# intel®

# XENIX* 286
# COMMUNICATIONS GUIDE

*XENIX is a trademark of Microsoft Corporation.

# XENIX* 286
# COMMUNICATIONS GUIDE

Order Number: 174461-001

*XENIX is a trademark of Microsoft Corporation.

| REV. | REVISION HISTORY | DATE |
|---|---|---|
| -001 | Original issue | 11/84 |

# CONTENTS

## CONTENTS                                                              PAGE

This manual provides information about installing and administering two XENIX 286 communications software subsystems: uucp and Micnet. This manual should be used in conjunction with the command and file entries in the *XENIX 286 Reference Manual* that describe the communications commands and files. Each subsequent chapter of this manual references related command and file entries in the beginning of the chapter.

## Prerequisites

This manual assumes that you, the reader, have a basic understanding of XENIX or UNIX.

## Manual Organization

This manual contains these chapters:

1. **Introduction:** Prerequisites, manual organization, and notation.

2. **Uucp:** Installing and administering uucp networks.

3. **Micnet:** Installing and administering Micnet networks.

This manual contains one appendix:

A. **Related Publications:** Descriptions and ordering information for all XENIX 286 Release 3 manuals and any other publications referenced by this manual.

## Notation

These notational conventions are used in this manual:

● Literal names are bolded where they occur in text, e.g., **/sys/include**, **printf**, **dev_tab**, **EOF**.

● Syntactic categories are italicized where they occur and indicate that you must substitute an instance of the category, e.g., *filename*.

- Program listings and examples of XENIX commands or dialogue are set in a distinct font.

- In examples of dialogue with the XENIX 286 system, characters entered by the user are printed in bold type, e.g., **cat myfile.**

- In syntax descriptions, optional items are enclosed in brackets, e.g., [ -n ].

- Items that can be repeated one or more times are followed by an ellipsis ( ... ).

- Items that can be repeated zero or more times are enclosed in brackets and followed by an ellipsis ( [ ]... ).

- A choice between items is indicated by separating the items with vertical bars ( | ).

This chapter describes how to install and administer uucp communications software on your XENIX system. Uucp sites can be connected either over telephone lines using modems or over direct serial lines. One application of uucp is to connect Micnet sites that are geographically separated and that cannot be connected with a direct line. Capabilities provided by uucp include

• The **uucp** command for copying files to or from remote sites

• The **uux** command for executing commands at remote sites

This chapter should be read in conjunction with the following entries in the section "Commands" of the *XENIX 286 Reference Manual*: **cu, uuclean, uucp** (includes **uulog** and **uuname**), **uustat, uusub, uuto** (includes **uupick**), and **uux**.

A uucp system is a set of files and programs that enable you to use the **uucp** and **uux** commands to transfer files and commands between computers connected by your communication system. Before you can use the **uucp** and **uux** commands, you must install the uucp system by creating or modifying a number of uucp system files.

The uucp system actually provides two different methods of interaction with other computers. One method requires a dial-in line through which remote computers can log in and transfer files and commands. With this method, your computer is called a "dial-in site." The other method requires a dial-out line through which your computer can call other computers. With this method, your computer is called a "dial-out site." Each method requires its own set of uucp system files.

Although you can install files for both methods, only one method can be used at a particular time on a particular line. However, you can use alternate methods at different times of the day by creating a shell script that automatically enables or disables the line to permit dialing in or dialing out. Also, if a site has multiple lines, some may be dial-out lines and some may be dial-in lines.

The following sections explain how to install a dial-in or dial-out communications line using a modem. This chapter uses the Hayes Smartmodem 1200 as an example modem. This is not an endorsement of this particular brand of modem, and another modem can be substituted. The last section of this chapter lists the **dial** program used to communicate with the modem. An executable (binary) version of this program is supplied with your XENIX system. This program is specific to the Hayes Smartmodem 1200. If you use another modem, either the other modem must interface to the **dial** program in the same way as the Hayes modem or you must write a new **dial** program for the new modem, using the listing in this chapter as an example.

Uucp can also be used with direct serial lines instead of dial-up connections. Differences in procedures for direct serial lines are included as needed.

## What You Need

To install a uucp system using a telephone line on your computer, you will need

* A modem (e.g., Hayes Smartmodem 1200)

* A standard telephone jack for access to the telephone system (touch tone line required)

* An RS-232 serial port on your computer

* An RS-232 cable to connect the serial line to the modem

For proper operation of the modem, the RS-232 cable must provide the pin connections shown below. Note that the computer's serial connector must have a DTE (Data Terminal Equipment) configuration. The modem is assumed to have a DCE (Data Communications Equipment) configuration.

### Pin Connections

| Computer (DTE) | Modem (DCE) | Description | Direction |
|---|---|---|---|
| 1 | 1 | Protective Ground | NA |
| 2 | 2 | Transmit Data | to modem |
| 3 | 3 | Receive Data | from modem |
| 6 | 6 | Data Set Ready | from modem |
| 7 | 7 | Signal Ground | NA |
| 8 | 8 | Carrier Detect | from modem |
| 20 | 20 | Data Terminal Ready | to modem |
| 22 | 22 | Ring Indicator | from modem |

Note that if a terminal was previously connected to the serial port that you are using for your modem, then the port must be rejumpered to place it in the DTE configuration. Refer to the appropriate hardware reference manual for instructions on configuring DTE serial ports.

The Hayes modem and other telephone modems are typically FCC registered devices, and you must notify your telephone company that such a device is being used. The FCC (Federal Communications Commission) may impose other restrictions on the use of your modem; any such restrictions should be described in the manual(s) for your modem.

Finally, since many of the tasks you must perform require special permissions, you must log in to your computer's super-user account before performing them. Check with your system's administrator before proceeding with this installation, or refer to the *XENIX 286 System Administrator's Guide* for instructions on how to log in as the super-user.

2-3

## Installing the Modem

Installing the modem is the first step in creating a complete communication system. The installation has four steps:

1.    Choose a serial line properly configured as DTE.

2.    Set the dialing configuration.

3.    Connect the modem.

4.    Test the connection.

The following sections explain each step in detail.

### Choose a Serial Line

You must choose the RS-232 serial line you wish to use with the system and connect it to the modem. If there are no lines available, you must install a new serial line or make one available by removing the device connected to it. If you remove a terminal, make sure no one is logged in on that terminal.

Once you have chosen a serial line, find the name of the device special file associated with the line. The file name should have the form

    /dev/tty*xy*

where *xy* specifies the line. For example, **/dev/ttyc0** names the first line on the iSBC®
188/48 board.

Be sure that the selected line supports modem control. For example, **/dev/ttyf0,** the
second iSBC 286/10 serial port, does not support modem control.

### Configure the Modem

In this communication system, your modem can be used to send and receive calls. You must establish the correct options on the modem. For example, for the Hayes Smartmodem 1200, follow these steps:

1.    Remove the front cover of the modem and locate the 8-pin configuration switch.
      (See the *Smartmodem 1200 Owner's Manual* for instructions on how to remove the
      cover and then locate the switch.)

2.    Set the pins on the configuration switch to the following positions:

pins              1     2     3     4     5     6     7     8
positions         up    up    down  down  up    up    up    down

- Switch 1 (up) enables system control of Data Terminal Ready (DTR). The computer drops DTR if it wants the modem to hang up. The modem drops DTR if the line is hung up at the other end.

- Switch 2 (up) causes the Smartmodem to respond to commands with response messages in English.

- Switch 3 (down) causes the Smartmodem to send response messages to the computer (either numbers or English words).

- Switch 4 (down) disables echoing of commands sent to the Smartmodem.

- Switch 5 (up) enables automatic answering of an incoming call.

- Switch 6 (up) causes the modem to hang up on loss of Carrier Detect (CD).

- Switch 7 (up) indicates a single-line connection to an RJ11 modular telephone jack (i.e., a standard single-phone connection).

- Switch 8 (down) enables Smartmodem command recognition.

These switches are described in greater detail in the *Smartmodem 1200 Owner's Manual*.

3.    Replace the front cover.

For other modems, the equivalent functions may be set by switches, be set by sending commands to the modem, or be the power-on defaults for the modem. To use another modem, you may have to modify the **dial** program listed at the end of this chapter. Most modems advertised as "Hayes-compatible" can be configured to work with the existing **dial** program.


## Connect the Modem

Once your modem's dialing configuration is set, you are ready to connect the modem to your computer.

**NOTE**

You should turn off your computer system's power before connecting the modem.

Review your modem's installation instructions and then follow these steps:

1.  Connect the RS-232 serial cable to the serial line connector on the modem, then to the serial line connector on your computer.  Make sure the cable is properly connected.

2.  Plug the telephone line cable into the telephone connector on the modem, then into the telephone wall jack.

3.  Plug in the modem's power cord.

## Test the Connection

As the last step of the modem installation, you should test the modem to make sure that it can initiate a call and send and receive data.  Once you have verified that the modem is working, you can begin to use the communication system.  To test the modem, follow these steps:

1.  Turn on the modem's power.

2.  Log in as the super-user.

3.  Disable logins on the modem's serial line by typing

        disable /dev/tty*xy*

    where *xy* specifies the serial line.

4.  Edit the file **/usr/lib/uucp/L-devices.**  Add a line (if it is not there already) of the form

        DIR tty*xy* 0  1200

    where *xy* specifies the serial line and 1200 is the baud rate.  Do not insert any blank lines into the file, as uucp cannot handle them.

5.  If the modem has a speaker, turn up the volume so that you will be able to hear the modem dialing.

6.  Use the **cu** command to connect your terminal to the modem line:

        % cu -l/dev/tty*xy*  dir
        connected

    **cu** responds with "connected".  Note that the device special file specified must have read permission.

7.    Enter the command and dialing string needed by the modem to dial a specific remote system.  For example:

ATDT9,7691234

Command and dialing string format should be described in the manual for the modem.  The example string is for the Hayes Smartmodem 1200.  "AT" gets the modem's attention, "D" dials, "T" indicates tone dialing, and the comma in the string indicates a pause to wait for a dial tone (an outside line).  The modem will dial the number.  If your modem has a speaker and the volume is up, you should hear each digit being dialed.

8.    If dialing the number results in a busy signal, then the modem will hang up. Otherwise, you will be connected to the remote system.  The modem may send back a message indicating that a connection has been made, e.g.:

CONNECT

9.    Enter a carriage return.  You should get a login message from the remote system.

If there is not a remote system that you can call, a simpler test is to call the number of the phone line that you are calling out on.  You should hear (if your modem has a speaker) the modem make the connection, get a busy signal, and then hang up.  This tests the computer-modem-phone connections and the modem's dialing function but does not test data transmittal through the connection.

## Creating a Dial-in Line

You can create a dial-in line for use by remote terminals or computers by enabling the modem's serial line with the **enable** command after making sure your modem's serial line has an appropriate entry in the file **/etc/ttys**.  Once the line is enabled, any user at a remote terminal or computer can log in to your computer by calling your modem and following the ordinary login procedure.  To create a dial-in line, follow these steps:

1.    Log in as the super-user.

2.    Disable login dialogue for the modem's serial line by typing

disable /dev/tty*xy*

where *xy* specifies the serial line.

