *expect-send* string:

```
wl Any ACU 1200 5556013 "" \r ogin:-BREAK-ogin: uucpx word: xyz
```

This example says expect nothing, but send a carriage return and wait for **ogin:** (for **Login:**). If you don't get **ogin:**, send a **BREAK**. If you next receive **ogin:**, send the login name **uucpx**. When you receive **word:** (for **Password:**), send the password **xyz.**

There are several escape characters that cause specific actions when they are a part of a string sent during the login sequence (Table 4–4).

Table 4–4 Escape characters for UUCP communications

Escape Character	Description
\b	Send or expect a backspace character.
\c	If at the end of a string, suppress the new line that is normally sent. Ignored otherwise.

Continued

Table 4–4 Escape characters for UUCP communications

Escape Character	Description
`\d`	Delay two seconds before sending or reading more characters.
`\e`	Echo check off.
`\n`	Send a new-line character.
`\p`	Pause for approximately 1/4 to 1/2 second.
`\r`	Send or expect a carriage return.
`\s`	Send or expect a space character.
`\t`	Send or expect a tab character.
`\E`	Start echo checking. (From this point on, whenever a character is transmitted, it will wait for the character to be received before doing anything else.)
`\K`	Send or expect a break character.
`\N`	Send or expect a null character (ASCII NUL).
`\\`	Send or expect a reverse solidus character (\).
`\BREAK`	Send or expect a break character.
`\EOT`	Send or expect EOT new line twice.
`\`*ddd*	Collapse the octal digits (*ddd*) into a single character.

Dialcodes file

The **Dialcodes** file (**/etc/uucp/Dialcodes**) contains the dialcode abbreviations that can be used in the *Phone* field of the **Systems** file. Each entry has the format:

```
abb dial-seq
```

where *abb* is the abbreviation used in the **Systems** file *Phone* field and *dial-seq* is the dial sequence that is passed to the dialer when that particular **Systems** file entry is accessed.

The entry

```
jt 9=847-
```

would be set up to work with a *Phone* field in the **Systems** file such as `jt7867`. When the entry containing `jt7867` is encountered, the sequence 9=847–7867 would be sent to the dialer if the token in the dialer-token-pair is `\T`.

Permissions file

The **Permissions** file (**/etc/uucp/Permissions**) specifies the permissions that remote hosts have with respect to login, file access, and command execution. There are options that restrict the remote host's ability to request files and its ability to receive files queued by the local site. Another option is available that specifies the commands that a remote site can execute on the local host. Note that the **Permissions** prototype file sent with this software release is most restrictive.

Permissions file entries

Each entry is a logical line with physical lines terminated by a \ to indicate continuation. Entries are made up of options delimited by white space. Each option is a name/value pair in the following format:

```
name=value
```

Note that no white space is allowed within an option assignment.

Comment lines begin with a **#** and they occupy the entire line up to a new-line character. Blank lines are ignored (even within multiline entries).

There are two types of **Permissions** file entries:

LOGNAME Specifies the permissions that take effect when a remote host logs in on (calls) your host.

MACHINE Specifies permissions that take effect when your host logs in on (calls) a remote host.

LOGNAME entries will contain a LOGNAME option and MACHINE entries will contain a MACHINE option. See your **/etc/uucp/Permissions.proto** file for more information on entry format.

Considerations

The following items should be considered when using the **Permissions** file to restrict the level of access granted to remote hosts:

- Each login ID used by remote hosts to log in for UUCP communications must appear in one and only one LOGNAME entry.
- Any site that is called whose name does not appear in a MACHINE entry will have the following default permissions/restrictions: local send and receive requests will be executed, the remote host can send files to your host's **/var/spool/uucppublic** directory, and the commands sent by the remote host for execution on your host must be one of the default commands (usually **rmail**).

Options

This section describes each option, specifies how each is used, and lists the default values.

REQUEST

When a remote host calls your host and requests to receive a file, this request can be granted or denied. The REQUEST option specifies whether the remote host can request to set up file transfers from your host.

```
REQUEST=yes
```

specifies that the remote host can request to transfer files from your host.

```
REQUEST=no
```

specifies that the remote host cannot request to receive files from your host. This is the default value. It will be used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote calls you) entry or a MACHINE (you call remote) entry. For reasons of security, you should have a unique login name and password for any remote machine that calls your host. Otherwise you don't know the caller's true identity.

SENDFILES

When a remote host calls your host and completes its work, it may attempt to take work your host has queued for it. The SENDFILES option specifies whether your host can send the work queued for the remote host.

```
SENDFILES=yes
```

specifies that your host may send the work that is queued for the remote host as long as it logged in as one of the names in the LOGNAME option. This string is mandatory if your host is in a passive mode with respect to the remote host.

```
SENDFILES=call
```

specifies that files queued in your host will be sent only when your host calls the remote host. The call value is the default for the SENDFILES option. This option is only significant in LOGNAME entries since MACHINE entries apply when calls are made out to remote hosts. If the option is used with a MACHINE entry, it will be ignored.

READ and WRITE

These options specify the various parts of the file system that the **uucico** program can read from or write to. The READ and

WRITE options can be used with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the **uucppublic** directory as shown in the following strings:

```
READ=/var/spool/uucppublic
WRITE=/var/spool/uucppublic
```

The strings

```
READ=/  WRITE=/
```

specify permission to access any file that can be accessed by a local user with **other** permissions.

The READ option is for requesting files; the WRITE option for depositing files. One of the values must be the prefix of any full pathname of a file coming in or going out. To specify more than one file, use colons to separate pathnames. To grant permission to deposit files in **/usr/news** as well as the public directory, use the following values with WRITE:

```
WRITE=/var/spool/uucppublic:/usr/news
```

If the READ and WRITE options are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the **/usr/news** pathname was the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

You should be careful what directories you make accessible for reading and writing by remote systems. For example, you probably wouldn't want remote hosts to be able to write over your **/etc/passwd** file so **/etc** shouldn't be open to writes.

NOREAD and NOWRITE

The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults.

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

permits reading any file except those in the **/etc** directory (and its subdirectories—remember, these are prefixes) and writing only to the default **/var/spool/uucppublic** directory. NOWRITE works in the same manner as the NOREAD option. The NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

CALLBACK

The CALLBACK option used in LOGNAME entries specifies that no transaction will take place until the calling system is called back. There are two situations when you would use CALLBACK. From a security standpoint, if you call back a machine you can be sure it is the machine it says it is. If you are doing long data transmissions, you can choose the machine that will be billed for the longer call.

```
CALLBACK=yes
```

specifies that your host must call the remote host back before any file transfers will take place.

The default for the CALLBACK option is `no`.

The CALLBACK option is very rarely used. Note that if two sites have this option set for each other, a transmission conversation will never get started.

COMMANDS

The COMMANDS option can be hazardous to the security of your system. Use it with extreme care.

The **uux** program generates remote execution requests and queues them for transfer to the remote host. Files and a command are sent to the target host for remote execution. Use the COMMANDS option in MACHINE entries to specify the commands that a remote host can execute on your host. Note that COMMANDS is not used in a LOGNAME entry; COMMANDS in MACHINE entries defines command permissions whether you call the remote system or it calls you.

```
COMMANDS=rmail
```

indicates the default commands that a remote host can execute on your host. Usng a command string is used in a MACHINE entry overrides the default commands. For instance,

```
MACHINE=owl:raven:hawk:dove \
COMMANDS=rmail:mail:lp
```

overrides the COMMAND default so that the hosts owl, raven, hawk, and dove can now execute **rmail**, **mail**, and **lp** on your host.

In addition to the names as specified above, there can be full pathnames of commands. For example,

```
COMMANDS=rmail:/usr/local/mail:/usr/bin/lp
```

specifies that command **rmail** uses the default path. The default paths for your host are **/bin**, **/usr/bin**, and **/usr/local**. When the remote host specifies **mail** or **/usr/bin/mail** for the command to be executed, **/usr/local/mail** will be executed regardless of the default path. Likewise, **/usr/bin/lp** is the **lp** command that will be executed.

Including the ALL value in the list means that any command from the remote hosts specified in the entry will be executed.

CAUTION: *If you use the ALL value, you give the remote host full access to your host. BE CAREFUL. This allows far more access than even normal users have.*

```
COMMANDS=/usr/local/mail:ALL:/usr/bin/lp
```

illustrates two points: The ALL value can appear anywhere in the string, and the pathnames specified for **mail** and **lp** will be used (instead of the default) if the requested command does not contain the full pathnames for **mail** or **lp**.

Whenever you specify potentially dangerous commands such as **cat** and **uucp** with the COMMANDS option, use the VALIDATE option with COMMANDS. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution server (**uuxqt**).

VALIDATE

The VALIDATE option verifies the caller's identity by requiring that privileged hosts have a unique login name and password for UUCP transactions. Use VALIDATE with the COMMANDS option when you specify commands that are potentially dangerous to your host's security. However, the login name and password associated with this entry be protected. If an outsider gets that information, that particular VALIDATE option can no longer be considered secure. (VALIDATE is merely an added level of security on top of the COMMANDS option, though it is a more secure way to open command access than ALL.)

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specifies that a remote host claiming to be **eagle**, **owl**, or **hawk** must use the login name **uucpfriend**.

