Chapter 7

UUCP

UUCP stands for "UNIX to UNIX Copy Program," and is a set of utilities that lets computers using versions of the UNIX operating system (such as IRIX) communicate with each other and with remote terminals. These utilities range from those used to copy files between computers (*uucp* and *uuto*), to those used for simple encoding and decoding (*uuencode* and *uudecode*), to those used for remote login and command execution (*cu* and *uux*).

The following topics are covered in this chapter:

The UUCP system is contained in the *eoe2* subsystem of your IRIX distribution, in the package called *eoe2.sw.uucp*. Use the *versions* command to determine if you have this subsystem installed.

UUCP connections using telephone lines and modems are used to distribute electronic mail and "net news" among thousands of computers in the USENET network.

As an administrator, you need to be familiar with the administrative tools, logs, and database files used by UUCP. This chapter provides details about the UUCP files, directories, daemons, and commands.

Choosing TCP/IP or UUCP

This section compares UUCP and the TCP/IP protocol suite for various purposes. You can use them together, each for the tasks for which it is best suited. Both UUCP and TCP/IP software are standard features of the IRIX operating system. To use the TCP/IP software, you must have one of these communications mechanisms:

- a connection to an Ethernet network
- the optional FDDI hardware and software
- the Serial Line Internet Protocol (SLIP) software

To use UUCP, you must be connected to a serial network or to any TCP/IP network.

TCP/IP provides reliable interactive and batch services. UUCP is a batch-mode service; when you issue a *uucp* command, it is placed in a queue with other commands. The system checks the queue at regular intervals and executes the commands that it finds. After your command is carried out, UUCP reports the results of the command. The time it takes to carry out a command on a remote station varies on different stations. Table 7-1 shows a comparison of features of TCP/IP and UUCP.

Table 7-1 Comparison of TCP/IP and UUCP

TCP/IP Features	UUCP Features
runs on Ethernet and FDDI, and over serial lines	runs over serial lines or over TCP/IP links
transfers files interactively	transfers files in batch mode
executes commands on remote stations interactively	executes commands on remote stations in batch mode
sends mail interactively or in batch mode	sends mail in batch mode
starts a shell on a remote station	starts a shell on a remote station
provides remote login facilities with *rlogin/telnet*	provides remote login facilities with *cu*
transfers data to any station running TCP/IP	transfers data to any station running UUCP

Hardware Requirements for UUCP

Before your computer can communicate with other computers through UUCP, you must set up the hardware to complete the communications link. The cables and other hardware you will need depend on how you want to connect the computers: direct links, telephone lines, or local area networks.

Note: Refer to IRIX Admin: Peripheral Devices and the Personal System Administration Guide for information on setting up modems and other hardware.

Direct links
: You can create a direct link to another computer by running cables between serial ports on the two computers. Direct links are useful if two computers communicate regularly and are physically close—within 50 feet of each other. You can use a limited-distance modem to increase this distance somewhat. Transfer rates of up to 38,400 bits per second (bps) are possible when computers are directly linked. Such direct links are now rarely used because local area networks provide faster, easier-to-use connections.

Telephone lines
: Using a modem capable of dialing telephone numbers, your computer can communicate with other computers over standard phone lines. The modem dials the telephone number requested by the networking utilities. The computer it is trying to contact must have a modem capable of answering incoming calls.

UUCP Commands

UUCP programs can be divided into two categories: user programs and administrative programs. The subsections that follow describe the programs in each category.

UUCP User Programs

The UUCP user programs are in */usr/bin*. No special permission is needed to use these programs; they are all described in the reference pages.

cu
: Connects your computer to a remote computer so you can log in to that computer, allowing you to transfer some files or execute commands on either computer without dropping the initial link.

uucp — Lets you copy a file from one computer to another. This program creates work files and data files, queues the job for transfer, and calls the *uucico* daemon, which in turn attempts to contact the remote computer.

uuto — Copies files from one computer to a public spool directory on another computer (*/var/spool/uucppublic/receive*). Unlike *uucp*, which lets you copy a file to any accessible directory on the remote computer, *uuto* places the file in an appropriate spool directory and sends mail to the remote user who requested that it be picked up with *uupick*.

uupick — Retrieves the files placed under */var/spool/uucppublic/receive* when files are transferred to a computer that is using *uuto*.

uux — Creates the work, data, and execute files needed to execute commands on a remote computer. The work file contains the same information as work files created by *uucp* and *uuto.* The execute files contain the command string to be executed on the remote computer and a list of the data files. The data files are those files required for the command's execution.

uustat — Displays the status of requested transfers (*uucp*, *uuto*, or *uux*). This program also provides a way to control queued transfers.