3.    Use the **cat** command to examine the contents of the **/etc/ttys** file.

4.    Locate the entry in this file that corresponds to your serial line.  The correct entry contains the name of your serial line and must have the form

03tty*xy*

where *xy* specifies your serial line.

5.    If necessary, use a XENIX text editor to change the entry or create a new entry for your serial line. Make sure the first two characters in the entry are "03". The first digit is "0" for disabling the line; it will be changed to "1" when the line is enabled. (Note that editing this file and changing an entry's first digit to "1" will not enable the line. The file is only interpreted by the kernel when it gets a signal to do so, i.e., when an **enable** command is given for a device with an entry in the file that indicates that it is currently disabled.) The second digit is "3" to indicate a line that is initially at 1200 baud. For more information about the format of this file, see the entries **getty** in the section "Commands" and **ttys** in the section "Files" in the *XENIX 286 Reference Manual*.

6.    Save the edited file and exit the editor.

7.    You may need to remake the device special file for the line to set bit 6 in the minor device number, to indicate that modem control should be used on the line. Execute the command

        l   /dev/tty*xy*

where x̲y̲ specifies your serial line, to list the line's directory entry, which includes the device major and minor numbers for the line. Make a note of the major number and the minor number. If bit 6 (64) of the minor number is set, modem control is already enabled on the line. Otherwise, execute the following commands:

        rm   /dev/tty*xy*
        mknod   /dev/tty*xy* c  *maj  min*

where *xy* specifies your serial line, *maj* is the device major number for the line, and *min* is the new device minor number for the line, equal to the previous minor number plus 64. Note that setting bit 6 of the minor number is done *only* for dial-in lines.

8.    Type

        enable  /dev/tty*xy*

where *xy* specifies your serial line. This enables the line for logins.

Your computer can now receive calls from remote terminals or computers and prompt for a login name.

## Creating a Dial-out Line

### NOTE

Your modem's serial line cannot be both dial-in and dial-out at the same time. However, you can alternate between dial-in and dial-out at different times of the day by enabling or disabling the serial line as needed. Make sure you wait at least one minute between each invocation of the **enable** and **disable** commands.

**NOTE**

The same device special file cannot be used for both dial-in and dial-out. Dial-in lines must have a device minor number with bit 6 set, for modem control. Dial-out lines must have bit 6 of the minor number clear.

You can create a dial-out line by creating the call unit files, editing the **/usr/lib/uucp/L-devices** file, and then disabling the modem's serial line. With a dial-out line, you can call and log in to other computers using the **cu** command. **cu** uses the **/usr/lib/uucp/L-devices** file to locate the modem's serial line and line speed. The call unit files are actually specially named files that are linked to your modem's serial line and used by uucp system programs.

The following sections describe how to create the necessary files and enable your line.

## Create the Call Unit Files

You must create two new device files, called "call unit" files. Follow these steps:

1. Log in as the super-user.

2. Check for any existing call unit files by using the l command. Type

        l  /dev/cu*

   and examine the output. Call unit file names have the form

        /dev/cua*xy*

   and

        /dev/cul*xy*

   where *xy* specifies the corresponding serial line. If these files exist, you can skip the remaining steps and continue with editing the **/usr/lib/uucp/L-devices** file.

3. Use the **mknod** command to create the call unit files. For example, if your serial line is **/dev/ttyc0**, enter the commands

        mknod  /dev/cuac0  c  *maj  min*
        mknod  /dev/culc0  c  *maj  min*

   where *maj* and *min* are the line major and minor numbers. The major number should be identical to the major number of the device special file for the line. The minor number should be the same as the minor number for the dial-in line but with bit 6 clear.

4.  Use the **chmod** command to change the access mode of the call unit files to read and write for everyone.  For example, the command

    chmod ugo + rw /dev/cuac0 /dev/culc0

    sets the appropriate permissions for the **/dev/cuac0** and **/dev/culc0** files.


## Edit the L-devices File

The **/usr/lib/uucp/L-devices** file defines the devices you intend to use to implement the dial-out line.  The file is also used by programs in the uucp system (as described later).  The file contains one or more entries of the form

   *type  line  call-unit  speed*

where *type* must be "ACU" if you are using a modem or "DIR" if you are using a direct serial line.  ("ACU" stands for "Automatic Call Unit", recalling a time when dialing was done by separate hardware attached to the modem, hardware that is incorporated within most modern modems.)  *line* and *call-unit* are the **culxy** and **cuaxy** call unit file names respectively; if you are using a direct serial line, then *line* should be the line's file name in **/dev** and *call-unit* should be **0**.  *speed* is the line speed or baud rate for transmissions. The line and call-unit files are assumed to be in the **/dev** directory, so the full path name is not required.  For example, the entry

   ACU  culc0  cuac0  1200

defines **/dev/culc0** as the line, **/dev/cuac0** as the call-unit, and **1200** as the line speed.

With the Hayes modem, *speed* must be set to 1200.

If the same line were instead used for a direct serial line at 9600 baud, you would include the entry

   DIR  ttyc0  0  9600

Blank lines are not allowed in the file and may cause it to be used incorrectly.

Use a XENIX text editor to edit or create the file.  If creating the file, make sure you create the file in the **/usr/lib/uucp** directory.  Then use the **chmod** command to give the file read permission for everyone.  Read permission for all users is required for uucp to function properly.


## Disable the Serial Line

A serial line must be disabled before it can be used as a dial-out line.  Follow these steps:

1.  Make sure your modem has been installed and tested.

2.  Log in as the super-user.

3.   Disable the modem's serial line by typing

        disable  /dev/tty*xy*

     where *xy* specifies your modem's serial line.  If the line was already disabled, the command displays a status message that you can ignore.

You can now call other computers that have dial-in lines by using the **cu** command.  **cu** is described in the section "Commands" in the *XENIX 286 Reference Manual*.

## Installing a Uucp System

To use uucp, you must first create or modify a number of uucp system files.  Recall that uucp provides two different methods of interacting with other computers.  One method requires a dial-in line through which remote computers can log in and transfer files and commands.  The other method requires a dial-out line through which your computer can call other computers.  Note that despite the "dial-in" and "dial-out" terms, both methods may also use direct serial lines instead of phone lines and modems. You must set up your files differently, depending on whether your computer is a "dial-in site" or a "dial-out site."

Although you can install files for both methods, only one method can be used at a particular time on a particular line.  However, you can use alternate methods at different times of the day, by creating a shell script that automatically enables or disables the line to permit dialing in or dialing out.  Also, if a site has multiple lines, some may be dial-out lines and some may be dial-in lines.

The following sections explain how to create files for both methods of interaction. They also explain how to create a transmission schedule and develop a **cron** script to implement the schedule.

### Choose a Uucp Site Name

In a uucp system, every computer belongs to a given "site."  A site is any computer or any Micnet network that can communicate with the uucp system.  To distinguish one site from another, every site must have a unique "site name."  A site name is any combination of letters and digits that begins with a letter and is no more than seven characters long.  The site name may then be used in **uucp** and **uux** commands to direct transmissions to the appropriate computer or Micnet network.

The site name should suggest some characteristic of the site, such as its location or affiliation.  For example, a site in Chicago can be named "chicago", or a site in the shipping department can be named "shipg".  The site name must be unique.  That is, no other computer that calls your computer or is called by your computer can have the same site name.

Once you have chosen a site name, you will need to add it to the **/etc/systemid** file as described in the next section.

## Create the systemid File

Each site must have a **/etc/systemid** file. The file defines the site name of the given site and associates the site with a Micnet network, if any. The file has the form

    *site-name*
    [ *machine-name* ]

where *site-name* is the name of the given site, and *machine-name* is the Micnet machine name for that computer. The machine name is optional only if the computer is not connected to a Micnet network. For example, the entries

    chicago
    brewster

define a site named "chicago" with a Micnet machine name of "brewster".

Since uucp systems are often created after a Micnet network has been established, the **systemid** file usually already exists on a given site. In this case, you must add the site name to the beginning of each **systemid** file on each computer in the Micnet network. Note that you may give more than one machine name if desired, but each name must be on a separate line. **systemid** is described in the section "Files" in the *XENIX 286 Reference Manual.*

## Create a Dial-in Site

You can create a dial-in site by setting up the uucp login information required by other computers that wish to log in and transfer files and commands. This information consists of the following:

- One or more **/etc/passwd** file entries

- User access information in the **/usr/lib/uucp/USERFILE** file

You can create this information by using a XENIX text editor and modifying or creating the appropriate files. The following sections explain the required format of the information.

Once the information is installed, you can enable the system for logins by creating a dial-in line. See the section "Creating a Dial-in Line" earlier in this chapter.

**Create Uucp Login Entries**

A dial-in site must provide a login entry for the sites that call it. This entry or entries must be placed in the **/etc/passwd** file. A uucp login entry has the same form as an ordinary user login entry but gives a special login directory and login program instead of the normal user directory and shell. Your XENIX system as supplied contains one uucp login entry, with user ID 4 and group ID 4. This login can be used by all dial-in sites, or additional login entries can be created. Any new login entries should have 4 as their group ID. To create a new uucp login entry, follow these steps:

1.   Choose a login name and a user ID for the uucp login. The name may be any combination of letters and digits that is no more than eight characters long. The user ID must be an integer in the range 1 to 65535. Make sure the name and user ID are unique; a login entry must not have the same name or user ID as any other login entry. The group ID for the new entry will be that of the uucp group, 4.

2.   Execute the command

         chmod ugo + w  /etc/passwd

     to get write permission on the **/etc/passwd** file.

3.   Invoke a XENIX text editor, giving **/etc/passwd** as the file to edit.

4.   Move to the end of the file and insert the new login entry using the form

         *login-name::user-ID*:4::/usr/spool/uucppublic:/usr/lib/uucp/uucico

     where *login-name* is the login name you have chosen, and *user-ID* is the user ID you have chosen. For example, if you have chosen "uuchcg" for the login name and "12" for the user ID, add the entry

         uuchcg::12:4::/usr/spool/uucppublic:/usr/lib/uucp/uucico

     to the end of the file.

5.   Save the new file and exit the editor.

6.   Create a new password for the login with the **passwd** command. Type

         passwd *login-name*

     where *login-name* is the login name you have chosen. The command will ask you to type the new password twice. It will then add the encrypted password to the new login entry. The command also makes the file **/etc/passwd** read-only again.

Note that you can create new login entries for each site that calls your site or use one entry for all sites.

### Create the USERFILE

The **/usr/lib/uucp/USERFILE** file defines which directories on the local machine a given site can access using the **uucp** and **uux** commands. You should create one **USERFILE** entry for each site or user with a login entry in the **/etc/passwd** file. Each entry has the form

[ *login* ], [ *site-name* ] *pathname...*

where *login* is the login name for a given site, *site-name* is the site name of a given site, and *pathname* is the full path name of a directory that the given site may access. Multiple path names can be given if separated by spaces. The *login* and *site-name* fields are both optional.

Blank lines are not allowed in the file and may cause it to be used incorrectly.

The following rules explain how access is granted for each entry:

• A calling site is granted access to those directories defined in the first entry containing its site name.