But what does this have to do with the COMMANDS option, which only appears in MACHINE entries? The entry links the MACHINE entry (and COMMANDS option) with a LOGNAME entry associated with a privileged login. This link is needed because the execution server is not running while the remote host is logged in. In fact, it is an asynchronous process with no knowledge of what host sent the execution request. Therefore, the real question is how does your host know where the execution files came from?

Each remote host has its own spool directory on your host. These spool directories have write permission given only to the UUCP programs. The execution files from the remote host are put in its spool directory after being transferred to your host.

When the **uuxqt** server runs, it can use the spool directory name to find the MACHINE entry in the **Permissions** file and get the COMMANDS list, or if the host name does not appear in the **Permissions** file, the default list will be used.

The following example shows the relationship between the MACHINE and LOGNAME entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/local/mail \
READ=/  WRITE=/

LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/  WRITE=/
```

The value in the COMMANDS option means that remote mail and **/usr/local/mail** can be executed by remote users.

In the first entry, you must make the assumption that when you want to call one of the hosts listed, you are really calling either **eagle**, **owl**, or **hawk**. Therefore, any files put into one of the **eagle**, **owl**, or **hawk** spool directories is put there by one of those hosts. If a remote host logs in and says that it is one of these three hosts, its execution files will also be put in the privileged spool directory. You therefore have to validate that the host has the privileged login **uucpz**.

You may want to specify different option values for the hosts your host calls that are not mentioned in specific MACHINE entries. This may occur when there are many hosts calling in, and the command set changes from time to time. The name `OTHER` for the host name is used for this entry as shown below:

```
MACHINE=OTHER \
COMMANDS=rmail:mail:/usr/local/Photo:/usr/local/xp
```

All other options available for the MACHINE entry may also be set for the hosts that are not mentioned in other MACHINE entries.

Combining MACHINE and LOGNAME entries

It is possible to combine MACHINE and LOGNAME entries into a single entry where the common options are the same. For example, the two entries

```
MACHINE=eagle:owl:hawk REQUEST=yes \
  READ=/  WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
  READ=/  WRITE=/
```

share the same REQUEST, READ, and WRITE options. These two entries can be merged as shown below:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes \
  READ=/  WRITE=/
```

Poll file

The **Poll** file (**/etc/uucp/Poll**) contains information for polling remote hosts. Each entry in the **Poll** file contains the name of a remote host to call, followed by a tab character (a space won't work), and the hours the host should be called. The format of entries in the **Poll** file are:

```
sys-name hour ...
```

For example the entry:

```
eagle  0 4 8 12 16 20
```

will provide polling of host **eagle** every four hours.

The **uudemon.poll** script does not actually perform the poll. It merely sets up a polling work file (always named **C.***file*), in the spool directory that will be seen by the scheduler, which is started by **uudemon.hour**.

Sysfiles file

The **/etc/uucp/Sysfiles** file lets you assign different files to be used by **uucp** and **cu** commands as **Systems**, **Devices**, and **Dialers** files. Here are some cases where this optional file may be useful.

- You may want different **Systems** files so requests for login services can be made to different addresses than UUCP services.
- You may want different **Dialers** files to use different handshaking for **cu** and **uucp**.
- You may want to have multiple **Systems**, **Dialers**, and **Devices** files. The **Systems** file in particular may become large, making it more convenient to split it into several smaller files.

The format of the **Sysfiles** file is:

```
service=w systems=x dialers=y devices=z
```

where:

w is replaced by **uucico**, **cu**, or both separated by a colon.

x is one or more files to be used as the **Systems** file, with each file name separated by a colon and read in the order presented.

y is one or more colon-separated files to be used as the **Dialers** file.

z is one or more colon-separated files to be used as the **Devices** file.

Each file is assumed to be relative to the **/usr/lib/uucp** directory, unless a full path is given (note that the configuration files that you can change, such as **Devices**, **Systems**, and so on, are actually located in **/etc/uucp**; links in **/usr/lib/uucp** point to them). To continue an entry onto the next line, precede the new-line character with a reverse solidus (\).

Here's an example of assigning a local **Systems** file in addition to the usual **Systems** file:

```
service=uucico:cu  systems=Systems:Local_Systems
```

If this is in **/etc/uucp/Sysfiles**, then both **uucico** and **cu** will first look in **/etc/uucp/Systems**.

When you assign different **Systems** files for **uucico** and **cu** services, your machine stores two different lists of systems. You can print the **uucico** list using the **uuname** command or the **cu** list using the **uuname –c** command.

Maxuuxqts

The **Maxuuxqts** file defines the maximum number of **uuxqt** programs that can run at once. You are limited only by the number of processes you want running on your CPU. The default is 2.

Maxuuscheds

The **Maxuuscheds** file defines the maximum number of **uusched** programs that can run at once. You are limited only by the number of processes you want running on your CPU. The default is 2.

remote.unknown

The **remote.unknown** file is a shell script that executes when a machine that is not in the **Systems** file attempts to start a conversation. It will log the conversation attempt into the file **/var/spool/uucp/.Admin/foreign** and fail to make a connection. If you change the permissions of this file so it cannot execute (**chmod 000 remote.unknown**), your system will accept any conversation requests.

UUCP spool files

The files described in this section are created in spool directories to lock devices, hold temporary data, or keep information about remote transfers or executions.

TM A temporary data file created by UUCP processes under the spool directory (**/var/spool/uucp/***host*) when a file is received from another host. *host* is name of the remote host that is sending the file. The names of the temporary files have the following format:

TM.*pid***.***ddd*

where *pid* is a process-ID and *ddd* is a three-digit sequence number starting at 000.

When the entire file is received, the **TM.***pid***.***ddd* file is moved to the pathname specified in the **C.***sysndddd* file (discussed later in this section) that caused the transmission. If processing is abnormally terminated, the **TM.***pid***.***ddd* file may remain in the *host* directory. These files will be automatically removed by **uucleanup**.

LCK A lock file created in the **/var/spool/locks** directory for each device in use. The lock file, which contains the process ID of the process that created the lock, prevents duplicate conversations and multiple attempts to use the same calling device. The names of lock files have the following format:

LCK..*str*

where *str* is a device name or a host name.

When a process ends normally, **uucico** deletes lock files. When a process or conversation ends abnormally (usually only when a system crashes), lock files may remain in the spool directory and prevent further UUCP transactions. After 90 minutes they are reused. To run UUCP before the 90 minutes have elapsed, delete the lock files.

C A work file created in a spool directory when work (file transfers or remote command executions) has been queued for a remote host.

The names of work files have the format:

C.*sysndddd*

where *sys* is the name of the remote host, *n* is the ASCII character representing the grade (priority) of the work; the **uucico** code sets this priority and you may change it with **uucp**(1) and **uux**(1). *dddd* is the four digit job sequence number assigned by **uucp**.

Work files contain the following information:

- Full pathname of the file to be sent or requested.
- Full pathname of the destination or user file name.
- User login name.
- List of options.
- Name of associated data file in the spool directory. If the **uucp –c** or **uuto –p** option was specified, a dummy name (**D.0**) is used.
- Mode bits of the source file.
- Remote user's login name to be notified upon completion of the transfer

D A data file created when it is specified in the command line to copy the source file to the spool directory. The names of data files have the following format:

D.*systmdddd*[*yyy*]

where *systm* is the first five characters in the name of the remote host, *dddd* is a four-digit job sequence number assigned by **uucp**, and *yyy* a sub-sequence number used when there are several **D.** files created for a work (**C.**) file.

X An execute file created in the spool directory prior to remote command executions. The names of execute files have the following format:

X.*sysndddd*

where *sys* is the name of the remote host, *n* is the character representing the grade (priority) of the work, and *dddd* is a four-digit sequence number assigned by **uucp**.

Execute files contain the following information:

- Requester's login name and host name

- Name of files required for execution
- Input to be used as the standard input to the command string
- System and file name to receive the command's standard output
- Command string
- Option lines for return status requests

Log files

Log files are created for each remote host with which your host communicates. There are directories for each of the **uucico**, **uucp**, **uux** and **uuxqt** commands with subdirectories under these for each machine making requests. The log files are kept in the directory **/var/spool/uucp/.Log**.

The information from the individual log files for each machine and each program (e.g., machine **dumbo** has a log file for **uucico** requests and a log file for **uuxqt** execution requests) can be accessed with the **uulog** program. These files are combined and stored in directory **/var/spool/uucp/.Old** whenever **uudemon.cleanup** is executed. This shell script saves files that are three days old. The three days can be easily modified in the **uudemon.cleanup** shell. If space is a problem, you might consider reducing the number of days the files are kept.

UUCP file cleanup

Invoke the **uustat** program regularly to display the status of connections to various machines and the size and age of the queued requests. Use **cron** to start the **uudemon.admin** shell at least once a day to send you the current status. Of particular interest are the age (in days) of the oldest request in each queue, the number of times a failure has occurred when attempting to reach that machine, the reason for the failure, and the age of the oldest execution request (**X.***file*).

Execution files older than a few days can probably be deleted since the only reason they have not been executed is because data files required for execution were not sent. These files are usually sent at the same time as the **X.***file*, so the problem is likely at the other end.

The **uucleanup** program, which is run from **uudemon.cleanup**, removes these files. Options to **uucleanup** specify the age for sending a warning message to the requester and age for deleting various files. Before deleting, the program tries to figure out what the job was and, if possible, tries to send it to the receiver. If this is not possible, it is returned to the sender.