UUCP Administrative Programs

Most of the administrative programs are in */usr/lib/uucp*, though the UUCP database files reside in */etc/uucp.* The only exception is *uulog*, which is in */usr/bin.* These commands are described in their respective reference pages.

You should use the *uucp* login ID when you administer UUCP because it owns the basic networking and spooled data files. The home directory of the *uucp* login ID is */usr/lib/uucp.* The other UUCP login ID is *nuucp*, used by remote computers that do not have their own login IDs to access your computer. A computer that logs in with *nuucp* receives *uucico* as its shell.

The following programs are the administrative utilities of UUCP:

uulog Displays the contents of a specified computer's log files. A log file is created for each remote computer with which your computer communicates. The log files contain records of each use of *uucp*, *uuto*, and *uux*.

uucleanup Cleans up the spool directory. This command is normally executed from a shell script called *uudemon.cleanup*, which is started by *cron*.

Uutry Tests call-processing capabilities and does a moderate amount of debugging. This command invokes the *uucico* daemon, in debug mode, to establish a communications link between your computer and the remote computer that you specify.

uucheck Checks for the presence of UUCP directories, programs, and support files. This program can also check certain parts of the *Permissions* file for obvious syntactic errors.

genperm Generates the *Permissions* file for stations that assign each remote station its own login ID.

The following programs are used for initializing different types of modems. The use of these programs is described in IRIX Admin: Peripheral Devices, Chapter 1, "Terminals and Modems."

fix-dsi This program initializes DSI modems

fix-hayes This program initializes Hayes modems.

fix-intel This program initializes Intel modems.

fix-telebit This program initializes Telebit modems.

fix-usr This program initializes U. S. Robotics modems.

fix-zyxel This program initializes Telebit modems.

UUCP Daemons

There are several daemons in UUCP. These daemons handle file transfers and command executions. They can also be run manually from the shell.

uucico — Selects the device used for the link, establishes the link to the remote computer, performs the required login sequence and permission checks, transfers data and execute files, logs results, and notifies the user by *mail* of transfer completions. It also starts *uuxqt* to execute any requested commands. When the local *uucico* daemon calls a remote computer, it "talks" to the *uucico* daemon on the remote computer during the session.

The *uucico* daemon is executed by the *uucp*, *uuto*, and *uux* programs, after all the required files have been created, to contact the remote computer. It is also executed by the *uusched* and *Uutry* programs.

uuxqt — Executes remote execution requests. This daemon searches the spool directory for execute files (always named *X.file*) that have been sent from a remote computer. When an *X.file* file is found, *uuxqt* opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. If the files are present and can be accessed, *uuxqt* checks the *Permissions* file to verify that it has permission to execute the requested command. The *uuxqt* daemon is executed by the *uudemon.hour* shell script, which is started by *cron*.

uusched — Schedules the queued work in the spool directory. Before starting the *uucico* daemon, *uusched* randomizes the order in which remote computers are called. *uusched* is executed by a shell script called *uudemon.hour*, which is started by *cron*.

uugetty — This program is very similar to the *getty* program except that it permits a line (port) to be used in both directions. *uugetty* is assigned to a port in the */etc/inittab* file if you want a port to be bi-directional. *uugetty* is described on the uugetty(1M) reference page.

Supporting Databases

The UUCP support files are in the */etc/uucp* directory.

Devices — Contains information concerning the location and line speed of the automatic call units (modems) and direct links.