• A calling site with a site name that does not appear in an entry is granted access to the directories defined for the first entry with no site name.

• A local user is granted access to those directories defined in an entry containing his or her login name.

• A user whose login name does not appear in an entry is granted access to directories defined in the first entry with no login name.

You may have more than one entry with the same login name if you wish, but you must make sure that at least one of these entries also has the site name of any calling site that can log in with that name, or that at least one entry has no site name.

For example, consider the following entries:

```
uuchcgo, chicago /usr /usr2/market
uucp, /usr/vendor
schmidt, /usr2/market /usr/vendor
,        /usr/spool/uucppublic
```

The site named "chicago" has access to files in the directories **/usr** and **/usr2/market** via the login **uuchcgo**. Sites logging in as uucp will be granted access to **/usr/vendor** only. A local user named "schmidt" is granted access to the directories **/usr2/market** and **/usr/vendor**. All other users (both local access and remote sites) have access to **/usr/spool/uucppublic** only.

## Create a Dial-out Site

You can create a dial-out site by installing the dialing information needed by your system to call and log in to other computers. This information consists of the following:

- Dialing abbreviations for remote computers in the **/usr/lib/uucp/L-dialcodes** file (optional)

- Information about logins on remote computers in the **/usr/lib/uucp/L.sys** file

- A transmission schedule in the form of a shell script to be run periodically by the **cron** program

You can create this information by using a XENIX text editor and modifying or creating the appropriate files. The following sections explain the required format of the information.

Once the information is installed, you can enable the system for calling other computers by creating a dial-out line. See the section "Creating a Dial-out Line" earlier in this chapter.


### Create the L-dialcodes File

The **/usr/lib/uucp/L-dialcodes** file defines abbreviations for often-used telephone prefixes and area codes. You may use these abbreviations in the **L.sys** file when forming the telephone numbers of remote sites.

The **L-dialcodes** file may contain one or more entries of the form

   *abbreviation   dial-sequence*

where *abbreviation* is any combination of letters and digits that begins with a letter, and *dial-sequence* is any combination of digits that represents a telephone prefix, area code, or any other part of a telephone number. For example, the entry

   tw  555

defines the abbreviation "tw" to be the telephone prefix "555".

Blank lines are not allowed in the file and may cause it to be used incorrectly.

Note that use of **L-dialcodes** and of dialing abbreviations is optional and is not needed for installing uucp.

### Create the L.sys File

The **/usr/lib/uucp/L.sys** file defines the names, telephone numbers, and login information for all remote sites that the local site communicates with directly. The file contains one or more entries of the form

     *site-name time device speed phone login*

where *site-name* is the name of the site to be called, *time* is a combination of letters and digits that gives the days of the week and times when the given site can be called, *device* is the name of the device through which the given site is to be called, *speed* is the line speed for the call, *phone* is the phone number of the given site, and *login* is the login information required to log in to the given site.

Blank lines are not allowed in the file and may cause it to be used incorrectly.

The field *time* defines when the given site can make calls to other sites. *time* has the form

     *days times*

where *days* is a list of one or more days of the week, and *times* is a range of times of day. The days of the week may be "Su", "Mo", "Tu", "We", "Th", "Fr", "Sa", "Wk" (for any week-day), "Any" (for any day), and "Never" (for call by special request only). The time of day must be given as a four-digit number. The first pair of digits gives the hour (in terms of a 24-hour clock), and the second pair gives the minutes. A range of times is a pair of times of the day separated by a hyphen (-). For example, the entry

    MoTuTh0800-1230

allows the given site to be called any Monday, Tuesday, or Thursday from 8:00 in the morning to 12:30 in the afternoon. Note that no spaces are allowed within the field or between the *days* and *times* subfields.

*device* must be the keyword "ACU" if you are using a modem. If you are using a direct line to the other site, then you must give the file name of the serial line that you intend to use (e.g., **ttyc0**).

*phone* must be the telephone number of the given site. It must have the correct number of digits (including area code if necessary) or be a combination of **L-dialcodes** abbrevations and digits. **L-dialcode** abbreviations must go before any digits. Hyphens must not be used. For example, "5551234" is a valid local number and "2065551234" is a valid long-distance number. If the abbreviation "tw" is defined to be "555", then "tw1234" may be used in place of "5551234".

With the Hayes modem, you can use a comma in a number to cause a delay when dialing. This is useful if you must dial for an outside line before placing the call. For example, the number "9,5551234" causes a delay immediately after the "9" has been dialed. After the delay, the rest of the number is dialed. If you are not using a modem, then *phone* must be the file name of the device you intend to use instead of a phone number.

*login* must be a sequence of names, numbers, and other information that represents the steps required to log in to the given site. This sequence has the form

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

where *expect* is the prompt or message that you expect the given site to return to the calling site, and *send* is the name, number, or other information that you wish to send in response to the expected prompt or message. For example, the following is the login sequence for a typical XENIX site:

    login: uuchcgo ssword: market

Note that "ssword: " is given instead of the complete prompt "Password: ". Only the last eight characters in each expected prompt or message are examined, so you do not need to give the preceding characters.

You can include a conditional response immediately after each expected prompt or message. This conditional response has the form

    *expect* [ *-send-expect1* ]...

where *expect* is the prompt or message you expect the given site to return, *send* is the name or number you wish to send if the prompt or message returned is not correct, and *expect1* is the prompt or message you expect after sending the conditional response. For example, the following *login* field shows how to invoke the "login" prompt if it is not immediately present:

    --login-EOT-login-uuchcgo ssword: market

There are two special keywords that you can use in the login sequence. The "EOT" keyword causes an end-of-transmission character to be sent, and the "BREAK" keyword causes a break character to be sent. (The break character is simulated using line speed changes and null characters and may not work on all devices and/or systems. The break character may also cause the target system to change speed to 300 baud. For these reasons, you may want to avoid the use of BREAK.)

The complete **L.sys** entry must be placed on one line as shown by the following example:

    chicago Any ACU 1200 5551234 login uuchcgo ssword: market

**Create a Transmission Schedule**

In the uucp system, the **uucico** program carries out all transmissions between your site and other sites, sending and receiving files and commands as long as there is work for it to do. On a dial-in site, **uucico** is always started whenever a calling site logs in, but on a dial-out site, **uucico** is started only when an explicit invocation of the program is given. This means you must periodically invoke the program on a dial-out site to ensure that all transmissions requested by the **uucp** and **uux** programs are completed. You can do this in one of two ways: invoke the program manually whenever you need it, or create a shell script and let the **cron** program invoke **uucico** automatically according to a schedule of transmissions.

The most convenient method is to let **cron** invoke **uucico** for you, perhaps at a time when the local workload on the system is least. To do this, you must choose a schedule of times for **uucico** to be invoked, then create a **/usr/lib/crontab** file entry for the schedule. A **crontab** entry has the form

*minutes hour day month day-of-week command-line*

The entry gives the days or dates and times of day to execute the given *command-line*. Each field except *command-line* must be an integer within an acceptable range, e.g., 0 to 59 for *minutes*. Days of the week are designated 0 to 6 with Sunday designated by 0. A sequence of values for a field other than *command-line* can be given by separating the values with commas. Fields are separated by spaces. An asterisk can be given in a field other than *command-line* to indicate the set of all acceptable values. The *command-line* that you specify should be the name of the shell script that you have created to invoke **uucico.**

You can add an entry to **crontab** by using any of the text editors. For more information about the **cron** command and the **crontab** file, see the entry **cron** in the section "Commands" in the *XENIX 286 Reference Manual*. For example

        15, 45 * * * * /usr/lib/uucp/transmit

invokes the shell script **transmit** every 30 minutes to sites for which requests are pending. The entry

        0 0 * * * /usr/lib/uucp/transmit

invokes **transmit** every day at midnight, and the entry

        15 2,4,6 * * * /usr/lib/uucp/transmit

invokes the script every day at 2:15, 4:15, and 6:15 in the morning.

A shell script is simply a text file that contains one or more XENIX commands and that has execute permission. If your uucp system is acting as both a dial-in and dial-out site at different times, then the script should have the form

        disable /dev/tty*xy*
        uucico -r1 -s*site-name*

        .
        .
        .

        enable /dev/tty*xy*

where *xy* specifies your modem's serial line, **-r1** specifies MASTER mode, and *site-name* is the name of the site you wish to call. The **-s** option specifies a specific system to be called (instead of calling all systems); the system is only called if the local system has files or commands to send it. Also, the system is only called when times in the **L.sys** file permit it. The **-S** option may be used to force a call to the specified system regardless of whether the local system has work for it and regardless of the times specified in the **L.sys** file. Give one **uucico** command for each site you wish to call. If you want to call all sites for which requests exist, give a single **uucico** command, but do not give the **-s** or **-S** option with the command. If your computer is strictly a dial-out site, then the **enable** and **disable** commands are not required.

For example, the script

```
disable /dev/ttyc0
uucico -r1 -schicago
enable /dev/ttyc0
```

will place a call to the "chicago" site after disabling the serial line. The line must be disabled in order to dial out on that line. It must be enabled to allow subsequent calls from other computers to be received.

You can create a shell script by using a XENIX text editor. For convenience, the script should be placed in the **/usr/lib/uucp** directory and must be given execute permissions for everyone. Note that you can also add uucp maintenance programs to the script. See the section "Creating Maintenance Shell Files" later in this chapter.

## Linking Micnet Sites

You can send and receive mail from other Micnet sites through the uucp system by defining a uucp alias in the **maliases** file of each computer in your Micnet site. A uucp alias is any alias having the form

*site-name*!

where *site-name* is the name of a Micnet site. Uucp aliases can be used in **mail** commands to direct mail through the uucp system to the desired Micnet site.

To use a uucp system with your Micnet network, follow these steps:

● Add the entry

uucp:

to the **maliases** file of the computer on which the uucp system is installed.

● For all other computers in your site, add the entry

uucp:*machine-name*?

to the **maliases** file. *machine-name* must be the name of the computer on which the uucp system is installed.

You can test the uucp alias by mailing a short letter to yourself via another site. For example, if you are on the site "chicago", and there is another Micnet site named "seattle" in the system, then the command

mail seattle!chicago!johnd

will send mail to the "seattle" site, then back to your "chicago" site, and finally to the user "johnd" in your Micnet network. Note that a uucp system usually performs its communication tasks according to a fixed schedule and may not return mail immediately. If there are problems, check that the uucp installation is correct and that you have added the correct uucp aliases to all **maliases** files in your site. Also, make sure that the remote site has the correct uucp aliases in its **maliases** files.

# Maintaining the System

This section explains how to maintain the uucp system. In particular, it explains how to display and merge the content of uucp log files, how to remove old requests and files from the spool directories, and how to solve some common problems.

You can automate some maintenance tasks by creating shell command files and initiating these files with **crontab** entries. Other tasks require manual modification. Some sample shell files are given toward the end of this section.