Public Area Cleanup

To keep the local file system from overflowing when files are sent to the public area, the **uudemon.cleanup** procedure is set up with a **find** command to remove any files that are older than seven days and directories that are empty. The interval may need to be shortened if there is not sufficient space to devote to the public area.

End of Chapter

5 Setting up UUCP to run over TCP/IP

This chapter explains how to use the Service Access Facility (SAF) and the Transport Layer Interface (TLI) to run UUCP on TCP/IP.

Setting up the server system

To set up the server system to receive UUCP connections via TCP/IP, add the following line to the **/etc/saf/tcp/_pmtab** file:

```
uucp:u:uusocrat:reserved:reserved:reserved:\x0002021C000000000000
000000000000::c:tirdwr:/usr/lib/uucp/uucico#
```

`21C` is a hexadecimal number representing port 540. You can use any port number in the range decimal 0 to 599 that is not already being used.

Setting up the client system

To set up the client system to run UUCP over TCP/IP, you need to add a TLIS (STREAMS-based TLI) device and add one or more TCP/IP-connected systems.

Adding a TLIS device

To add a TLIS device, add the following line to the **Devices** file:

```
TLIS,eg tcp - - TLIS \D nls
```

`TLIS` represents STREAMS-based `TLI`. `eg` represents the Network Listener Service.

Adding a TCP/IP-connected system

If you want UUCP to be able to connect to system **anchor** via TCP/IP on port 540, add a line such as the following to the **Systems** file:

```
anchor Any TLIS - \x0002021C80DE0A240000000000000000
```

`80DE0A24` is the hexadecimal IP address of **anchor**.

End of Chapter

6 Troubleshooting

This chapter describes how to troubleshoot a modem and how to solve common UUCP problems.

Troubleshooting an asynchronous modem

Following are troubleshooting hints for asynchronous modems:

Problem **Suggestion**

Modem does not seem to work at all.

- Bad modem. Swap with good one.
- Wrong cable. See Appendix C.
- Modem not echoing. See “Modem not echoing.”

Modem not echoing.

Set modem rear switch 6 to the down position and power the modem on, then off. Then reset rear switch 6 to the up position. Now set your modem registers as specified in Chapter 3.

Modem receives calls all right but will not dial out.

- Make sure the hayes entry in **/etc/uucp/Dialers** and **/etc/uucp/Devices** agrees with what this document says.
- Make sure the modem is echoing properly.
- Check the port service setup.
- If the modem has been working properly, delete and re-enter the port service for the tty line to which the modem is connected.

You can dial out but not receive calls.

- Check that the bidirectional bit is set to YES in the port service setup.
- Check the **/etc/ttydefs** file for the modifications specified in Chapter 3.
- Check modem register settings. See the hardware manual for your modem.

Characters being received are not correct.

- Check that both modems have been set to the same baud rate.
- Check the **/etc/ttydefs** file and make sure that the appropriate label is as specified in Chapter 3.

When a terminal is connected directly to the modem, it does not work.

- Check terminal setup.
- Check that you have the correct cable. See Appendix C.
- Reset modem to default settings. See “Modem not echoing.”

When the modem is powered off, it reverts to its default settings.

Make sure that Switch 6, on the modem’s read switch pack, is in the up position.

When trying to access modem, you get a “WARNING: can’t access device” error.

Check the permissions and owners of the **/dev/tty***XX* to which the modem is connected.

DTR light will not come on.

Port service has been disabled.

Not able to automatically log in to another system properly.

Check your setup in the **/etc/uucp/Systems** file for the system to which you are trying to connect.

Troubleshooting a synchronous modem

Following are troubleshooting hints for synchronous modems:

Modem does not work.

- Check for proper cable.
- Try connecting another modem.

Modem does not dial out.

- Check that required software such as SNA or X.25 is running.
- Check modem switch settings.

Solving common UUCP problems

Table 6–1 contains remedies for some of the common problems that may prevent UUCP from operating correctly.

Table 6–1 Troubleshooting hints for UUCP problems

Problem	Suggestion
Remote system down	Call the remote system's number and listen for the high-pitched tone of the answering modem. If there is none, you know the system or its modem is not operating. Call the system administrator of the remote system.
Incorrect login information for remote system	Dialing sequence: look in the **Systems** and **Dialcodes** files. Consider inserting pauses (commas) in the dialing sequence. Login name/password: make sure the login name/password in the local **Systems** file match the login name/password in the remote system's **passwd** file. Login sequence: examine the expect/send sequence in the **Systems** file and make sure it uses the correct conventions and login/password strings. Remote system name mismatch: make sure the system name in the **Systems** entry matches the nodename of the remote system. On the remote system, use the **uname –n** command to get the nodename.
Modem cannot make connection	Verify that your modem switches are properly set (refer to modem documentation). Read the **Dialers** file for guidance. Make sure your **Dialers** file is set correctly for touch tone or pulse.
Cannot dial in	Verify that your modem switches are correct. Put a phone on the line and call the remote system's modem. Make sure that the desired **ttymon** service is running on the port to which the modem is attached. Set up port services with the operation Device –> Port –> Port Service –> Add. If you set the port for bidirectional use, try typing anything or press New Line.
uucico or **cu** dies immediately	Use appropriate debugging switch first (for **uucico**(1M), **–x**; for **cu**(1), **–d**). If you get no response or a hangup, make sure **Systems**, **Devices**, **Permissions**, and **Dialers** files are present and readable by **uucp**. Make sure **passwd** and **group** files are correct on both systems.
Hung modem	Send the reset command (such as ATZ for Hayes modems) to the modem. You can do this with the **cu**(1) command or by turning the modem on and off.

Continued

Table 6–1 Troubleshooting hints for UUCP problems

Problem	Suggestion
Hung syac	Reload the **syac** with **/usr/sbin/tcload**. See the **tcload**(1M) manual page.
Modem in wrong state (such as echo mode)	Examine the modem switches. In the case of Hayes-compatible modems, sending ATZ to the modem may fix the problem. See the documentation for your modem. You can find some modem settings in the **Dialers** file.
Getty on line	Traditionally, you had to put a **getty** on a dial-in line but not on a dial-out line. For a bidirectional port, **uugetty** had to be used. DG/UX Release 5.4 and later systems use the SAF facility instead of **getty** and **uugetty**. On a DG/UX Release 5.4 or later system, you set up login port service, non-bidirectional, for a dial-in-only line; you set up no port service for a dial-out-only line; and you set up login port service, bidirectional, for a line that you can use both ways. See the Port menu of **sysadm**'s Device menu. *Managing the DG/UX™ System* covers the Port menu.
Wrong line speed	Make sure the modem line speed (baud rate) is compatible with the entry in the **/etc/inittab** file.
Wrong permissions or access	The device file (such as **/dev/tty***xx*) should permit reading and writing for any user. Permissions should be set at 666. All files in **/etc/uucp**, **/usr/lib/uucp**, and **/var/spool/uucp** should be owned by **uucp**.
Too many unsuccessful attempts	Remove the status files (files with names matching **/var/spool/uucp/.Status/***system_name*, where *system_name* is the name of the remote system in the UUCP request). Also remove any related lock files (**/var/spool/locks/LCK.***) and start **uucico** yourself to try and complete the job. You may also have to delete any temporary files (**/var/spool/uucp/***system_name***/TM***, where *system_name* is the name of the remote system in the UUCP request) that **uucico** might have created during a file transfer.
No server program	Make sure the **uucp** or **uux** commands start the **uucico** or **uuxqt** servers without problems.
Permissions file	Make sure this file does not prohibit the desired transfer. Make sure it contains an entry for the **uucp** login name on the remote system.
Bad modem or telephone line	Test the modem and cable and make sure they are functioning properly. Test the phone line for noise or interruptions.

Continued

Table 6–1 Troubleshooting hints for UUCP problems

Problem	Suggestion
Out of file system space	There is not enough space on the remote system to transfer the file. Contact the administrator of the remote system.
Bad pathname	Incorrect or illegal pathname. Enter the correct pathname.

UUCP error and status messages

This section contains error and status messages for the UUCP system. This appendix lists two types of error and status messages associated with UUCP connections:

- ASSERT errors are recorded in the **/var/spool/uucp/.Admin/errors** file.
- STATUS errors are recorded in individual machine files found in the **/var/spool/uucp/.Status** directory.

ASSERT error messages

When a UUCP process fails, ASSERT error messages may be generated and recorded in **/var/spool/uucp/.Admin/errors**. These messages include the file name, SCCS id, line number, and the text listed below. ASSERT messages reflect conditions which the system manager must correct. These errors are usually due to system problems.

Table 6–2 ASSERT error messages

Assert error message	Description/Action
BAD LINE	There is a bad line in the **Devices** file; there are not enough arguments on one or more lines.
BAD SPEED	A bad line speed appears in the **Devices/Systems** files (Class field).
BAD LOGIN_UID	The uid cannot be found in the **/etc/passwd** file. The file system is in trouble, or the **/etc/passwd** file is inconsistent.
BAD UID	Same as previous.
CAN'T ALLOCATE	A dynamic allocation failed.
CAN'T CHDIR	A **chdir**() call failed.