Dialers	Contains character strings required to negotiate with automatic call units (ACUs) or modems in establishing connections to remote computers.
Systems	Contains information needed by the *uucico* daemon and the *cu* program to establish a link to a remote computer. This file contains information such as the name of the remote computer, the name of the connecting device associated with the remote computer, when the computer can be reached, the telephone number, the login ID, and the password.
Dialcodes	Contains dial-code abbreviations that can be used in the phone number field of Systems file entries.
Permissions	Defines the level of access that is granted to remote users using *uucp* or *uux* when they attempt to transfer files or remotely execute commands on your computer.
Poll	Defines computers that are to be polled by your station and when they are polled.
Sysfiles	Assigns different or multiple files to be used by *uucico* and *cu*, such as *Systems*, *Devices*, and *Dialers* files.

The subsections that follow provide details on the structure of these files so you can edit them.

There are several other files that can be considered part of the supporting database; these files are not directly related to the process of establishing a link and transferring files. The files, *Maxuuxqts*, *Maxuuscheds*, and *remote.unknown*, are described briefly in "Other UUCP Files" on page 151.

The Devices File

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

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

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

```
Type Line Line2 Class Dialer-Token-Pairs
```

Entries for use with modems should always have the form

```
Name device null speed 212 x dialer
```

Entries for use over TCP/IP network connections have the form

```
TCP - - Any TCP uucp
```

Devices file fields are defined in the following sections.

The Type Field

The keyword used in the Type field is matched against the third field of *Systems* file entries. The Type field can contain one of these keywords: Direct, ACU, or a station name.

Direct — This keyword indicates a direct link to another computer or a switch (for *cu* connections only).

ACU — This keyword indicates that the link to a remote computer is made through an automatic call unit (automatic-dial modem).

Sys-Name — This value indicates a direct link to a particular computer. (Sys-Name is replaced by the name of the computer.) This naming scheme is used to convey the fact that the line associated with this *Devices* entry is for a particular computer in the *Systems* file.

You can designate a protocol to use for a device within this field. See the "Protocols" section at the end of the description of this file.

The Line Field

This field contains the device name of the line (port) associated with the *Devices* entry. For instance, if the automatic dial modem for a particular entry is attached to the */dev/ttyf5* line, the name entered in this field is *ttyf5*.

You should always use the *ttyf* devices when working with modems. These devices support hardware flow control, which is used by all modems that support V.32 or V.32*bis*.

The Line2 Field

If the keyword ACU is used in the Type field and the ACU is an 801-type dialer, the Line2 field contains the device name of the 801 dialer. (801-type ACUs do not contain a modem. Therefore, a separate modem is required and must be connected to a different line, defined in the Line field.) The need for a separate modem line means that one line would be allocated to the modem and another to the dialer. Since non-801 dialers do not normally use this configuration, they ignore the Line2 field, but it must still contain a hyphen (-) or the word **null** as a place holder. A place holder is necessary for most modems.

The Class Field

The keyword used in the Class field of the Devices file is matched against the fourth field of Systems file entries:

```
Devices: ACU ttyf5 null D9600 212 x telebit
```

```
Systems: eagle Any ACU D9600 14155551212 login:nuucp password:Oakgrass
```

Some devices can be used at any speed, so the keyword ‘‘Any’’ can be used in the Class field. If Any is used, the line will match any speed requested in a Systems file entry. If this field is Any and the Systems file Class field is Any, the speed defaults to 9600 bps. If the keyword *ACU* or *Direct* is used in the Type field, the Class field might contain only the speed of the device. However, the speed can also be preceded by a letter (for example, *C9600*, *D9600*) to differentiate between classes of dialers (Centrex or Dimension PBX). Including the dialer class is necessary in larger offices that have more than one type of telephone network: One network may be dedicated to serving only internal communications while another handles external. In such a case, it becomes necessary to distinguish which line(s) should be used for internal and which for external.

The Dialer-Token-Pairs Field

This field contains pairs of dialers and tokens. The Dialer portion may be the name of an automatic-dial modem, or ‘‘Direct’’ for a direct-link device. You can have any number of Dialer-Token-Pair (DTP) fields. The Token portion may be supplied immediately following the Dialer portion; if not present, the Token portion will be taken from a related entry in the Systems file.