## Displaying and Merging Log Files

You can display a record of the transmissions requested and completed to a given site or user by using the **uulog** command. The command displays the contents of the individual log files created for a given site or user and merges these entries with the system log file **LOGFILE**. The log files contain information about queued requests, calls to remote sites, execution of **uux** commands, and file copy results. The command has the form

        uulog [ -ssite-name ] [ -uuser ]

where –ssite-name names the site for which log files are to be displayed, and –uuser names the user whose log files are to be displayed. If you do not give a site-name or user, then log files for all sites or users, respectively, are displayed. The command places the new log files at the beginning of the existing **LOGFILE.**

The log files are originally created in the **/usr/spool/uucp** directory as individual files but should be copied to **LOGFILE** on a regular basis since they are not copied automatically. For example, the command

        uulog

merges all log files and displays their contents. The command

        uulog -schicago

merges only log files created for the site "chicago".

Note that each system's **LOGFILE** should be removed periodically, since it is copied each time new log files are added to it.

## Cleaning the Uucp Spool Directory

You can remove unwanted uucp system files from the uucp spool directory by using the uuclean command. The command removes temporary data, log, system status, and lock files from the spool directory if they are more than a given number of hours old. The command has the form

    uuclean -d*dir* -m -n*hours* -p*pre* -x*n*

where -d*dir* names the directory to be scanned, -m causes mail to be sent to the owner of each file removed, -n*hours* gives the age in hours of files to be removed, -p*pre* causes files with the given prefix to be examined and removed, and -x*n* directs the command to give the *n*th level of debugging output. Up to 10 file prefixes may be specified with the -p option. If -m is given, most mail will be sent to the owner of the uucp programs since most files put into the spool directory will be owned by the owner of the uucp programs. This is a result of the setuid bit being set on these programs. The default number of *hours* is 72 (3 days).

The uuclean program should be run once a day. You can invoke it automatically at a particular time each day by using the cron command. The command

    uuclean -pTM

removes all temporary data files that are at least three days old. The command

    uuclean -LCK -h1 -m

removes all lock files that are at least one hour old and mails a list of each file removed to the owner.

The uuclean command may also be run as needed to remove unwanted files after a system crash or an aborted uucp command.

## Reclaiming Log Files after a Crash

You can reclaim individual log files after a system crash by changing their access mode with the chmod command and then using the uulog command. After a transmission failure or system crash, the individual log file for the transmission may be left with access mode 0222, making it impossible for the uulog command to read the file. To reclaim the log file, you must use chmod to change the access mode to 0666. You can then let uulog merge it with LOGFILE.

## Reclaiming Data Files after a Crash

You can check the status of files transmitted from a remote site and possibly reclaim some or all of the data lost during an aborted transmission by examining system data files. The data files contain the contents of files copied from remote sites. These files are temporarily kept in the **/usr/spool/uucp** directory and their names have the form

TM.*pid.ddd*

where *pid* is a process ID and *ddd* is a sequential three-digit number starting at zero for each invocation of **uucico** and incremented for each file received.

The temporary data files are normally moved to the requested destination immediately after the transmission has finished. However, if a transmission has failed, the destination directory was not writable by uucp, or the system has crashed, the file remains in the spool directory. You can examine the contents of this file with the **cat** command. If desired, you can reclaim the file by moving it to a new location with the **mv** command. Leftover data files that cannot be reclaimed should be removed using the **uuclean** command.

## Checking the Transmission Status

You can check the status of transmissions between sites in the uucp system by examining the system status files. System status files contain information about login, dialup, or sequence check failure, as well as the talking status when two machines are conversing. The files are kept in the **/usr/spool/uucp** directory and their names have the form

STST.*site-name*

where *site-name* is the name of the remote site.

Normally, system status files are removed after each successful transmission, but when a failure occurs, the uucp system copies information about the failure to the file and leaves it in the directory. This prevents the uucp system from making further calls to the given site for about an hour, or for sequence check failures, until the file is removed.

To examine the status, use the **cat** command to display the contents of the file. If problems with transmissions are detected, it may indicate a problem with the modem or with the serial line connected to the modem.

If a system status file has been left due to a program or system crash, the file may prevent all subsequent transmissions to the given site. In this case, the file must be removed before attempting further calls.

## Checking for Locked Sites or Devices

You can make sure the uucp system is not intentionally preventing transmissions to a given site or through a given device by examining the system lock files. The uucp system creates a lock file for each site being called and for each device being used to call a site. Lock files prevent the uucp system from attempting to duplicate conversations with a given site, or from placing multiple calls on the same device. The lock files are kept in the **/usr/spool/uucp** directory and their names have the form

    LCK..*str*

where *str* is either a site name or the name of the calling device.

Since lock files prevent all calls to a given site or through a given device, it is wise to make sure no unnecessary lock files are left in the directory. If a transmission has been aborted or the system has crashed, the lock files will prevent subsequent transmissions for about 24 hours. If you want to place a call before this time, you must remove the file using the **uuclean** command.

## Creating Maintenance Shell Files

The **uulog** and **uuclean** command can be invoked automatically by placing the commands in a shell file and creating a **crontab** entry for the shell file. The system daemon **cron** will then invoke the commands at the given times, and most of the simple maintenance will be performed. For example, you can create a shell file that daily removes TM, ST, LCK, and C. or D. files for work that cannot be accomplished for reasons such as bad phone number and login changes. In this case, the shell file should contain the commands

    /usr/lib/uucp/uuclean -pTM -pC. -pD.
    /usr/lib/uucp/uuclean -pST -pLCK -n12

Note that the –n12 option causes the ST and LCK files older than 12 hours to be deleted. An appropriate **crontab** entry must be created to invoke the shell file automatically.

# Uucp Commands

This section describes four primary and two secondary uucp commands.  The primary commands are

**uucp**      This program creates work and gathers data files in the spool directory for the transmission of files.

**uux**       This program creates work and execute files, and gathers data files for the remote execution of XENIX commands.

**uucico**    This program executes the work files for data transmission.

**uuxqt**     This program executes XENIX commands found in execution files.

The secondary commands are

**uulog**     This program updates the log file with new entries and reports on the status of uucp requests.

**uuclean**   This program removes old files from the spool directory.

**dial**      This program directs the modem to dial a remote site.

Several other uucp commands are described only in the *XENIX 286 Reference Manual*: **cu**, **uuname** (in the **uucp** entry), **uustat**, **uusub**, **uuto**, and **uupick** (in the **uuto** entry).  **uucp** (includes **uulog**), **uux**, and **uuclean** are described in the *XENIX 286 Reference Manual* in addition to the descriptions in this chapter.  **uucico**, **uuxqt**, and **dial** are described only in this chapter.

Commands used by uucp users are **uucp**, **uux**, **uulog**, **uuname**, **uuto**, **uupick**, and **cu**.  Commands used only by the uucp system administrator are **uucico**, **uuxqt**, **uuclean**, **dial**, **uustat**, and **uusub**.  Of the system administrator commands, **uustat** and **uusub** can only be executed by the super-user.

**uucico** and **uuxqt** are normally executed periodically via **cron** as daemon programs that check for pending uucp work and do it.  Other commands are normally entered directly by the user or system administrator.

## Uucp Directories and Files

During execution of the uucp commands, the uucp system uses files from the following directories:

**/usr/lib/uucp**
          This is the directory used for uucp system files and all executable programs other than **uucp** and **uux**.

**/usr/spool/uucp**
>       This is the spool directory used during **uucp** execution.

**/usr/spool/uucp/.XFDIR**
>       This directory is used for failed **uux** commands.

**/usr/spool/uucp/.XQTDIR**
>       This directory is used during **uucp** execution of execute files.

Files are created in a spool directory for processing by the uucp daemons.   There are three types of files used for the execution of work:

Data files        Contain data for transfer to remote sites.

Work files        Contain directions for file transfers between sites.

Execute files     Contain directions for XENIX command executions that involve the resources of one or more sites.


## uucp--Site-to-Site File Copy

The **uucp** command is the user's primary interface with the uucp system.   **uucp**   is designed to be like the **cp** command, with the following syntax:

>       uucp [ *option* ]... *source... destination*

where *source* and *destination* may contain the prefix *site-name!*, which indicates the site on which the file or files reside or where they will be copied.

The options interpreted by **uucp** are

**-d**         Make directories when necessary for copying the file.

**-c**         Don't copy source files to the spool directory but use the specified source when the actual transfer takes place.

**-m**         Send mail on completion of the work.

The following options are used primarily for debugging:

**-s***dir*     Use directory *dir* for the spool directory.

**-x***num*    Use *num* as the level of debugging output.

The destination may be a directory name, in which case the file name is taken from the last part of the source name.   The source name may contain special shell characters such as "?*[ ]".   If a source argument has a *site-name!* prefix for a remote site, the file name expansion will be done on the remote site.

The command

        uucp  *.c  chicago!/usr/dan

will set up the transfer of all files with names ending with **.c** to the **/usr/dan** directory on the **chicago** machine.

The source and/or destination names may also contain a *~user* prefix. This translates to the login directory on the specified site. For names with partial path names, the current directory is prepended to the file name. File names with "../" are not permitted.

The command

        uucp  chicago!~dan/*.h  ~dan

will set up the transfer of files with names ending in **.h** in dan's login directory on site **chicago** to dan's local login directory.


### NOTE

> A single **uucp** command used to copy eight or more files may fail.
> To copy a large number of files easily, use the **tar** command to
> produce a single archive file that can then be transferred.


For each source file, the program will check the source and destination file names and the site part of each to classify the work into one of five types:

type 1.    Copy source to destination on local site.

type 2.    Receive files from other sites.

type 3.    Send files to remote sites.

type 4.    Send files from remote sites to another remote site.

type 5.    Receive files from remote sites when the source contains special shell characters as mentioned above.

After the work has been set up in the spool directory, the **uucico** program must be started to try to contact the other machine to execute the work (except for type 1 work).


### Copying Files to a Local Destination

A **cp** command is used to do type 1 work. The **-d** and the **-m** options are not honored in this case.

**Receiving Files from Other Sites**

For type 2 work, a one-line work file is created for each file requested and is put in the spool directory with the following fields, each separated by a blank:

1.   R

2.   The full path name of the source or *~username/pathname*. The *~username* part will be expanded on the remote site. This path name will be checked for accessibility in the remote **USERFILE** after **uucico** makes the connection.

3.   The full path name of the destination file. If the *~username* notation is used, it will be immediately expanded to be the login directory for the user.

4.   The user's login name.

5.   A "-" followed by an option list. (Only the -m and -d options will appear in this list.)

**Sending Files to Remote Sites**

For type 3 work, a work file is created for each source file and the source file is copied into a data file in the spool directory. (A -c option on the **uucp** program will prevent the data file from being made. In this case, the file will be transmitted from the indicated source.) Path names are checked using **USERFILE** to verify access to the requested directory. The fields of each entry are given below.

1.   S

2.   The full path name of the source file.

3.   The full path name of the destination or *~username/filename*.

4.   The user's login name.

5.   A "-" followed by an option list.

6.   The name of the data file in the spool directory.

7.   The file mode bits of the source file in octal print format (e.g., 0666).

**Copying Files between Sites**

For types 4 and 5 work, uucp generates a **uucp** command line and sends it to the remote machine; the remote **uucico** executes the command line.