Continued

Table 6–2 ASSERT error messages

Assert error message	Description/Action
CAN'T CHMOD	A **chmod**() call failed.
CAN'T CREATE	A **create**() call failed.
CAN'T CLOSE	A **close**() or **fclose**() call failed.
CAN'T FORK	An attempt to **fork** and **exec** failed. The current job should not be lost, but will be attempted later (**uuxqt**). No action need be taken.
CAN'T LINK	A **link**() call failed.
CAN'T LOCK	An attempt to make a **LCK** file (in **/var/spool/locks**) failed.
CAN'T OPEN	An **open**() or **fopen**() failed.
CAN'T READ	A **read**(), **fgets**(), etc., failed.
CAN'T STAT	A **stat**() call failed.
CAN'T UNLINK	An **unlink**() call failed.
CAN'T WRITE	A **write**(), **fwrite**(), **fprint**(), etc., failed.
FILE EXISTS	The creation of a work file (**C.***file*) or data file (**D.***file*) was attempted, but the file exists. This occurs when there is a problem with the sequence file access. This problem usually indicates a software error. See the release notice for instructions on filling out a Software Trouble Report (STR) to send to the Data General Customer Support Center.
PERMISSIONS file: BAD OPTION	There is a bad line or option in the **Permissions** file.
PKCGET READ	The remote machine probably hung up. No action need be taken.
PKXSTART	The remote machine aborted in a nonrecoverable way. This can generally be ignored.
RETURN FROM fixline ioctl	An **ioctl**() call, which should never fail, failed. There is a system driver problem. See the release notice for instructions on filling out a Software Trouble Report (STR) to send to the Data General Customer Support Center.
SYSLST OVERFLOW	An internal table in **gename.c** overflowed. A big/strange request was attempted.
SYSTAT OPEN FAIL	There is a problem with the modes of **/var/spool/uucp/.Status**, or there is a file with bad modes in the directory.

Continued

Table 6–2 ASSERT error messages

Assert error message	Description/Action
TOO MANY FILES	Same as *SYSLST OVERFLOW.*
TOO MANY LOCKS	There is an internal problem. See the release notice for instructions on filling out a Software Trouble Report (STR) to send to the Data General Customer Support Center.
ULIMIT TOO SMALL	The ulimit for the current user process is too small. File transfers may fail, so transfer is not attempted.
WRONG ROLE	This is an internal logic problem. See the release notice for instructions on filling out a Software Trouble Report (STR) to send to the Data General Customer Support Center.
XMV ERROR	There is a problem with some file or directory. It is likely the spool directory, since the modes of the destinations were supposed to be checked before this process was attempted.

STATUS messages

Status messages are stored in the **/var/spool/uucp/.Status** directory. This directory contains a separate file for each remote machine that your system attempts to communicate with. These files contain status information on the attempted communication, whether it was successful or not. What follows is a list of the most common messages that may appear in these files.

Table 6–3 STATUS messages

Status message	Description/Action
ASSERT ERROR	An ASSERT error occurred. Check the **/var/spool/uucp/.Admin/errors** file for the error message.
BAD LOGIN/MACHINE COMBINATION	The machine called us with a login name or machine name that does not agree with the **Permissions** file.
CALLBACK REQUIRED	The called machine requires that it call your DG/UX system.

Continued

Table 6–3 STATUS messages

Status message	Description/Action
CALLER SCRIPT FAILED	This is usually the same as `DIAL FAILED`; however, if it occurs often, suspect the caller script in the **Dialers** file. Use **Uutry** to check.
CAN'T ACCESS DEVICE	The device tried does not exist or the modes are wrong. Check the appropriate entries in the **Systems** and **Devices** files.
CONVERSATION FAILED	The conversation failed after successful startup. This usually means that one side went down, the program aborted, or the line (link) was dropped.
DEVICE FAILED	The open of the device failed.
DEVICE LOCKED	The calling device to be used is currently locked and in use by another process.
DIAL FAILED	The remote machine never answered. It could be a bad dialer or the wrong phone number.
LOGIN FAILED	The login for the given machine failed. It could be a wrong login/password, wrong number, a very slow machine, or failure in getting through the Dialer-Token-Pairs script.
NO DEVICES AVAILABLE	There is currently no device available for the call. Check to see that there is a valid device in the **Devices** file for the particular system. Check the **Systems** file for the device to be used to call the system.
OK	Things are OK.
REMOTE DOES NOT KNOW ME	The remote machine does not have the nodename of your computer in its **Systems** file.
REMOTE HAS A LCK FILE FOR ME	The remote site has a **LCK** file for your computer. They could be trying to call your machine. If they have an older version of UUCP, the process that was talking to your machine may have failed leaving the **LCK** file. Check to see if the remote process that created the **LCK** file is hung.
REMOTE REJECT AFTER LOGIN	The login used by your computer to login does not agree with what the remote machine was expecting.
REMOTE REJECT, UNKNOWN MESSAGE	The remote machine rejected the communication with your computer for an unknown reason. The remote machine may not be running a standard version of UUCP.
STARTUP FAILED	Login succeeded, but initial handshake failed.

Continued

Table 6–3 STATUS messages

Status message	Description/Action
SYSTEM NOT IN Systems	The system is not in the **Systems** file.
TALKING	Local UUCP and remote UUCP are communicating successfully.
WRONG TIME TO CALL	A call was placed to the system at a time other than what is specified in the **Systems** file.
WRONG MACHINE NAME	The called machine is reporting a different name than expected.

End of Chapter

7 Accessing on-line information

This chapter discusses DG/UX tools that can help you access UUCP information on line. Two types of information are available on line: reference manual pages (available only on line) and the manual you are now reading (available in printed form or accessible on line if you have installed the Interleaf® WorldView program.

Displaying reference information

▶ To display a particular entry, use the **man** command. For example:

```
% man uucp ↵
```

For manual pages that document more than one command, you can access the man page either by the man page name or the command name. You have two ways to display the **uuencode**(1) manual page:

```
% man uuencode ↵
% man uudecode ↵
```

Listing all UUCP reference entries

▶ To display a list of all UUCP manual pages with a one-line summary for each, use the following **apropos** command:

```
% apropos 'cu( uu' ↵
```

Getting a synopsis of a reference entry

▶ To display the list of commands documented in a particular manual page, plus a one-line summary of the man page, use the **whatis** command. For example:

```
% whatis uucp ↵
```

For more information, see the **man** and **whatis** man pages.

Browsing this manual electronically

▶ To browse an electronic copy of this manual or any other manual in the DG/UX documentation set, you can use the Interleaf® WorldView program. Follow these steps:

1. Load WorldView from CD ROM.
2. Issue the following command from a shell on an AViiON workstation: **`iview`**
3. When the WorldView window appears, move your cursor to the icon directly beneath the menu item "Bookmarks" and to the left of the "ABC" icon. The icon you want looks like a square containing several little pages. Click your left button on this icon.
4. The "Titles" window appears; the collection specified is "DG/UX System Documentation Collection." Scroll through the list until you find a document that you wish to view; either double-click on that item or select the item (by clicking on it once) and then click on "Open."
5. To exit from WorldView, choose the "File → Exit" option.

End of Chapter

A How the DG/UX system handles modem signals and connections

This appendix describes the response of the DG/UX system to common modem signals. It also describes how the operating system establishes a connection with a modem, as well as when and how it performs a disconnection.

When a modem receives a call

The following events occur when someone calls your modem, assuming that your modem is correctly configured and you are running **ttymon** on the port the modem is connected to.

1. The modem notices the telephone call, and answers the telephone.
2. The two modems talk and negotiate connect speed, compression type, and error correction type.
3. In response to text from the modem, or the Data Carrier Detect (DCD) signal, **ttymon** sends the "login:" prompt to the modem and login proceeds.

When a call ends

The call normally ends in either of two ways. The caller either hangs up or logs out.

Hanging up

To hang up, you can either press the BREAK key three times or unplug the telephone line connected to the modem.

If you hang up, **ttymon** terminates your **login** process. **ttymon** then briefly lowers the Data Terminal Ready (DTR) line to the modem. This resets the modem and prepares it for the next call. (As an alternative to hanging up, you can press the T/D switch with the phone on hook.)

Logging out

If you log out, **ttymon** briefly lowers the DTR line to the modem. This resets the modem and prepares it for the next call.

Expected modem behavior

Ttymon requires that the modem respond to the loss of the DTR signal by disconnecting and entering command state. When the modem disconnects, it must also drop Data Carrier Detect (DCD) and Data Set Ready (DSR) signals.

When you log off, DTR will drop and will not be reasserted until after the modem drops DCD and DSR.

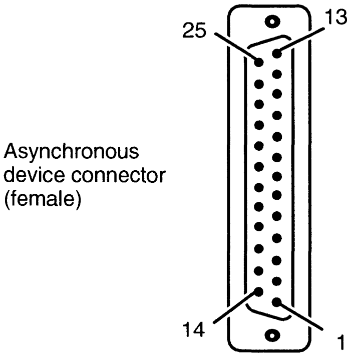
End of Appendix

B Interface connector pin assignments

This appendix describes the interface connector pin assignments for AViiON computers.

VAC/16, VDC/16, and VDC/8P asynchronous serial port connectors

Serial devices connect to the serial ports through RS-232C 25-pin female DB25 connectors. Asynchronous connectors are located either on a VDA/255 (or VDA/128) cluster box or on a VAC/16 junction box. Figure B–1 shows the signals and pin numbers for these asynchronous connectors.