## uux--Site-to-Site Execution

The **uux** command is used to set up the execution of a XENIX command where the execution machine and/or some of the files are remote. The syntax of the **uux** command is

        uux [ - ] [ *option* ]... *command-string*

where *command-string* is made up of one or more arguments. All special shell characters such as "<>|^" must be quoted either by quoting the entire command string, or by quoting the character as a separate argument. Within the command string, command and file names may contain a *site-name!* prefix. All arguments that do not contain a "!" will not be treated as files. (They will not be copied to the execution machine.) The - option is used to indicate that the standard input for the given command should be inherited from the standard input of the **uux** command. The only option is essentially for debugging: -*xnum* directs the command to use *num* as the level of debugging output.

The only commands that can be executed with **uux** are **rmail, rnews, uucp** and **uusend.** (Note that the **rnews** and **uusend** commands are not part of Intel's XENIX product, though they may be available on other UNIX-type systems that you access with uucp.) For example, the command

        pr abc | uux -chicago!rmail joe

pipes the output of "pr abc" as standard input to an **rmail** command to user **joe** to be executed at site **chicago.**

**uux** generates an execute file that contains the names of the files required for execution (including standard input), the user's login name, the destination of the standard output, and the command to be executed. This file is either put in the spool directory for local execution or sent to the remote site using a generated send command (type 3 above).

For required files that are not on the execution machine, **uux** will generate receive command files (type 2 above). These command files will be put on the execution machine and executed by the **uucico** program. (This will work only if the local site has permission to put files in the remote spool directory as controlled by the remote **USERFILE.**)

The execute file will be processed by the **uuxqt** program on the execution machine. It is made up of several lines, each of which contains an identification character and one or more arguments. The order of the lines in the file is not relevant and some of the lines may not be present. Each line is described below.

**User Line**

        U *user site*

where *user* and *site* are the requestor's login name and site.

### Required File Line

F *filename real-name*

where *filename* is the generated name of a file for the execute machine, and *real-name* is the last part of the actual file name (contains no path information). Zero or more of these lines may be present in the execute file. The **uuxqt** program will check for the existence of all required files before the command is executed.

### Standard Input Line

I *filename*

The standard input is either specified by a "<" in the *command-string* or inherited from the standard input of the **uux** command if the – option is used. If a standard input is not specified, **/dev/null** is used.

### Standard Output Line

O *filename site-name*

The standard output is specified by a ">" within the *command-string*. If a standard output is not specified, **/dev/null** is used. (Note that the use of ">>" is not implemented.)

### Command Line

C *command* [ *arguments* ]...

The *arguments* are those specified in the command string. The standard input and standard output will not appear on this line. All required files are moved to the execution directory (a subdirectory of the spool directory) and *command* is executed using the shell. In addition, a shell **PATH** statement is prepended to the command line as specified in the **uuxqt** program.

After execution, the standard output is copied or set up to be sent to the proper place.

### uucico--Copy In, Copy Out

The **uucico** command performs the following major functions:

- Scan the spool directory for work.

- Place a call to a remote site.

- Negotiate a line protocol to be used.

- Execute all requests from both sites.

- Log work requests and work completions.

**uucico** may be started by a system daemon, by the user (this is usually for testing), or by a remote site. (The **uucico** program should be specified as the shell field in the **/etc/passwd** file for the uucp logins.)

When started with the **-r1** option, the program is considered to be in MASTER mode. In this mode, a connection will be made to a remote site. If started by a remote site, the program is considered to be in SLAVE mode.

MASTER mode operates in one of two ways. If no site name is specified (the **-s** option not specified), then **uucico** scans the spool directory for sites to call. If a site name is specified, then that site will be called and work will only be done for that site.

The **uucico** program can be started directly by the user or by another program, such as a shell script invoked by **cron**. There are several options used for execution:

**-r1**      Start the program in MASTER mode. This is used when **uucico** is started by a program or **cron** shell.

**-s**_site-name_

Do work only for site _site-name_. If **-s** is specified, a call to the specified site will be made only if there is work for site _site-name_ in the spool directory and will only call when times in the **L.sys** file permit it.

**-S**_site-name_

Do work only for site _site-name_. If **-S** is specified, a call to the specified site will be made even if there is no work for the site in the spool directory. This is useful for polling sites that do not have the hardware to initiate a connection.Unlike **-s**, this option ignores the call times for the _site-name_ given in the **L.sys** file.

The following options are used primarily for debugging:

**-d**_dir_     Use directory _dir_ for the spool directory.

**-x**_num_    Use _num_ as the level of debugging output.

The next part of this section describes the major steps within the **uucico** program.

**Scanning for Work**

The names of the work-related files in the spool directory have the format

    *type . site-name grade number*

where *type* is "C" for copy command file, "D" for data file, or "X" for execute file; *site-name* is the remote site; *grade* is a character; and *number* is a four-digit, padded sequence number. For example, the file

    C.res45n0031

is a work file for a file transfer between the local machine and the "res45" machine.

The scan for work is done by looking through the spool directory for work files (files with prefix "C."). A list is made of all sites to be called. **uucico** calls the site specified by the –s or –S option and processes the corresponding work files.

**Calling a Remote Site**

The call is made using information from several files that reside in the uucp program directory. At the start of the call process, a lock is set to forbid multiple conversations between the same two sites. The lock file name has the form

    LCK..*str*

where *str* is the device name. The file is in the **/usr/spool/uucp** directory.

The site name is found in the **L.sys** file. The information contained for each site is

1.    Site name

2.    Times to call the site (days-of-week and times-of-day)

3.    Device or device type to be used for call

4.    Line speed

5.    Phone number if field 3 is "ACU" or the device name (same as field 3) if not

6.    Login information (multiple fields)

The time field is checked against the present time to see if the call should be made.

The phone number may contain abbreviations (e.g., mh, py, boston) that are translated into dial sequences using the **L-dialcodes** file.

The **L-devices** file is scanned using device type and line speed fields from the **L.sys** file to find an available device for the call. The program will try all devices that satisfy these fields until the call is made or until no more devices can be tried. If a device is successfully opened, a lock file is created so that another copy of **uucico** will not try to use it. If the call is complete, the login information in the last field of **L.sys** is used to log in.

The conversation between the two **uucico** programs begins with a handshake started by the SLAVE site. The SLAVE sends a message to let the MASTER know it is ready to receive the site identification and conversation sequence number. The response from the MASTER is verified by the SLAVE and, if acceptable, protocol selection begins. The SLAVE can also reply with a call-back-required message, in which case the current conversation is terminated.

### Selecting Line Protocol

The remote site sends a message

> P *proto-list*

where *proto-list* is a string of characters, each representing a line protocol.

The calling program checks the protocol list for a letter corresponding to an available line protocol and returns a use protocol message of the form

> U*code*

where *code* is either a one-character protocol letter or "N", which means there is no common protocol.

### Processing Work

The initial role of MASTER or SLAVE for the work processing is the mode in which each program starts. (The MASTER has been specified by the –r1 option.) The MASTER program does a work search similar to the one used in the section "Scanning for Work" above.

There are five messages used during the work processing, each specified by the first character of the message:

**S**     Send a file

**R**     Receive a file

**C**     Copy complete

**X**     Execute a **uucp** command

**H**     Hang up

The MASTER will send "R", "S", or "X" messages until all work from the spool directory is complete, at which point an "H" message is sent. The SLAVE will reply with the first letter of the request and either the letter "Y" or "N" for yes or no. For example, the message "SY" indicates that it is permitted to send a file.

The send and receive replies are based on permission to access the requested file or directory using the **USERFILE** and read/write permissions of the file or directory. After each file is copied into the spool directory of the receiving site, a copy-complete message is sent by the receiver of the file. The message "CY" will be sent if the file has successfully been moved from the temporary spool file to the actual destination. Otherwise, a "CN" message is sent. (In the case of "CN", the transferred file will be in the spool directory with a name beginning with "TM".) The requests and results are logged on both sites.

The hangup response is determined by the SLAVE program by a work scan of the spool directory. If work for the remote site exists in the SLAVE's spool directory, an "HN" message is sent and the programs switch roles. If no work exists, an "HY" response is sent.

### Terminating a Conversation

When an "HY" message is received by the MASTER, it is echoed back to the SLAVE and the protocols are turned off. Each program sends a final "OO" message to the other. The SLAVE program will clean up and terminate. The MASTER will proceed to call other sites and process work as long as possible or terminate if a -s or -S option was specified.

### uuxqt--Uucp Command Execution

The **uuxqt** program is used to process execute files generated by **uux**. The **uuxqt** program is started by the **uucico** program. The program scans the spool directory for execute files (prefix X.). Each one is checked to see if all the required files are available and if so, the command line or send line is executed.

The execute file is described in the section "**uux**--Site-to-Site Execution" above.

File execution is accomplished by executing the shell command

    sh -c

with the command line after appropriate standard input and standard output have been opened. If a standard output is specified, the program will create a send command or copy the output file as appropriate.

### Security

In unrestricted uucp systems, once a user logs in to another site through the uucp system, the user can execute any commands and copy any files normally accessible to the uucp login. It is up to the individual sites to be aware of this and apply the protections that they feel are necessary to prevent unauthorized use of files and commands.

The uucp system does provide a certain level of security. For example, a calling site does not get a standard shell when it logs in. Instead, the **uucico** program is started and all work is done through it. **uucico** checks the path names of files to be sent or received to prevent access to restricted directories. The **USERFILE** supplies the information for these checks. To prevent execution of possibly damaging commands, **uuxqt** can only execute the commands **rmail, rnews, uucp,** and **uusend** on a remote site. **rmail** is one of the underlying mail programs that helps deliver mail. (Note that the **rnews** and **uusend** commands are not part of Intel's XENIX product, though they may be available on other UNIX-type systems that you access with uucp.) Finally, the **L.sys** file is owned by uucp and has mode 0400 to protect the phone numbers and login information for remote sites.

## Troubleshooting

This section gives troubleshooting suggestions for specific uucp problems:

PROBLEM:        After invoking "uucp x y", you get the error message "permission denied x".

SOLUTION:      Make sure that x is readable by everyone. Make sure the entry on your own system's **USERFILE** allows you to access the file to be transferred.

PROBLEM:        You start **uucico** but nothing transfers and the message in the log file reads "NO CALL (RETRY TIME NOT REACHED)" or "CAN NOT CALL (SYSTEM STATUS)".

SOLUTION:      Go into the spool directory and remove the file **STST.**site-name, where site-name is the site you're attempting to call. Also remove any lock files (**LCK.***) present for that site or the serial line used.

PROBLEM:        You start **uucico** but the login to the remote system fails.

SOLUTION:      Check your hardware connections. If a direct line is being used, make sure that one side is DTE (Data Terminal Equipment) and the other side is DCE (Data Communications Equipment). If a modem is being used, test it as described in the section "Test the Connection" earlier in this chapter. Then make sure that the other system is running and also connected properly.

PROBLEM:            The file you're transferring makes it to **/usr/spool/uucp** on the remote
                    system, but does not make it to its final destination, and the log file
                    states that copy failed.

SOLUTION:           Make sure that the directory that you are sending to has write
                    permissions by "world" (all others outside the owner and group), so that
                    uucp can write to the directory.