Pin	Signal
1	Chassis Ground (CG)
2	Transmit Data (TxD) <
3	Receive Data (RD) >
4	Request to Send (RTS) <
5	Clear to Send (CTS) >
6	Data Set Ready (DSR) >
7	Signal Ground (SG)
8	Data Carrier Detect (DCD) >
20	Data Terminal Ready (DTR) <

< indicates received by controller
> indicates transmitted from controller

Figure B–1 Asynchronous serial connector signals

VAC/16, VDA/128, and VDA/255 boards are factory-configured as Data Communications Equipment (DCE). The female device connectors on the computer unit bulkhead and your junction and cluster boxes are also DCE.

VAC/16 controllers connect to their 8-port junction boxes through a data-transmission cable with either 64- or 68-pin connectors at either end. The signals and pin assignments for these connectors are identical except for the last four pins of the 68-pin connector, which are not used. Refer to the *HPS VMEbus Multiplexer (HPS-6236 / 6237) Technical Manual* for detailed information.

VSC/3i synchronous serial port connectors

Devices connect to the VSC/3i ports (or *channels*) through 25-pin female DB25 connectors marked "CH A," "CH B," and "CH C." In addition to RS-232C, the VSC/3i supports RS-530, RS-449, X.21, and V.35 standards via board jumpering and special cables. The signals supported by each connector on your VSC/3i depend on the interface type you select when configuring your VSC/3i. Figure B–2 shows a VSC/3i rear panel connector, and Table B–1 lists the signals and pin numbers for each electrical interface supported by VSC/3i connectors.

IMPORTANT: This section describes the signals leaving the VSC/3i DB25 connector. On a VSC/3i, special cables are available which map these signals to other pins to support a given standard (for example, RS-449 uses a 37-pin arrangement).

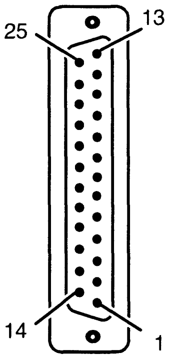


Figure B–2 VSC/3i rear panel connector (female)

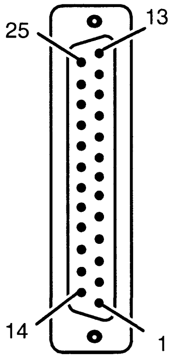
Table B–1 VSC/3i connector signals

Pin #	Electrical Interface		
	RS-232	**RS-422**	**V.35**
1	Chassis Ground	Shield A	Shield
2	Transmit Data (TxD) >	Transmit Data (TxD) A >	Send Data (SD) A >
3	Receive Data (RD) <	Receive Data (RD) A <	Receive Data (RD) A <
4	Request to Send (RTS) >	Request to Send (RTS) A >	Request to Send (RTS) >
5	Clear to Send (CTS) <	Clear to Send (CTS) A <	Clear to Send (CTS) <
6	Data Set Ready (DSR) <	Data Set Ready (DSR) A <	Data Set Ready (DSR) <
7	Signal Ground (SG)	Signal Ground (SG) A	Signal Ground (SG)
8	Data Carrier Detect (DCD) <	Data Carrier Detect (DCD) A <	Data Carrier Detect (DCD) <
9		Receive Clock (RxC) B <	Serial Clock Receive (SCR) B <
10		Data Carrier Detect (DCD) B <	
11		Transmit Clock (TxC) B >	Serial Clock Transmit (SCTE) B >
12		Transmit Clock (TxC) B <	Serial Clock Transmit (SCT) <
13		Clear to Send (CTS) B <	
14		Transmit Data B >	Send Data (SD) B >
15	Transmit Clock (TxC) <	Transmit Clock (TxC) A <	Serial Clock Transmit (SCT) A <
16		Receive Data (RxD) B <	Receive Data (RxD) B <
17	Receive Clock (RxC) <	Receive Clock (RxC) A <	Serial Clock Receive (SCR) A <
18			
19		Request to Send (RTS) B >	
20	Data Terminal Ready (DTR) >	Data Terminal Ready (DTR) A >	Data Terminal Ready (DTR) >
21			
22		Data Set Ready (DSR) B <	
23		Data Terminal Ready (DTR) B >	
24	Transmit Clock (TxC) >	Transmit Clock (TxC) A >	Serial Clock Transmit (SCTE) A >
25			

NOTE: < indicates signal received by controller.
> indicates signal transmitted by controller.
A and B indicate polarity.

VSC/3 and VSC/4 synchronous serial port connectors

Synchronous serial devices connect to the VSC/3 and VSC/4 ports through RS-232C 25-pin female DB25 connectors. Figure B–3 shows the signals and pin numbers for VSC/3 and VSC/4 synchronous connectors.



VSC/3
connector
(female)

Pin	Signal and Direction	
1	Chassis Ground (CG)	
2	Transmit Data (TxD)	>
3	Receive Data (RD)	<
4	Request to Send (RTS)	>
5	Clear to Send (CTS)	<
6	Data Set Ready (DSR)	<
7	Signal Ground (SG)	
8	Data Carrier Detect (DCD)	<
15	Transmit Signal Timing (TxC)	<
17	Receive Signal Timing (RxC)	<
20	Data Terminal Ready (DTR)	>
24	Transmit Signal Timing (TxC)	>

< indicates received by controller
> indicates transmitted from controller

Figure B–3 VSC/3 and VSC/4 connector signals

End of Appendix

C Modem cable matrix

Table C–1 provides the DGC 005 part number for the cable required to connect your Data General Microcom modem to a specific AViiON communication interface.

Table C–1 Data General 005 part numbers for asynchronous modem cables

Interface	2-meter	6-foot	10-foot	15-foot	25-foot
AV 100/200/300 series			35445		33701
AV 310CD/500 series			35445		33701
AV 400/4000 series			35445		33701
AV 530/4300/4600 series			36256	36257	36258
AV 5200/6200/8000 series	34187	32917		32918	32919
AV 5200+/7000+ series	34187	32917		32918	32919
AV 625 series	34187	32917		32918	32919
AV 7400/8400 series			36256	36257	36258
AV 8500/9500 series			36256	36257	36258
VAC16 async. controller			36256	36257	36258
VDA/128/255 cluster			36256	36257	36258
TermServer Box					33764
VSC synch. controller	34187	32917		32918	32919

Table C–2 shows the model numbers and corresponding part numbers for asynchronous modem cables.

Table C–2 Asynchronous modem cable model numbers and 005 part numbers

Model number	Part number
1084M	33701
1084M-A	35445
15290E002, E006, E015, E025	34187, 32917, 32918, 32919
15369E010, E015, E025	36256, 36257, 36258

Table C–3 provides the DGC 005 part number for the cable required to connect a synchronouts modem to a specific AViiON VME controller.

Table C–3 Data General 005 part numbers for synchronous modem cables

AV series interface	2-m. RS 232	6-ft. RS 232	15-ft. RS 232	25-ft. RS 232	15-ft. RS 449	15-ft. RS 530	15-ft. V.35	15-ft. X.21
530/4300	34187	32917	32918	32919				
4605/4625 5200+/7000+ 6200/8000 7400/8400 8500/9500	34187	32917	32918	32919	39805	39804	39806	39800

Table C–4 shows the model numbers and corresponding part numbers for synchronous modem cables.

Table C–4 Synchronous modem cable model numbers and 005 part numbers

Model number	Part number
15290E002, E006, E015, E025	34187, 32917, 32918, 32919
15408E015	39805
15408E015	39804
15408E015	39806
15408E015	39800

End of Appendix

D Setting up a modem to use SLIP

This appendix tells you how to set up a modem to use the Serial Line Internet Protocol (SLIP) on DG/UX systems. SLIP enables you to run TCP/IP applications such as **telnet** and **ftp** over point-to-point serial lines. Systems running SLIP generally use modems to communicate across telephone lines.

This appendix discusses the following topics:

- Getting an overview of SLIP
- Configuring a modem to use SLIP

Getting an overview of SLIP

SLIP provides a point-to-point serial link over which you can run TCP/IP applications. The link is usually established through modems, though two hosts can be connected directly with serial cables. Because SLIP works over asynchronous lines, applications run significantly slower than over Ethernet or other local area network (LAN) media.

Data General's SLIP is based on *RFC 1055 (A Nonstandard for Transmission of IP Datagrams over Serial Lines)*. Header compression as defined in *RFC 1144 (Compressing TCP/IP Headers for Low-Speed Serial Links)* is also supported. If you use header compression, SLIP performance will improve.

SLIP can often be more convenient than other networking alternatives. Serial lines tend to be relatively inexpensive and easier to install than other sorts of media, and you can use existing telephone lines to establish SLIP links. Where your LAN cabling is restricted due to terrain, distance, and other factors, you can make network connections with serial lines.

Figure D–1 shows a SLIP connection between **sys23**, a remote host, and **sys05**, a host connected to an Ethernet LAN.