PROBLEM:            On system a, you do a **uucp** command to send something to system b.
                    **uucico** is started on system b, but the file does not transfer.  The log
                    file contains the message "SEND PERMISSION (UUCP DENIED)".

SOLUTION:           On system b, make sure the **USERFILE** gives system a access to **/usr**,
                    because system a must be able to write in the spool directory.


# dial Program

The **dial** program is used by dial-out sites to place calls to other computers.  The **dial**
program initially supplied with XENIX 286 Release 3 is designed to work with the Hayes
Smartmodem 1200.  To use another modem with your uucp software, you may have to
create your own version of **dial**.  Source code for **dial** is not supplied with the initial
XENIX 286 Release 3 product, but it is listed in this section.  You can use this listing as
the basis for your own customized **dial** program.  Your source file should be **dial.c** and
can be compiled using the C compiler **cc**, as described in the *XENIX 286 Programmer's
Guide*.

```
/*
 *
 *      Copyright (C) Microsoft Corporation, 1983
 *
 * Simple dialer program for the Hayes "Smart" Modem 1200
 *
 *      See Hayes manual for command definitions
 *
 * Usage:   dial ttyname telnumber speed
 *
 *      returns 0 if a connection was made
 *             -1 otherwise
 */

#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <termio.h>
```

```
#define SAME    0
char *setup = "M1 F1 DT";          /* Speaker on, Full Duplex, Touch tone */
struct termio term;
int baudrate;                       /* baud rate of modem */
char buffer[80];
int alrmint( );

main(argc, argv)
int argc;
char *argv[ ];
{
    FILE *fdr, *fdw;
    int fd;

    if( argc != 4) {
        fprintf(stderr, "Usage:  dial devicename [number] speed\n");
        exit(-1);
    }
    if( (fd = open(argv[1], O_RDWR|O_NDELAY)) < 0 ) {
        fprintf(stderr, "dial: Can't open device: %s for reading.\n",
                    argv[1]);
        exit(-1);
    }

    switch(atoi(argv[3])) {
        case 300:
            baudrate = B300;
            break;
        case 1200:
            baudrate = B1200;
            break;
        default:
            baudrate = B1200;
    }
    /*
    * set line for no echo and specific speed
    */
    ioctl(fd, TCGETA, &term);
    term.c_cflag &= ~CBAUD;
    term.c_cflag |= CLOCAL|HUPCL|baudrate;
    term.c_lflag &= ~ECHO;
    term.c_cc[VMIN] = 1;
    term.c_cc[VTIME] = 0;
    ioctl(fd, TCSETA, &term);
    fcntl(fd, F_SETFL, fcntl(fd, F_GETFL, 0) & ~O_NDELAY);
```

```
        if( (fdr  =  fopen(argv[1], "r"))  = =  (char *)NULL) {
                fprintf(stderr, "dial: Can't open device: %s for reading.\n",
                        argv[1]);
                exit(-1);
        }
        if( (fdw  =  fopen(argv[1], "w"))  = =  (char *)NULL) {
                fprintf(stderr, "dial: Can't open device: %s for writing.\n",
                        argv[1]);
                exit(-1);
        }
        setbuf(fdw, 0);     /* Want unbuffered I/O */
        /*
         * setup for timeout in 10 seconds if no response
         */
        signal(SIGALRM, alrmint);
        alarm(10);
reread:
        fprintf(fdw, "AT\r");    /* Put Hayes into command mode */
        if(fgets(buffer, sizeof buffer, fdr)  = =  (char *)NULL)
                exit(-1);
        if(strncmp(buffer, "OK", 2) != SAME) { /* got back an OK? */
                sleep(1);
                goto reread;
        }
        alarm(0);    /* turn off alarm */
        sleep(1);
        fprintf(fdw, "AT %s %s\r", setup, argv[2]);/* put out dialing string */
        /*
         * turn off CLOCAL now, since we want modem interrupts to work
         * setup alarm.  (Longer timeout period for longer numbers)
         *
        ioctl(fd, TCGETA, &term);
        term.c_cflag &= ~CLOCAL;
        ioctl(fd, TCSETA, &term);
        */

        alarm((4*strlen(argv[2])) + 5);
again:
        if(fgets(buffer, sizeof buffer, fdr)  = =  (char *)NULL)
                exit(-1);
        if(strncmp(buffer, "NO CARRIER", 10)  = =  SAME) {
                exit(-1);
        }
        if(strncmp(buffer, "CONNECT", 7) != SAME) {
                goto again;
        }
        exit(0);
}

alrmint( )
{
        exit(-1);
}
```

A Micnet network supports communications between two or more independent XENIX systems. The network consists of computers connected by serial communication lines (that is, RS-232 ports connected by cable). Each computer in the network runs as an independent system, but users can communicate with the other computers in the network through the **mail, rcp,** and **remote** commands. These commands pass information such as mail, files, and even other commands from one computer to another.

If you want to construct a network using dial-up (modem) connections over phone lines, you must use uucp. (Uucp can also be used with direct connections between computers.) Installing uucp is described in Chapter 2.

It is the system administrator's task to build and maintain a Micnet network. The system administrator must decide how the computers are to be connected, make the actual physical connections, then use the **netutil** command to define and start the network. This chapter explains how to plan a network and then build it with the **netutil** command. In particular, it describes

- How to choose machine names and aliases

- How to draw the network topology map

- How to assign serial lines

- How to create the Micnet files

- How to distribute the Micnet files

- How to test the Micnet network

This chapter should be used in conjunction with the following command and file entries in the *XENIX 286 Reference Manual:*

- **aliashash, daemon.mn, netutil, rcp,** and **remote** entries in "Commands"

- **aliases, micnet, systemid,** and **top** entries in "Files"

## Planning a Network

To build a Micnet network, the **netutil** command requires that you provide the names of the computers that will be in the network, a description of how the computers will be connected, a list of the serial lines to be used, the names of the users who will use the network, and the aliases (if any) by which they will be known.

To keep the task as simple as possible, you should take some time to plan the network and make lists of the information you need to supply. To help you make these lists, the following sections suggest ways to plan a network.

## Choosing Machine Names

A Micnet network requires that each computer in the network have a unique "machine name." A machine name helps distinguish each computer from other computers in the network. It is best to choose machine names as the first step in planning the network. This step prevents confusion later on when you build the network with the **netutil** command.

A machine name should suggest the location of the computer or the people who are users on the computer; however, you may use any name you wish. The name must be unique, begin with a letter, and consist of letters and digits. The Micnet programs only use the first eight characters of each name, so be sure those characters are unique.

The **netutil** program saves the machine name of a computer in a **/etc/systemid** file. One file is created for each computer. After you have built and installed the network, you can find out the machine name of the computer you are using by displaying the contents of this file.

## Choosing a Network Topology

The network topology is a description of how the computers in the network are connected. In any Micnet network, there are two general topologies from which all topologies can be constructed. These are "star" and "serial".

In a star topology, all computers are directly connected to a central computer. All communications pass through the central computer to the desired destination.

In a serial topology, the computers form a chain, with each computer directly connected to no more than two others. All communications pass down the chain to the desired destination.

A network may be strictly star, strictly serial, or a combination of star and serial topologies. The only restriction is that no network may form a ring. For example, you cannot close up a serial network by connecting the two computers at each end.

The kind of topology you choose depends on the number of computers you have to connect, how quickly you want communications to proceed, and how you want to distribute the task of passing along communications. A star topology provides fast communication between computers but requires both a large portion of the central computer's total operation time and a large number of serial lines on the central computer. A serial topology distributes the communication burden evenly, requiring only two serial lines per computer but is slow if the chain is very long (communication between computers can take several minutes). Often a combination of star and serial topologies makes the best network. In any case, make the choice you think best. If you discover that another topology would be better, you may change the network at any time.

## Drawing a Network Topology Map

A network topology map is a sketch of the connections between computers in the network. You use the map to plan the number and location of the serial lines used to make the network.

You can make the map while you work out the topology. Simply arrange the machine names of each computer in the network on paper, then mark each pair of computers you wish to connect with serial lines. For example, the topology map for three computers might look like this:

```
a -------------- b -------------- c
```

As you draw, make sure that there is no more than one connection between any two computers in the network. Furthermore, make sure that no rings are formed (a ring is a series of connections that form a closed circle). Multiple connections and rings are not permitted.

## Assigning Lines and Speeds

Once you have made the topology map, you can decide which serial lines to use. Since every connection between computers in the network requires exactly two serial lines (one on each computer), you need to be careful about assigning the lines. Follow these steps:

1.  Make a list of the serial lines (tty lines) available for use on each computer in the network. You can display a list of the serial lines on a computer by displaying the file **/etc/ttys**. This file is described in the "Files" section of the *XENIX 286 Reference Manual*.

2.  Using the topology map, first pick a computer, then assign one serial line to each connection shown for that computer. The serial lines must be from the list of available lines for that computer. No line may be assigned more than once. For example, if computer "a" has only one available serial line (**ttya0**), then the topology map should look like this:

    ```
    a -------------- b -------------- c
    ttya0
    ```

3.  Repeat step 2 for all computers in the topology map. Make sure that each connection is assigned a line and that no two connections on any given computer have the same line. When finished, the map might look like this:

    ```
    a -------------- b -------------- c
    ttya0   ttya0 ttya1   ttya0
    ```

4.    Using the topology map, assign a serial line transmission speed for each computer
      pair. The speed may be any in the normal range for XENIX serial lines (i.e., 110,
      300...19200). Transmission speeds are a matter of preference. In general, a higher
      speed means a smaller amount of time to complete a transmission, but a greater
      demand on system input and output capabilities.

5.    After the topology map is completely filled in, make a list of all computer pairs,
      showing their machine names, serial lines, and transmission speeds. You will use
      this list when installing the network.

## Choosing Aliases

Once you have decided how to connect the computers in the network, you can choose
aliases for users in the network. An alias is a simple name that represents both a
location (computer) and an user. Aliases are used by the **mail** command to enable you to
refer to specific computers and users in a network without giving the explicit machine
and user names. Although not a required part of the network, aliases can make the
network easier to use and maintain.

There are three kinds of aliases: standard, machine, and forward. A standard alias is a
name for a single user or a group of users. A machine alias is a name for a computer or
an entire network (called a site). A forward alias is a temporary alias for a single user
or group of users. A forward alias enables users who normally receive network
communications at one computer to receive them at another.

When you build a network with the **netutil** program, you will be asked to provide
standard aliases only. (You can incorporate machine and forward aliases into the
network at your leisure.) Each standard alias must have a unique name and a list of the
login names of the users it represents. You may choose any name you wish as long as it
consists of letters and numbers, begins with a letter, and does not have the same
spelling as the login names. The name should suggest the user or group of users it
represents. The login names must be the valid login names of users in the network.

To help you prepare the aliases for entry during the **netutil** program, follow these steps:

1.    Make a list of the user aliases (i.e., the aliases that refer to just one user) and the
      login names of each corresponding user.

2.    Make a separate list of the group aliases (i.e., the aliases that refer to two or more
      users) and the login names or user aliases (from the first list) of the corresponding
      users. A group alias may have any number of corresponding users.