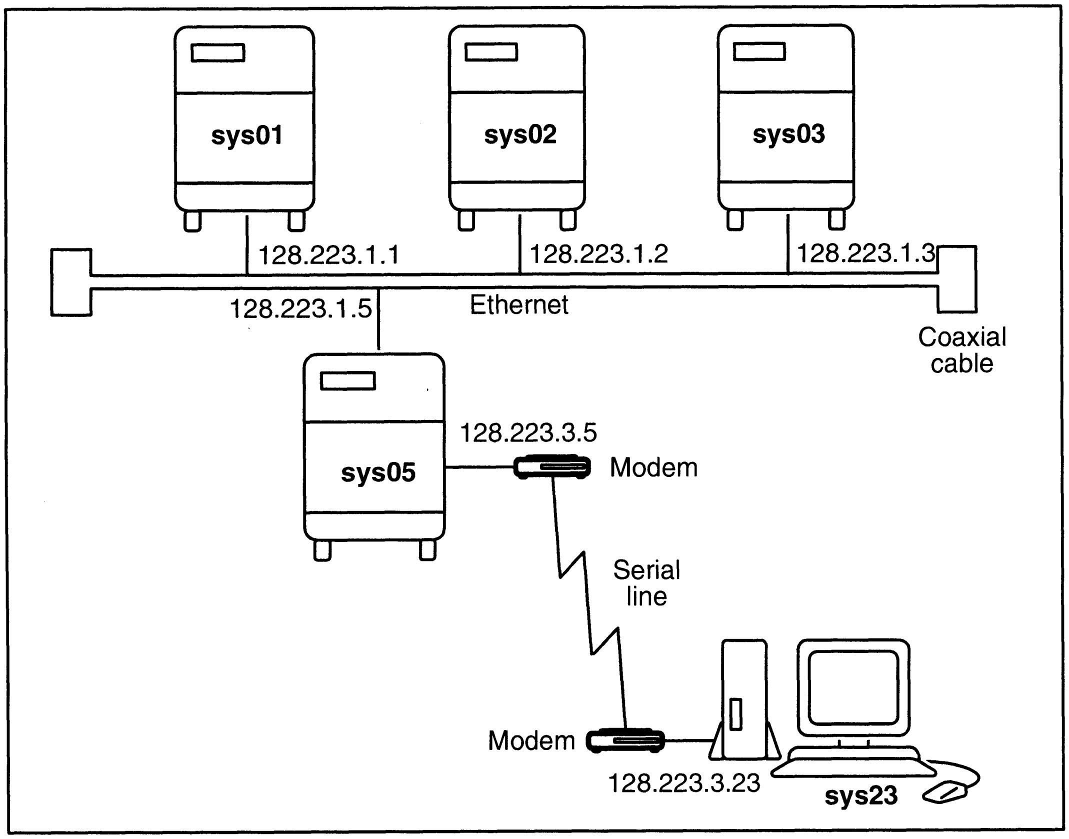


Figure D–1 SLIP Connection to a Host on an Ethernet LAN

Establishing a SLIP link between **sys23** and **sys05** enables either host to run TCP/IP applications (such as **telnet** and **ftp**) between the two systems. Furthermore, this link would enable **sys23** to connect to any system available to **sys05,** assuming appropriate routing entries are installed.

Configuring a modem to use SLIP

The capabilities and operating efficiency of SLIP depend in part on your modem configuration. You should select a modem that supports data compression and error correction to maximize throughput and minimize retransmissions.

Table D–1 shows typical front switches settings for a Microcom QX/4232hs error-correcting modem (Data General model 18901, part number 119-2087). These settings apply for both call-out and call-in SLIP services.

Tables D–1 and D–2 use the Hayes command set. If your modem does not use Hayes™ commands, substitute the equivalent commands.

Table D–1 Modem front switch settings

Switch	Setting	AT command	Remarks
1	Up	AT&D2	Respond to DTR
2	Down	none	AT Command Mode
3	Up	none	AT Command Mode
4	Up	ATE1	Command echo enabled
5	Up	ATS0=1	Auto-answer enabled
6	Down	AT&C1\D0	CD follows modem carrier, DSR and CTS on
7	Down	AT&C1\D0	CD follows modem carrier, DSR and CTS on
8	Down	AT–H0	Smart mode
9	Up	none	Not used
10	Up	none	Asynchronous operation

Table D–2 shows typical back switch settings for the same type of modem.

Table D–2 Modem back switch settings

Switch	Setting	AT command	Remarks
1	Down	AT\Q3	Bidirectional hardware flow control
2	Up	AT\Q3	Bidirectional hardware flow control
3	Down	AT\N3	Microcom Networking Protocol™ (MNP®) and LAPM auto-reliable mode
4	Up	AT\N3	MNP and LAPM auto-reliable mode
5	Down	AT\V2	Use V.42 result codes
6	Up	none	Restore configuration stored with **AT&W** or **AT*W** in the event of a reset, **ATZ**, or power up
7	Down	none	Read switches in the event of reset, **AT&F**, **ATZ** or power up
8	Down	AT\J0	BPS rate adjust disabled

Once you have set your modem up correctly, you can issue the **AT\S** command, or its equivalent on your system, to review additional configuration options. Table D–3 shows typical configuration options for the the Microcom QX/4232hs (Data General model 18901) modem. If you are not using this modem, use the table as a general guide to set up your modem.

Note that the serial port is set to operate at 19200 baud and the modem port is set to operate at its maximum rate of 9600 baud. For maximum throughput on data transfer, configure your serial port at the maximum available baud rate to be able to take advantage of data compressing modems.

Table D–3 Modem configuration options

Option	State	AT command
MODEM BPS	9600	`AT%G0`
MODEM FLOW	OFF	`AT\G0`
MODEM MODE	AUT	`AT\N3`
AUTO ANS.	on	`ATS0=1`
SERIAL BPS	19200	`AT%U0`
BPS ADJUST	off	`AT\J0`
ANSWER MESSGS	off	`atq2`
SERIAL FLOW	bhw	`at\q3`
PASS XON/XOFF	off	`at\x0`
PARITY	8n	`at`
BREAK	5	`at\k5`
EXIT CHAR	043	`ats2=43`
CMD ECHO	on	`ate1`
RESULTS	on	`atq0`
RESULT TYPE	mnpx	`atv1\v2`
CONN MNP–	0	`at-m0`
SPEED MATCH	1	`at%l1`
EQUALIZER	on	`at:e1`
FALLBACK	2	`at-q2`
DATA ECHO	off	`at\e0`
INACT TIMER	00	`at\t0`
AUTO RETRAIN	on	`at%e1`
COMPRESSION	all	`at%c3`
MAX BLK SIZE	256	`at\a3`
AUTO BUFF	0	`at\c0`
AUTO CHAR	000	`at%a0`
EMULATING HP	off	`at\h0`
PAUSE TIME	002	`ats8=2`
DTR	2	`at&d2`
CARR DET	1	`at&c1`
DSR	0	`at\d0`
RING IND	1	`at\r1`
SPKR CTRL	1	`atm1`
LEASE LINE	0	`at&l0`
ASYNC/SYNC	0	`At&m0`

Continued

Table D–3 Modem configuration options

Option	State	AT command
CTS/RTS	0	**at&r0**
LNG SPC DISC	off	**aty0**
SIM RING	0	**at:r0**
CD DELAY	000	**at:u0**
CTS DELAY	000	**at:v0**
DSR DELAY	000	**at:x0**
DISC DELAY	000	**at%d0**
REM CHAR	042	**at*s42**
REM ENABLE	off	**at*e0**
REM SEC	off	**at*r0**
RDLB ENABLE	on	**at&t4**
DIAL MODE	4	**atx4**
PULSE DIAL	60%	**at&p0**
V.24 TST MODES	off	**at%h0**
GUARD TONE	0	**at&g0**
ASYNC PROTOCOL	none	**at:k0**
KERMIT MARK	001	**at:q1**
PAR CHK	0	**at-p0**
MANUAL DIAL	0	**at:d0**
CONT 1200 BPS	off	**at*h0**
DETECT PHASE	on	**at-j1**
MNP EXT SVC	on	**at-k1**
RUN DIAGS	on	**at$d1**
READ SWITCHES	on	**at$k0**
O/A BUTTON	on	**at$o0**
S/A SWITCH	on	**at$s0**
TD ENABLE	on	**at$t0**
AUTOLOGON VIEW	on	**at$v1**
AUTOLOGON ANS	0	**at$a0**
CLOCK SOURCE	mdm	**at&x0**
BELL	on	**atb1**

For more information about SLIP, see the SLIP chapter in *Managing TCP/IP on the DG/UX™ System.*

End of Appendix

E Quick reference: setting up UUCP and modem for basic dial-in service

This appendix contains a quick summary of basic modem information.

Cable connections

VDC or VAC ports (DCE) AV8500/9500 DUART port DG Part #005-3625{6,7,8}	Other systems' DUART port DG Part #005-33701	AV530/4300/4600 DUART port 9–25 pin adpt. used w. VDC cable DG Part #005–38420
Host ... Modem	Host ... Modem	Host ... Modem
1 1	1 1	1 8
2 3	2 2	2 2
3 2	3 3	3 3
4 5	4 4	4 6
5 4	5 5	5 7
6 20	6 6	6 20
7 7	7 7	7 5
8 8	8 8	8 4
20 6	20 20	9 22

Port service behavior

Ttymon requires that the modem respond to the loss of the Data Terminal Ready (DTR) signal by disconnecting and entering command state. When the modem disconnects, it must also drop Data Carrier Detect (DCD) and Data Set Ready (DSR) signals.

When you log out, DTR drops and is not reasserted until after the modem drops DCD and DSR.

cu setup

1. Make **uucp** the owner of the port with a command such as:

```
# chown uucp /dev/tty09
```

2. Add a line such as the following to **/etc/uucp/Devices**:

```
Direct tty09,M - 9600 direct
```

3. Access the modem with a command such as:

```
# cu -l /dev/tty09
```

A response of "Connected" indicates that you have opened the port.

4. Issue a modem command:

   ATH

   The modem should respond with “OK”.

   The lack of an “OK” response could indicate:

   - The modem’s result codes are turned off.
   - The cable is not correct.
   - The modem is broken.

Modem commands

Begin each command with **AT**. End each command with a carriage return (Ctrl-M):

1. Display the modem status with commands such as:

\S	Microcom
&V	Most Hayes compatible type modems
L5 & **L7**	Multitech

2. The following commands are specific to the Microcom QX4232hs modem that Data General sells. Other modems may vary.

E1	command echo on
Q2	disable answer mode result codes **Q0** result codes always on **Q1** result codes always off
S0=1	Auto answer on first ring
\N3	MNP & LAPM auto-reliable
\J0	Bps rate adjust disable. This setting locks the serial port baud rate and eliminates the need for autobaud matching or hunt sequences. Refer to **ttydefs**, items 5 & 6.
&D2	Disconnect when DTR drops and enter command state.
&C1	CD follows the state of the carrier from the remote system
\D1	DSR follows off hook & CTS follows CD On other modems this is typically **&S1**
\Q4	Enable unidirectional XON/XOFF flow control.
\G0	Disable modem port flow control.

3. If you make any changes, store the new settings in a stored profile with ***W** on the Microcom modem or with **&W** on most other modems.

Port service setup

1. TTY definition label=M9600

   a. For modem lines, always use a label that starts with M.

2. hangup=yes

   yes **Ttymon** forces a hangup on the line by setting the speed to zero before setting the speed to the specified value and initializing the line.

   no Do not force hangup before initializing the line. Patches **dgux_5.4.1.p83** and **dgux_5.4.2.p63** correct a problem with hangup. This fix is included in DG/UX 5.4 Release 2.01.

3. connect on carrier=no

   yes Invoke service (/usr/bin/login) as soon as a carrier is detected. **ttymon** does not perform baud-rate searching and does not display **/etc/issue**. The "login:" prompt is displayed by the invoked service (**/usr/bin/login**).

   no **Ttymon** performs baud rate searching, displays **/etc/issue**, and displays the initial "login:" prompt.

4. bidirectional=yes

   yes Allow dialing out with **cu** or **uucp**, if the port is free. This setting causes **ttymon** to use the **uucp** lock files that are located in **/var/spool/locks**.

   no Allow dialing in only.

5. wait–read=1. This value depends on the behavior of your modem for returning result codes. If your modem is set to return result codes in answer mode, this value will probably need to be at least 8.

   none Display prompt without waiting for a character to be typed. If bidirectional=yes, 1 new-line character will still be required before displaying a prompt.

   0 Wait for any character before displaying a prompt.

   1 Wait for 1 new-line character before displaying a prompt.

   n Wait for *n* new-line characters before displaying a prompt.

6. timeout=0

   0 **Ttymon** never times out while it is in control. The invoked service (**/usr/bin/login**) has control from the time the "password:" prompt is first displayed until you get the shell prompt. The built-in time-out value for **login** is 60 seconds.

   n Allow *n* seconds of inactivity, from the time the "login:" prompt is first displayed until the "password:" prompt is displayed, before timing out and terminating the login.

ttydefs

1. Use the **sttydefs –l M9600** command to view the **/etc/ttydefs** file. This example assumes that you chose a “TTY Definition Label” of M9600 in the port service setup.

2. echo

initial flags	`-echo -echoe -echok`
final flags	`echo echoe echok`

3. Modem signals

   no clocal — implies –clocal. When CD drops, a disconnect is sensed and a hangup signal is sent to all processes in the group, causing them to be terminated.

   hupcl — causes DTR to be lowered on the last close of the port, i.e., logging out.

4. Flow control

software	ixon xoff
hardware	ctsxon rtsxoff –ixon –ixoff

5. Autobaud: no

   A “yes” indicates that **ttymon** will perform baud-rate searching and will match the baud rate of the first ASCII character. This field should normally be “no”, because the modem should be set so that its serial port is locked at its highest speed.

6. nextlabel: M9600

   A break character selects this label next. For the same reason as the previous step, the “nextlabel:” field should match the “ttylabel:”, which will disable hunt sequences.

Port service commands and log files

1. Display port service setup

   a. Commands
   `pmadm -l -p` *`pmtag`* **`-s`** *`svctag`*
   `admportservice -o list` *`svctag`*

   b. File accessed
   /etc/saf/*pmtag***/_pmtab**

   c. Information retrieved

   - FLGS
     u create **utmp** entry
     x service is disabled

- Port-monitor-specific example:

  ```
  /dev/tty10 b - /usr/bin/login 0 M9600
  ```

  device

  flags

b	bidirectional = y
c	connect on carrier = y
h	hangup = n
r	wait–read <> none

  wait–read value, "–" means "none"

  service command

  time-out value

  tty definition label

2. List all port monitors and their status

   ```
   sacadm -l
   ```

3. Stop a port monitor

   ```
   sacadm -k -p pmtag
   ```

4. Start a port monitor

   ```
   sacadm -s -p pmtag
   ```

   ```
   admportmonitor -o start pmtag
   ```

5. Associate a port monitor tag to a PID number

   ```
   who -l
   ```

6. Show who is controlling the modem port

   ```
   fuser -u /dev/ttyXX
   ```

7. Enable a port service

   ```
   pmadm -e -p pmtag -s svctag
   ```

8. Disable a port service

   ```
   pmadm -d -p pmtag -s svctag
   ```

9. Location of log files

   /var/saf/_log

   /var/saf/*pmtag*/log

End of Appendix

Glossary

The terms defined in this section are important to modems in the DG/UX environment.

ACU (Automatic Call Unit) A device that can dial a telephone number to establish a connection with a modem. An ACU can be separate from a modem but is usually included in the modem.

Automatic dial modem A modem that contains an ACU. Most modems today are of this type.

Baud rate The rate of data transfer in bits per second.

Bidirectional port service A port service that can call as well as receive. DG/UX UUCP supports connections with systems running old UUCP. To enable such a connection, you need to set up the **inittab** and **Systems** files correctly.

Datagram A self-contained package of data carrying the necessary information to route itself from source to destination. It is the unit of transmission in the IP protocol. To cross a particular network, a datagram is encapsulated inside a packet.

DCE Data communications equipment, a device such as a modem used for sending and receiving computer signals over a communications facility such as a telephone system or a Public Data Network.

Devices file A UUCP file, **/etc/uucp/Devices**, which contains entries for dial-out devices used to connect with remote systems. These entries include the location and line speed of automatic call units (modems).

Dialcodes file A UUCP file, **/etc/uucp/Dialcodes**, which contains dial-code abbreviations that entries in the **Systems** file may use in the phone number field.

Dialers file A UUCP file, **/etc/uucp/Dialers**, which contains character strings required to negotiate with automatic calling devices to connect to remote computers or terminals. This file contains sample chat scripts. A chat script comprises modem command strings that the **uucico** program sends to a modem when trying to connect to a remote system.

DTE Data terminal equipment, a business machine such as a computer, printer, or terminal.

HoneyDanBer The newer of the two versions of UUCP, the version used by the DG/UX system. Functional differences between the two versions of UUCP are reflected in the **/etc/inittab** file. The HoneyDanBer version allows for bidirectional login by adding a bidirectional port service (formerly implemented by respawning **uugetty** instead of **getty**). For more information on port services and the SAF facility, see *Managing the DG/UX™ System*.

host A computer being accessed by peripheral devices or other computers.

inittab file A file located in **/etc** that contains the system initialization table used by the **init** program. This file defines what actions the operating system is to take upon going to various run levels.

IP (Internet Protocol) A kernel-level protocol that defines unreliable, connectionless delivery of datagrams. An IP datagram contains the addresses of its source and destination, and the data transmitted. Connectionless service means that the protocol treats each datagram as a separate entity; the protocol can deliver packets out of sequence or can drop packets. IP defines the exact format of data as it travels through a network, but delivery of data is not guaranteed. IP was developed by the Defense Advanced Research Projects Agency (DARPA).

modem A device (MOdulator-DEModulator) that converts strings of bits on a serial line into audible tones for transmision through a telephone line or that converts audible tones from a telephone line into strings of bits.

packet The unit of data sent across a packet-switching network. The format of a packet typically is defined by the protocol.

parallel communication Transmitting all the bits in a byte or word simultaneously instead of serially. Printers often use this type of communication.

port A logical communications channel in a computer. An asynchronous port may have an asynchronous modem or terminal connected to it.

Public data network (PDN) Any of a number of high-speed communications networks that provide faster, more reliable data transmission than a public long distance telephone network.

SLIP The Serial Line Internet Protocol, a link-level protocol that lets you run network applications such as **telnet** and **ftp** over point-to-point serial lines.

server program A process running continuously in the server computer to respond to requests from client computers.

serial communication Transmitting each bit sequentially instead of in parallel. Modems use this type of communication.

server system A computer providing resources such as file storage for other computers.

service One or more remote programs, each implementing one or more remote procedures.

Systems file A UUCP file, **/etc/uucp/Systems**, which contains information needed by the UUCP software to establish a link to a remote host. It contains information such as the name of the remote host, when the host can be reached, the telephone number of its modem, and the login name and password expected by the remote system.

TCP (Transmission Control Protocol) A reliable connection-based transport method for transferring data over networks. It is used in conjunction with the Internet Protocol. The two are typically referred to as TCP/IP. It was developed by the Defense Advanced Research Projects Agency (DARPA).

Telephone network Either a public long-distance telephone network or a public data network.

ttymon A port monitor program that can monitor both modem and terminal lines.