Note that there are a number of predefined group aliases. The name **all** is the
predefined alias for all users in the network. The machine names of the computers in
the network are predefined aliases for the users on each computer. Do not use these
names when defining your own aliases.

## Building a Network

You build a network with the **netutil** command.   **netutil** enables you to define the machines, users, and serial lines that make up the network.  The command produces the following menu:

```
Micnet network utility
        1        install new network configuration
        2        save the current network configuration to disk
        3        restore the network configuration from disk
        4        start the network
        5        stop the network

    enter desired option number or name:
```

To build a network, you must first create the Micnet files that define the network, then transfer these files to each computer in the network.  After each computer receives the files, you may start the network and use it to communicate between computers.

The following sections describe how to build the network.

### Creating the Micnet Files

The Micnet files are created with the install option of the **netutil** command.  The install option asks for the names, aliases, and serial lines of each computer in the network.  As you supply this information, **netutil** automatically creates the files needed for each computer.  These files can then be transferred to the other computers in the network with the save and restore options of **netutil.**  This means you can build the entire network from just one computer.

To use the install option, follow these steps:

1.   Log in as the super-user.

2.   Type

        netutil

    and press the RETURN key.  The program displays the network utility menu.  The install option is the first item in the menu.

3.   Type the number 1 and press the RETURN key.  The program displays the
     following message:

          Compiling new network topology
          Overwrite existing network files? (yes/no)?

     Type **y** and press the RETURN key to overwrite the files.  The existing network
     files must be overwritten to create the new network.  The first time you install
     the network, these files contain default information that need not be saved.  If you
     install the system a second time or expand the system, it may be wise to save a
     copy of these files before starting the install option.  The files can be saved on a
     flexible disk with the save option described later in this chapter.

     Once you have typed **y** the program displays the following message:

          Enter the name of each machine
               (or press RETURN to continue installation.)
          Machine name:

4.   Enter a machine name by typing the name and pressing the RETURN key.  You
     may enter more than one name on a line by separating each with a comma or a
     space.  After you have entered all the names, simply press the RETURN key to
     continue to the next step.  The program displays the names you have entered and
     asks if you wish to make changes.

5.   Type **y** (for "yes") if you wish to enter all the names again.  Otherwise, type **n** (for
     "no") or just press the RETURN key to move on to the next step.  If you type **n**,
     the program displays the message

          For each machine, enter the names of the machines
               to be connected with it
          Machine *name*:
           Connect to:

6.   Using the list of machine pairs you created when planning the network, enter the
     machine names of the computers connected to the given computer.  You may
     enter more than one name on a line by separating each name with a comma or a
     space.  When you have entered the machine names of all computers connected to
     the given computer, press the RETURN key.  The program asks for the names of
     the computers connected to the next computer.

7.   Repeat step 6 for all remaining computers.  As the program asks for each new set
     of connections, it will show a list of the machine names it already knows to be
     connected with the current computer.  You need not enter these names.  The
     program automatically checks for loops.  If it finds a loop, it ignores the machine
     name that creates the loop and asks for another.

     Finally, when you have given the connections for all computers in the network, the
     program displays a list of the connections and asks if you wish to make
     corrections.

8.    Type **y** if you wish to enter the connections again.  Otherwise, type **n** to move to the next step.  If you type **n**, the program displays the message

> For each machine pair, enter the tty name and tty speeds
> The following tty names are conventional for intel boards:
> iSBC 544    : ttya[0-f]          iSBC 534    : ttyb[0-f]
> iSBC 188/48 : tty[c-d][0-b]      iSBC 286/10 : ttyf0
> For the *name1* < = = > *name2* machine pair:
>    Tty on *name1*:

9.    Using the list of serial line assignments you created when planning the network, type the serial line name or suffix (last two characters) (e.g., **ttya3** or **a3**) for the first computer in the pair and press the RETURN key.  The program displays the message

> Tty on *name2*:

10.   Type the serial line name or suffix for the second computer in the pair and press the RETURN key.  The program displays the message

> Speed:

11.   Type the speed (e.g., 9600) and press the RETURN key.  The program asks for the serial lines and transmission speed of the next pair.

12.   Repeat steps 9 to 11 for all remaining machine pairs.  When you have given serial lines and speeds for all pairs, the program displays this information and asks if you wish to make corrections.

13.   Type **y** if you wish to enter the serial lines and speeds again.  Otherwise, type **n** to move to the next step.  The program displays the message

> Enter the names of users on each machine:
>
> For machine *name*:
>    Users on *name*:

14.   Enter a name by typing the login name of a user  on the given computer, then press the RETURN key.  The user names must be unique across the entire Micnet network, not just on each machine.  You may enter more than one name on a line by separating each name with a comma or a space.  When you have entered all names for the given computer, press the RETURN key.  The program displays the names of the users on the computer and asks if you wish to make corrections.

15.   Type **y** if you wish to enter the user names again.  Otherwise, type **n.** If you type **n,** the program asks for the users on the next computer.

16.   Repeat steps 14 and 15 for all remaining computers.  When you have given names of users for every computer, the program asks if you wish to enter aliases:

> Do you wish to enter any aliases? (yes/no)?

17. Type **y** if you wish to enter aliases. Otherwise, type **n** to complete the installation. If you type **y**, the program displays the message

> Each alias consists of two parts, the first is the alias name,
> the second is a list of one or more of the following:
>> valid user names
>> previously defined aliases
>> machine names

> Aliases:

18. Using the list of aliases you created when planning the network, type the name of an alias and press the RETURN key. The program displays the message

> Users/Aliases:

19. If the alias is to name a single user, type the login name of that user and press the RETURN key. The program asks for another alias.

    If, on the other hand, the alias is to name several users, type the login names of the users, or if one or more of the users to be named by the alias are already named by other aliases, type the aliases instead of the login names, or if all the users on one computer are to be named by the alias, type the machine name instead of the login names. In any case, make sure that each item typed on the line is separated from the next by a comma or a space. If there are more items than can fit on the line, type a comma after the last item on that line and press the RETURN key. You can then continue on the next line. After all names and aliases have been typed, press the RETURN key. The program asks for another alias.

20. Repeat steps 18 and 19 for all remaining user aliases in your list. When you have given all aliases, press the RETURN key. The program displays a list of all aliases and their users and asks if you wish to make corrections.

21. Type **y** if you wish to enter all aliases again. Otherwise, type **n** to complete the installation.

Once you direct **netutil** to complete the installation, it copies the information you have supplied to the network files, displaying the name of each file as it is updated. Once the files are updated, you may use the save option to copy the Micnet files to flexible disk.

## Saving the Micnet Files

You can save copies of the Micnet files on flexible disk with the save option of the **netutil** program. Saving the files enables you to transfer them to the other computers in the network. To save the files, follow these steps:

1.     Log in as the super-user.

2.     Type

          netutil

       and press the RETURN key. The command displays the network utility menu.

3.     Insert a blank, formatted flexible disk into your flexible disk drive and wait for the drive to accept the disk.

4.     Type the number **2** and press the RETURN key. The program responds:

          Save to:

       Enter the full path name ( **/dev/**xxx ) of the flexible disk drive being used and press the RETURN key. The program copies the Micnet files to the flexible disk.

5.     Remove the flexible disk from the drive. Using a soft-tipped marker (do not use ball point pen), label the disk "Micnet disk".

As soon as all files have been copied, you can transfer them to all computers in the network. Note that the Micnet files are saved on the flexible disk in **tar** format. The Micnet files can be saved to any other device or file in your XENIX system by specifying a different "Save to" path name.

## Restoring Micnet Files

The last step in building a Micnet network is to copy the Micnet files from the Micnet disk to all computers in the network. Do this with the restore option of the **netutil** program. For each computer in the network, follow these steps:

1.     Log in as the super-user.

2.     Insert the Micnet disk into your flexible disk drive and wait for the drive to accept the disk.

3.     Type

          netutil

       and press the RETURN key. The command displays the network utility menu.

4.    Type the number **3** and press the RETURN key.  The program responds:

Restore from:

Enter the full path name ( **/dev/xxx** ) of the flexible disk drive being used and press the RETURN key.  The program begins to copy the network files to the appropriate directories. It displays the name of each file as it is copied.  Finally, the program displays the message

Enter the name of this machine:

5.    Type the machine name of the computer you are at and press the RETURN key. The program copies this name to the new **/etc/systemid** file for the computer.  If necessary, it also disables the serial lines to be used on the computer, preparing them for use with the network.

When the files have been copied, you may start the network with the start option.


## Starting the Network

Once the Micnet files have been transferred to a computer, you can start the network with the start option of the **netutil** program.  The start option starts the Micnet programs that perform the tasks needed to communicate between the computers in the network.

To start the network, follow these steps for each computer in the network:

1.    Log in as the super-user.

2.    Type

netutil

and press the RETURN key.  The system displays the network utility menu.

3.    Type **4** and press the RETURN key.  The program searches for the **/etc/systemid** file.  If it finds the file, it starts the network.  If not, it asks you to enter the machine name of the computer and then creates the file.  The program also asks if you wish to log errors and transmissions.  In general, these are not required except when checking or testing the network.  When starting the network for the first time, type **n** to each question and press the RETURN key.

Once the network has started, you may move to the next computer and start the network there.

Note that, for convenience, you can let each computer start the network automatically whenever the system itself is started.  Simply include the command

/usr/lib/mail/daemon.mn

in the system initialization file **/etc/rc** of each computer.

## Testing a Micnet Network

After you have started a network for the first time, you should test the network to see that it is properly installed. In particular, you must determine whether or not each computer is connected to the network and communicating across the network.

To test the network, you will need to know how to use the **mail** command, described in "mail: The XENIX Mail System" in the *XENIX 286 User's Guide*. The following sections explain how to test the network and how to correct the network if problems are discovered.

### Checking the Network Connections

You can make sure that all computers are connected to the network by mailing a short message to **all** (the alias for all users in the network) with the **mail** command. Follow these steps:

1.    Choose a computer.

2.    Log in as the super-user.

3.    Use the **mail** command and the **all** alias to mail the message

          Micnet test

      to all users in the network.

4.    Check each user's mailbox in the network to see if the message was received. To check the mailboxes, log in as the super-user at each computer and use the **cat** command to display the contents of each user's mailbox. The name of each user's mailbox has the form

          /usr/spool/mail/*login-name*

      where *login-name* is the user's login name.

If all users receive the message, the network is properly installed. If the users at one or more computers fail to receive the message, then one or more computers are not properly connected to the network. To fix the problem, you need to locate the computer that has failed to make a connection. The next section explains how to do this.

## Using the LOG File to Locate a Problem

You can locate a problem with connections by examining the LOG files on each computer in the network. The LOG files contain a record of the interaction between each pair of computers. There are two LOG files for each pair of computers (one file on each computer). The LOG files on any given computer are kept in subdirectories of the **/usr/spool/micnet** directory. Each subdirectory has as its name the machine name of the other computer in the pair. You can examine the contents of a LOG file by typing

    cat  /usr/spool/micnet/*machine-name*/LOG

and pressing the RETURN key. The *machine-name* must be the name of a computer that is paired with the computer you are at.