UUCP (UNIX to UNIX CoPy) A set of programs and data files that allow you to transfer files and to execute remote commands between UNIX systems. In this section, the name UUCP refers to the entire system of programs and databases; the name **uucp** refers to the **uucp**(1) command itself.

UUCP connection A direct or a dial-up connection. A direct connection is simply a physical wire between machines. A dial-up connection uses telephone lines and a modem at both ends to connect machines.

End of Glossary

Index

Within the index, a range of page numbers indicates that the reference spans those pages.

TIPS ORDERING PROCEDURES

TO ORDER

1. An order can be placed with the TIPS group in two ways:
   a. MAIL ORDER – Use the order form on the opposite page and fill in all requested information. Be sure to include shipping charges and local sales tax. If applicable, write in your tax exempt number in the space provided on the order form.

   b. Send your order form with payment to:

   Data General Corporation
   ATTN: Educational Services/TIPS G155
   4400 Computer Drive
   Westboro, MA 01581–9973

   c. TELEPHONE – Call TIPS at (508) 870–1600 for all orders that will be charged by credit card or paid for by purchase orders over $50.00. Operators are available from 8:30 AM to 5:00 PM EST.

METHOD OF PAYMENT

2. As a customer, you have several payment options:
   a. Purchase Order – Minimum of $50. If ordering by mail, a hard copy of the purchase order must accompany order.

   b. Check or Money Order – Make payable to Data General Corporation. Credit Card – A minimum order of $20 is required for MasterCard or Visa orders.

SHIPPING

3. To determine the charge for UPS shipping and handling, check the total quantity of units in your order and refer to the following chart:

Total Quantity	Shipping & Handling Charge
1–4 Items	$5.00
5–10 Items	$8.00
11–40 Items	$10.00
41–200 Items	$30.00
Over 200 Items	$100.00

If overnight or second day shipment is desired, this information should be indicated on the order form. A separate charge will be determined at time of shipment and added to your bill.

VOLUME DISCOUNTS

4. The TIPS discount schedule is based upon the total value of the order.

Order Amount	Discount
$0–$149.99	0%
$150–$499.99	10%
Over $500	20%

TERMS AND CONDITIONS

5. Read the TIPS terms and conditions on the reverse side of the order form carefully. These must be adhered to at all times.

DELIVERY

6. Allow at least two weeks for delivery.

RETURNS

7. Items ordered through the TIPS catalog may not be returned for credit.
8. Order discrepancies must be reported within 15 days of shipment date. Contact your TIPS Administrator at (508) 870–1600 to notify the TIPS department of any problems.

INTERNATIONAL ORDERS

9. Customers outside of the United States must obtain documentation from their local Data General Subsidiary or Representative. Any TIPS orders received by Data General U.S. Headquarters will be forwarded to the appropriate DG Subsidiary or Representative for processing.

TIPS ORDER FORM

Mail To:
Data General Corporation
Attn: Educational Services/TIPS G155
4400 Computer Drive
Westboro, MA 01581 - 9973

BILL TO:	SHIP TO: (No P.O. Boxes - Complete Only If Different Address)
COMPANY NAME________________	COMPANY NAME________________
ATTN:________________	ATTN:________________
ADDRESS________________	ADDRESS (NO PO BOXES)________________
CITY________________	CITY________________
STATE____________ ZIP________	STATE____________ ZIP________

Priority Code ____________________ (See label on back of catalog)

Authorized Signature of Buyer (Agrees to terms & conditions on reverse side) ________________ Title ________________ Date ________ Phone (Area Code) ________ Ext. ______

ORDER #	QTY	DESCRIPTION	UNIT PRICE	TOTAL PRICE

ORDER TOTAL	
Less Discount See B	–
SUB TOTAL	
Your local* sales tax	+
Shipping and handling – See A	+
TOTAL – See C	

Tax Exempt # or Sales Tax (if applicable) ________________

A SHIPPING & HANDLING

□ UPS	ADD
1–4 Items	$5.00
5–10 Items	$8.00
11–40 Items	$10.00
41–200 Items	$30.00
200+ Items	$100.00

Check for faster delivery

Additional charge to be determined at time of shipment and added to your bill.

□ UPS Blue Label (2 day shipping)

□ Red Label (overnight shipping)

B VOLUME DISCOUNTS

Order Amount	Save
$0–$149.99	0%
$150–$499.99	10%
Over $500.00	20%

C PAYMENT METHOD

□ Purchase Order Attached ($50 minimum)
P.O. number is__________. (Include hardcopy P.O.)

□ Check or Money Order Enclosed

□ Visa □ MasterCard ($20 minimum on credit cards)

Account Number □□□□□□□□□□□□□□□□ Expiration Date □□□□

____
Authorized Signature
(Credit card orders without signature and expiration date cannot be processed.)

THANK YOU FOR YOUR ORDER

PRICES SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.
PLEASE ALLOW 2 WEEKS FOR DELIVERY.
NO REFUNDS NO RETURNS.

* Data General is required by law to collect applicable sales or use tax on all purchases shipped to states where DG maintains a place of business, which covers all 50 states. Please include your local taxes when determining the total value of your order. If you are uncertain about the correct tax amount, please call 508–870–1600.

Form 702
Rev. 8/87

DATA GENERAL CORPORATION
TECHNICAL INFORMATION AND PUBLICATIONS SERVICE

TERMS AND CONDITIONS

Data General Corporation ("DGC") provides its Technical Information and Publications Service (TIPS) solely in accordance with the following terms and conditions and more specifically to the Customer signing the Educational Services TIPS Order Form. These terms and conditions apply to all orders, telephone, telex, or mail. By accepting these products the Customer accepts and agrees to be bound by these terms and conditions.

1. CUSTOMER CERTIFICATION

Customer hereby certifies that it is the owner or lessee of the DGC equipment and/or licensee/sub–licensee of the software which is the subject matter of the publication(s) ordered hereunder.

2. TAXES

Customer shall be responsible for all taxes, including taxes paid or payable by DGC for products or services supplied under this Agreement, exclusive of taxes based on DGC's net income, unless Customer provides written proof of exemption.

3. DATA AND PROPRIETARY RIGHTS

Portions of the publications and materials supplied under this Agreement are proprietary and will be so marked. Customer shall abide by such markings. DGC retains for itself exclusively all proprietary rights (including manufacturing rights) in and to all designs, engineering details and other data pertaining to the products described in such publication. Licensed software materials are provided pursuant to the terms and conditions of the Program License Agreement (PLA) between the Customer and DGC and such PLA is made a part of and incorporated into this Agreement by reference. A copyright notice on any data by itself does not constitute or evidence a publication or public disclosure.

4. LIMITED MEDIA WARRANTY

DGC warrants the CLI Macros media, provided by DGC to the Customer under this Agreement, against physical defects for a period of ninety (90) days from the date of shipment by DGC. DGC will replace defective media at no charge to you, provided it is returned postage prepaid to DGC within the ninety (90) day warranty period. This shall be your exclusive remedy and DGC's sole obligation and liability for defective media. This limited media warranty does not apply if the media has been damaged by accident, abuse or misuse.

5. DISCLAIMER OF WARRANTY

EXCEPT FOR THE LIMITED MEDIA WARRANTY NOTED ABOVE, DGC MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE ON ANY OF THE PUBLICATIONS, CLI MACROS OR MATERIALS SUPPLIED HEREUNDER.

6. LIMITATION OF LIABILITY

A. CUSTOMER AGREES THAT DGC'S LIABILITY, IF ANY, FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED THE CHARGES PAID BY CUSTOMER FOR THE PARTICULAR PUBLICATION OR CLI MACRO INVOLVED. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO CLAIMS FOR PERSONAL INJURY CAUSED SOLELY BY DGC'S NEGLIGENCE. OTHER THAN THE CHARGES REFERENCED HEREIN, IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOST PROFITS AND DAMAGES RESULTING FROM LOSS OF USE, OR LOST DATA, OR DELIVERY DELAYS, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY THEREOF; OR FOR ANY CLAIM BY ANY THIRD PARTY.

B. ANY ACTION AGAINST DGC MUST BE COMMENCED WITHIN ONE (1) YEAR AFTER THE CAUSE OF ACTION ACCRUES.

7. GENERAL

A valid contract binding upon DGC will come into being only at the time of DGC's acceptance of the referenced Educational Services Order Form. Such contract is governed by the laws of the Commonwealth of Massachusetts, excluding its conflict of law rules. Such contract is not assignable. These terms and conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior oral or written communications, agreements and understandings. These terms and conditions shall prevail notwithstanding any different, conflicting or additional terms and conditions which may appear on any order submitted by Customer. DGC hereby rejects all such different, conflicting, or additional terms.

8. IMPORTANT NOTICE REGARDING AOS/VS INTERNALS SERIES (ORDER #1865 & #1875)

Customer understands that information and material presented in the AOS/VS Internals Series documents may be specific to a particular revision of the product. Consequently user programs or systems based on this information and material may be revision–locked and may not function properly with prior or future revisions of the product. Therefore, Data General makes no representations as to the utility of this information and material beyond the current revision level which is the subject of the manual. Any use thereof by you or your company is at your own risk. Data General disclaims any liability arising from any such use and I and my company (Customer) hold Data General completely harmless therefrom.

Managing
Modems and UUCP
on the
DG/UX™ System

069–000698–00

Cut here and insert in binder spine pocket