Each LOG file should contain a "startup message" that lists the name of each computer in the pair and the serial line through which the pair is connected. It also shows the date and time at which the network was started. For example:

    daemon.mn:  running  as  MASTER
    Local system:  a
    Remote system:  b, /dev/tty02
    Tue  Sep  27  22:30:35  1983

A startup message is added to the file each time the network starts successfully. If the message is not present, then one or more of the network files and directories cannot be found. Make sure that you have used the restore option to transfer all the network files to the computer. Also, make sure that the **/etc/systemid** file contains the correct machine name for the given computer.

Each LOG file will contain a "handshake" message if the connection between the computer pair has been established. The message

    first  handshake  complete

is added to the file on a successful connection. If the message is not present, make sure that the network has been started on the other computer in the pair. The network must be started on both computers before any connection can be made. If the network is started on both computers yet no handshake message appears, then the serial line may be improperly connected or damaged. Check the serial line to make sure that the cable is firmly seated and attached to the correct RS-232 connectors on both computers. If necessary, replace the cable with one known to work.

If both the startup and handshake messages appear in the LOG file but the network is still not working, you can make the network log a record of the errors it encountered while transmitting and a record of each transmission by stopping and then restarting the network with the –e and –x switches.

The -e (for "errors") switch causes error messages generated by the program to be copied to the file. Each message lists the cause of the error and the subroutine that detected the error. For example, the message

    rsync: bad Probe resp: 68

shows that the **rsync** subroutine received a bad response (character 68 hexadecimal) from the other computer. You may use this information to track down the cause of the problem. One common problem is stray information being passed down the serial line by electronic noise. Make sure that the serial line's cable is properly protected against noise, e.g., make sure it does not lie near any electric motor, generator, or other source of electromagnetic radiation. Also, make sure the cable is in good condition.

The -x (for "transmissions") switch causes a record of normal transmissions between computers to be copied to the file. Each entry lists the direction, byte count, elasped time, and time of day of the transmission. For example, the entry

    rx: 0c 01 22:33:49

shows that 12 characters (0c hexadecimal) were received (**rx**) at **22:33:49.** The elapsed time for the transmission was 1 second. You can use the records to see if messages are actually being transmitted.

To start the network with the -e or -x switches, type **y** (for "yes") when the start option asks if you wish to log errors or transmissions.

## Stopping the Network

You can stop the network with the stop option of the **netutil** command. The option halts the Micnet programs and halts communication between computers in the network.

To stop the network, follow these steps on each computer in the network:

1.    Log in as the super-user.

2.    Type

       netutil

    and press the RETURN key. The command displays the network utility menu.

3.    Type **5** and press the RETURN key. The command stops the network programs running on the computer.

## Modifying the Micnet Network

You can modify a Micnet network at any time by changing one or more of the Micnet files. You can reinstall the network with the **netutil** program. For very small changes (for example, correcting the spelling of an alias), you can modify the Micnet files directly with a text editor. The files and their contents are described in the section "Files" in the *XENIX 286 Reference Manual*.

In general, you should copy a file before making any changes. You can make a copy with the **cp** command. Once one or more files have been changed on one computer, the files must be transferred to the other systems in the network using the save and restore options. These options can be used only after you have stopped the network.

Note that changes to the **aliases** file will not be incorporated into the system until the **aliashash** command is executed. This command produces the **aliases.hash** file needed by the network to resolve aliases.

## Using Uucp with Micnet

You can send and receive mail from other Micnet sites by installing a uucp system on one computer in your site and defining a uucp alias in the **maliases** file. A uucp system is a set of XENIX or UNIX programs that provide communication between computers using either ordinary phone lines or direct serial lines. A uucp alias is any alias having the form *sitename!* where *sitename* is the name of a Micnet site. Uucp aliases can be used in **mail** commands to direct mail through the uucp system to the desired Micnet site. To use a uucp system with your Micnet network, follow these steps:

1.    Install a uucp system on one computer in the Micnet site. Installation of a uucp system is described in Chapter 2.

2.    Add the entry

        uucp:

      to the **maliases** file of the computer on which the uucp system is installed.

3.    For all other computers in your site, add the entry

        uucp:*machine-name*?

      to the **maliases** file. *machine-name* must be the name of the computer on which the uucp system is installed.

You can test the uucp system by mailing a short letter to yourself via another site. For example, if you are on the site "chicago", and there is another Micnet site named "seattle" in the uucp system, then the command

        mail  seattle!chicago!johnd

will send mail to the "seattle" site, then back to your "chicago" site, and finally to the user "johnd" in your Micnet network. Note that a uucp system usually performs its communications tasks according to a fixed schedule and may not return mail immediately.

If there are problems, check that the uucp installation is correct and that you have added the correct uucp aliases to all **maliases** files in your site. Also make sure that the remote site has the correct uucp aliases in its **maliases** files.

Copies of the following publications can be ordered from

> Literature Department
> Intel Corporation
> 3065 Bowers Avenue
> Santa Clara, CA 95051

*Overview of the XENIX 286 Operating System*, Order Number 174385 -- XENIX history, XENIX uses, basic XENIX concepts, and an overview of other XENIX manuals.

*XENIX 286 Installation and Configuration Guide*, Order Number 174386 -- how to install XENIX on your hardware and tailor the XENIX configuration to your needs.

*XENIX 286 User's Guide*, Order Number 174387 -- a tutorial on the most-used parts of XENIX, including terminal conventions, the file system, the screen editor, and the shell.

*XENIX 286 Visual Shell User's Guide*, Order Number 174388 -- a XENIX command interface ("shell") that replaces the standard command syntax with a menu-driven command interpreter.

*XENIX 286 System Administrator's Guide*, Order Number 174389 -- how to perform system administrator tasks such as adding and removing users, backing up file systems, and troubleshooting system problems.

*XENIX 286 Communications Guide*, Order Number 174461 -- (this manual) installing, using, and administering XENIX networking software.

*XENIX 286 Reference Manual*, Order Number 174390 -- all commands in the XENIX 286 Basic System.

*XENIX 286 Programmer's Guide*, Order Number 174391 -- XENIX 286 Extended System commands used for developing and maintaining programs.
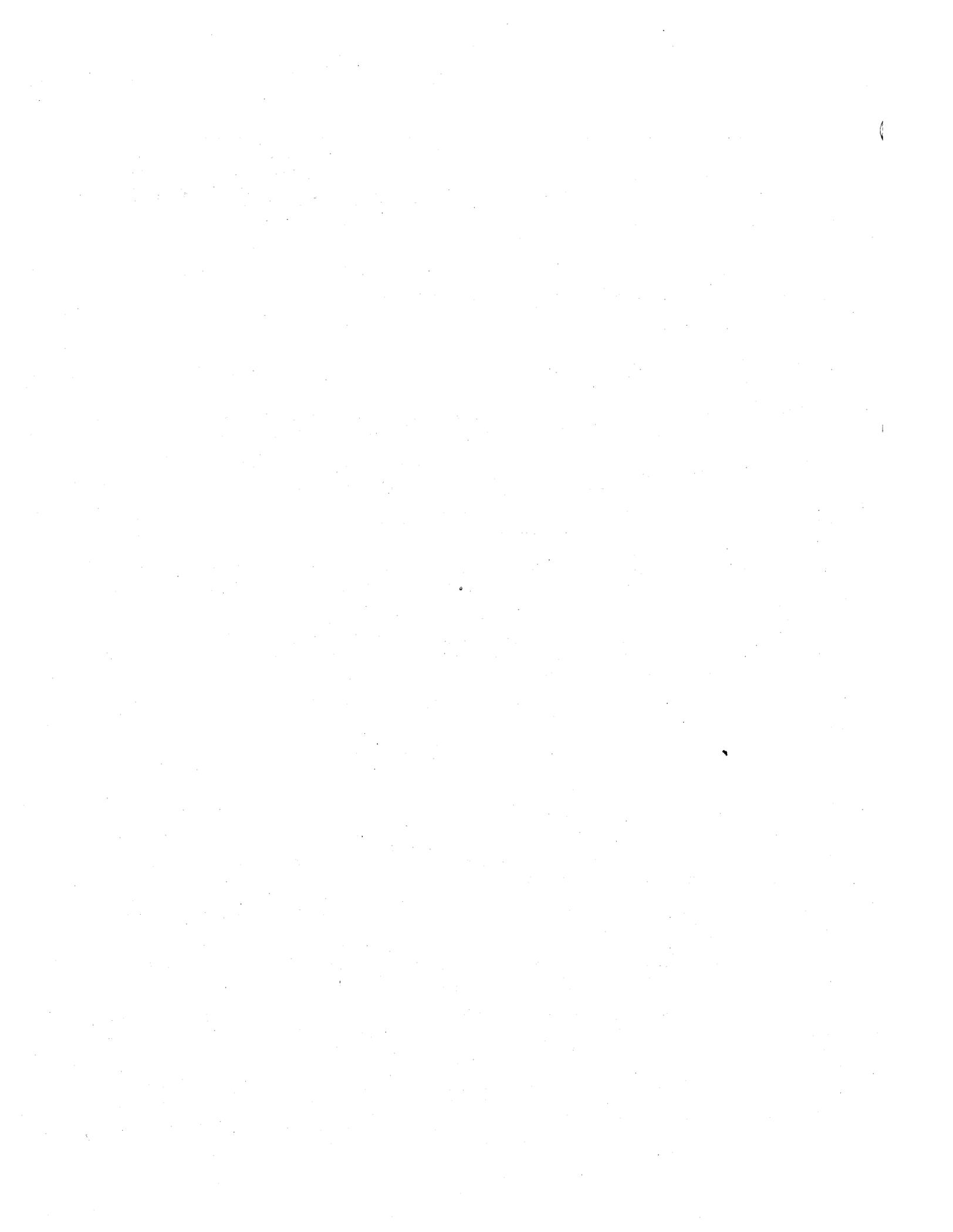
*XENIX 286 C Library Guide*, Order Number 174542 -- standard subroutines used in programming with XENIX 286, including all system calls.

*XENIX 286 Device Driver Guide*, Order Number 174393 -- how to write device drivers for XENIX 286 and add them to your system.

*XENIX 286 Text Formatting Guide*, Order Number 174541 -- XENIX 286 Extended System commands used for text processing and formatting.

C is described in *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie. One copy is supplied with Intel's XENIX product. Additional copies can be ordered from the publisher, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.

The Hayes Smartmodem 1200, used as an example in Chapter 2, "Uucp", is described in the *Smartmodem 1200 Owner's Manual*, available from Hayes Microcomputer Products, Inc., 5923 Peachtree Industrial Blvd., Norcross, GA 30092.

# REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

   _____

   _____

   _____

   _____

2. Does this publication cover the information you expected or required? Please make suggestions for improvement.

   _____

   _____

   _____

   _____

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

   _____

   _____

   _____

   _____

4. Did you have any difficulty understanding descriptions or wording? Where?

   _____

   _____

   _____

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). _____

NAME _____ DATE _____

TITLE _____

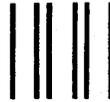COMPANY NAME/DEPARTMENT _____

ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____

        (COUNTRY)

Please check here if you require a written reply ☐

**WE'D LIKE YOUR COMMENTS ...**

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

# intel ®