This field has the format:

```
dialer   token   dialer   token
```

The last pair may or may not be present, depending on the associated device (dialer). In most cases, the last pair contains only a *Dialer* portion and the *Token* portion is retrieved from the Phone field of the *Systems* file entry. A valid entry in the *Dialer* portion may be defined in the *Dialers* file.

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

If an automatic-dial modem is connected directly to a port on your computer, the *DTP* field of the associated *Devices* file entry will have only one pair. This pair is normally the name of the modem. This name is used to match the particular *Devices* file entry with an entry in the *Dialers* file. Therefore, the *Dialer* field must match the first field of a *Dialers* file entry:

```
Devices: ACU ttyf2 null 9600 212 x telebit

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

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

If a direct link is established to a particular computer, the *DTP* field of the associated entry contains the keyword *Direct*. This is true for both types of direct-link entries, *Direct* and *System-Name*.

If an automatic-dial modem is connected to a switch, your computer must first access the switch; the switch then makes the connection to the modem. This type of entry requires two *Dialer-Token-Pairs*. The *Dialer* portion of each pair (fifth and seventh fields of entry) is used to match entries in the *Dialers* file:

```
Devices: ACU ttyf2 null 9600 212 x t2500 telebit T25

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

Dialers: T25 =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
```

In the first pair, *t2500* is the *Dialer* and *telebit* is the *token* that is passed to the Develcon switch to tell it which device (telebit modem) to connect to your computer. This token is unique for each modem switch since each switch may be set up differently. Once the telebit modem has been connected, the second pair is accessed, where *T25* is the dialer and the token (the telephone number) is retrieved from the *Systems* file. (See the discussion of the Systems file's Phone field in "The Systems File" on page 138.)

Two escape characters can appear in a *DTP* field:

`\T` Indicates that the Phone (Token) field should be translated by means of the *Dialcodes* file.

`\D` Indicates that the Phone (Token) field should not be translated by means of the *Dialcodes* file. If no escape character is specified at the end of a *Devices* entry, the `\D` is assumed (default).

Device Protocols

You can define the protocol to use with each device. In most cases it is not needed since you can use the default or define the protocol with the particular station you are calling. (See the discussion of the *Systems* file, specifically the Type field.) If you do specify the protocol, you must do it in the form Type, Protocol. Available protocols are

g This protocol is slower and more reliable than **e**. It is good for transmission over noisy telephone lines. This is the default protocol.

e This protocol is faster than **g**, but it assumes error-free transmission, such as over a TCP/IP network connection.

t This protocol, like **e**, is for use in an error-free environment, such as a TCP/IP network connection. The **t** protocol is used by systems running BSD UNIX operating systems.

The Dialers File

The *Dialers* file (*/etc/uucp/Dialers*) specifies the initial conversation that must take place on a line before it can be made available for transferring data. This conversation is usually a sequence of ASCII strings that is transmitted and expected, and it is often used to dial a phone number with an ASCII dialer (such as an automatic-dial modem).

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

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

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

The *Dialer* field matches the fifth and additional odd-numbered fields in the *Devices* file.

The *substitutions* field is a translate string: The first of each pair of characters is mapped to the second character in the pair. This technique is usually used to translate the equal sign (=) and the hyphen (-) characters into whatever the dialer requires for "wait for dial tone" and "pause."

The *expect-send* fields are character strings.

The following list describes some of the escape characters used in the *Dialers* file. Table 7-2 lists escape sequences used by UUCP.

Table 7-2 UUCP Escape Sequences

Escape sequence	Meaning
\N	Send or expect a null character (ASCII NUL).
\b	Send or expect a backspace character.
\c	If at the end of a string, suppress the newline that is normally sent. Ignored otherwise.
\d	Delay two seconds before sending or reading more characters.
\p	Pause for approximately 1/4 to 1/2 second.
\E	Start echo checking. (From this point on, whenever a character is transmitted, login processing will wait for the character to be received before proceeding.)
\e	Turn off echo checking.
\n	Send a newline character.
\r	Send or expect a carriage return.
\s	Send or expect a space character.
\t	Send or expect a tab character.
\\\\	Send or expect a backslash (\\) character.
EOT	Send or expect EOT newline twice.
BREAK	Send or expect a break character.

Table 7-2 (continued) UUCP Escape Sequences

Escape sequence	Meaning
\K	Same as BREAK.
ddd	Send or expect the character represented by the octal digits *ddd*.

Here is a sample line from the *Dialers* file:

```
penril  =W-P    "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c :
\E\TP > 9\c OK
```

The *penril* entry in the *Dialers* file is executed as follows. First, the phone number argument is translated, replacing any equal sign with a **W** (wait for dial tone) and replacing any hyphen with a **P** (pause).

The handshake given by the remainder of the line works as follows:

""	Wait for nothing; in other words, proceed to the next step.
\d	Delay for two seconds.
>	Wait for a "greater than" sign (>).
Q\c	Send a Q with no terminating new line.
:	Wait for a colon (:).
\d-	Delay for two seconds.
>	Wait for a "greater than" sign (>).
s\p9\c	Send an s, pause for 1/2 second, send a 9, send no terminating new line.
)-W\p\r\ds\p9\c-)	Wait for a closing parenthesis [)]; if it is not received, process the string between the hyphens as follows: Send a W, pause, send a carriage return, delay, send an s, pause, send a 9 without a new line, and then wait for the closing parenthesis.
y\c	Send a y without a new line.
:	Wait for a colon (:)

\E\TP	Enable echo checking. (From this point on, whenever a character is transmitted, handshake processing waits for the character to be received before proceeding.) Then send the phone number. The `\T` instructs the program to take the phone number passed as an argument, apply the *Dialcodes* translation and the modem function translation specified by field two of this entry, then send a **P.**
9\c	Send a 9 without a new line.
OK	Wait for the string **OK.**

The Systems File

The *Systems* file (*/etc/uucp/Systems*) contains the information needed by the *uucico* daemon to establish a communications link to a remote computer. Each entry in the file represents a computer that can call or be called by your computer. In addition, UUCP software by default is configured to prevent any computer that does not appear in this file from logging in to your computer. (Refer to "Other UUCP Files" on page 151 for a description of the *remote.unknown* file.) More than one entry may be present for a particular computer. The additional entries represent alternative communications paths that are tried in sequential order.

Using the *Sysfiles* file, you can define several files to be used as *Systems* files. See "The Sysfiles File" on page 150 for details.

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

```
System-name Time Type Class Phone Login
```

These fields are defined in the following sections.

The System-name Field

This field contains the node name of the remote computer.

The Time Field

This field is a string that indicates the day-of-week and time-of-day when the remote computer can be called. The format of the Time field is

day[*time*][;*retry*]

The *day* portion is a list of one or more day specifiers. These specifiers are listed below.

Su, Mo, Tu, We, Th, Fr, Sa
: These abreviations stand for individual days of the week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday.

Wk
: Specifies any weekday, Monday through Friday.

Any
: Specifies any day.

Never
: Specifies that this station never initiates calls to the remote computer.If the Time field is **Never**, your computer will never initiate a call to the remote computer. The call must be initiated by the remote computer. In other words, your computer is in a passive mode with respect to the remote computer. (For more information on permissions, see "The Permissions File" on page 142.)

Here is an example of a Time field:

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

This example allows calls from 5:00 p.m. to 8:00 a.m., Monday through Friday, and any time Saturday and Sunday. The example would be an effective way to call only when phone rates are low, if immediate transfer is not critical.

The *time* portion should be a range of times such as 0800–1230. If no *time* portion is specified, any time of day is assumed to be allowed for the call. A time range that spans 0000 is permitted. For example, `0800-0600` means all times are allowed other than at times between 6 a.m. and 8 a.m.

An optional subfield, *retry*, is available to specify the minimum time (in minutes) before a retry, following a failed attempt. The default wait is 5 minutes after the first failure, 10 after the second, and so on until a delay of about 24 hours. If the retry subfield is present, that wait is used after every failure. The subfield separator is a semicolon (;). For example, "Any;9" is interpreted as "call any time, but wait at least 9 minutes before retrying after a failure occurs."

The Type Field

This field contains the device type that should be used to establish the communications link to the remote computer. The keyword used in this field is matched against the first field of *Devices* file entries:

```
Systems: eagle Any ACU,g D1200 3251 login:nuucp password: Oakgrass

Devices: ACU ttym2 - D1200 penril
```

You can define the protocol used to contact the station by adding it on to the Type field. The example just given shows how to attach the protocol **g** to the device type **ACU**. For direct connects, use the name of the station to which you are connecting. See “Device Protocols” on page 135.

The Class Field

This field is used to indicate the transfer speed of the device used to establish the communications link. It may contain a letter and speed (for example, *C1200*, *D1200*) to differentiate between classes of dialers. (See the discussion of the Class field in “The Devices File” on page 131.) Some devices can be used at any speed, so the keyword *Any* may be used. This field must match the Class field in the associated *Devices* file entry as shown here:

```
Systems: eagle Any ACU D1200 NY3251 login:nuucp password:Oakgrass

Devices:  ACU ttym2 - D1200 penril
```

If information is not required for this field, use a hyphen as a place holder for the field.

The Phone Field

This field is used to provide the phone number (token) of the remote computer for automatic dialers. The phone number is made up of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be one that is listed in the *Dialcodes* file. For example:

```
Systems: eagle Any ACU D1200 NY3251 login:nuucp password: Oakgrass

Dialcodes:  NY 9=1212555
```

In this string, an equal sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A hyphen (-) in the string instructs the ACU to pause four seconds before dialing the next digit.

If your computer is connected to a modem switch, you may access other computers that are connected to that switch. The *Systems* file entries for these computers does not have a phone number in the Phone field. Instead, this field contains the token that must be passed on to the switch so it will know which computer your computer wishes to communicate with. This token is usually just the station name. The associated *Devices* file entry should have a **\D** at the end of the entry to ensure that this field is not translated by means of the *Dialcodes* file.

The Login Field

This field contains login information given as a series of fields and subfields of the format:

expect send

The *send* string is sent when the *expect* string is received. The expect field may be made up of subfields of the form:

expect [*–send–expect*] **...**

The send field is sent if the prior expect is not successfully read and the expect following the send is the next expected string. For example, with "login--login," UUCP expects the string "login." If UUCP receives that string, it goes on to the next field. If it does not receive the string, it sends nothing, followed by a new line, then looks for "login" again. If no characters are initially expected from the remote computer, the characters **""** (null string) should be used in the first expect field. Note that all send fields are sent followed by a newline unless the send string is terminated with a **\c**.

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

```
owl Any ACU 1200 NY6013 "" \r login:-BREAK-login: uucpx word: xyzzy
```

This example means causes UUCP to send a carriage return and wait for the string "login:" If it doesn't receive the string, it sends a break character and waits for "login:" again. When it does receive the string "login," it sends the login name **uucpx**; waits to receive the string "word:" (the last part of "Password:"); and sends the password **xyzzy**.

Several escape sequences cause specific actions when they are a part of a string sent during the login sequence. These are the same escape sequences used in the *Dialers* file, listed in Table 7-2.

The Dialcodes File

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

```
abb dial-seq
```

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

For example, the following entry would work with a Phone field in the *Systems* file such as `jt7867`:

```
jt 9=555-
```

When the entry containing `jt7867` was encountered, the sequence `9=555-7867` would be sent to the dialer if the token in the dialer-token-pair was **\T**.

The Permissions File

The *Permissions* file (*/etc/uucp/Permissions*) specifies the permissions that remote computers have with respect to login, file access, and command execution. There are options that restrict the remote computer's ability to request files and its ability to receive files queued by the local site. Another option specifies the commands that a remote site can execute on the local computer.

The program */etc/uucp/genperm* is recommended for creating a sample or default *Permissions* file from the *Systems* file.

How Permissions File Entries Are Structured

Each entry is a logical line with physical lines terminated by a backslash (\) to indicate continuation. (Note that such continuations are not possible in most other UUCP files.) Entries are made up of options delimited by white space. Each option is a name/value pair in the following format:

name=value

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

Comment lines begin with a number sign (#) and occupy the entire line up to a newline character. Blank lines are ignored (even within multi-line entries).

There are two types of *Permissions* file entries:

LOGNAME — Specifies the permissions that take effect when a remote computer logs in to (calls) your computer.

MACHINE Specifies the permissions that take effect when your computer logs in to (calls) a remote computer.

LOGNAME entries begin with a LOGNAME option and MACHINE entries begin with a MACHINE option.

Permissions File Considerations

Keep these rules in mind when using the *Permissions* file to restrict the level of access granted to remote computers:

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

Permissions File Options

This section describes each option, specifies how it is used, and lists its default value.

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

The string that follows specifies that the remote computer can request to transfer files from your computer:

```
REQUEST=yes
```

The following string specifies that the remote computer cannot request to receive files from your computer:

```
REQUEST=no
```

This is the default value. It will be used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote calls you) entry or a MACHINE (you call remote) entry.

A note on security: When a remote computer calls you, you cannot verify its identity unless you have a unique login and password for that computer.

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

The string shown here specifies that your computer may send the work that is queued for the remote computer as long as it logged in as one of the names in the LOGNAME option:

```
SENDFILES=yes
```

This string is mandatory if your computer is in a "passive mode" with respect to the remote computer.

The string that follows specifies that files queued in your computer will be sent only when your computer calls the remote computer:

```
SENDFILES=call
```

The call value is the default for the SENDFILE option. This option is significant only in LOGNAME entries, because MACHINE entries apply when calls are made to remote computers. If the option is used with a MACHINE entry, it will be ignored.

READ and WRITE
These options specify the various parts of the filesystem that *uucico* can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

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

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

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

These strings specify permission to access any file that can be accessed by a local user with "other" permissions:

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

Because this suggestion may compromise security, use it only if required.

The value of these entries is a colon-separated list of pathnames. The READ option is for requesting files, and the WRITE option for depositing files. One of the values must be the prefix of any full pathname of a file coming in or going out. To grant permission to deposit files in */usr/news* as well as in the public directory, the following values would be used with the WRITE option:

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

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

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

NOREAD and NOWRITE

The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults. The strings shown here would permit one remote computer to read any file except those in the */etc* directory (and its subdirectories—remember, these are prefixes) and to write only to the default */var/spool/uucppublic* directory:

```
READ=/ NOR EAD=/etc WRITE=/var/spool/uucppublic
```

NOWRITE works in the same manner as the NOREAD option. NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

CALLBACK

The CALLBACK option is used in LOGNAME entries to specify that no transaction will take place until the calling station is called back. You would use CALLBACK for two reasons: From a security standpoint, if you call back a station you can be sure it is the station it says it is. If you are doing long data transmissions, you can choose the station that will be billed for the longer call.

The string that follows specifies that your computer must call the remote computer back before any file transfers will take place:

```
CALLBACK=yes
```

The default for the CALLBACK option is

```
CALLBACK=no
```

The CALLBACK option is very rarely used. Note that if two sites have this option set for each other, a conversation cannot be started.

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

The *uux* program generates remote execution requests and queues them to be transferred to the remote computer. Files and a command are sent to the target computer for remote execution.

The COMMANDS option can be used in MACHINE entries to specify the commands that a remote computer can execute on your computer. Note that COMMANDS is not used in a LOGNAME entry; COMMANDS in MACHINE entries defines command permissions, whether you call the remote station or it calls you.

This string indicates the default commands that a remote computer can execute on your computer:

```
COMMANDS=rmail
```

If a command string is used in a MACHINE entry, the default commands are overridden. For instance, in the following example, the entry overrides the COMMANDS default so that the computers *eagle*, *owl*, and *hawk* can now execute *rmail* and *rnews* on your computer:

```
MACHINE=eagle:owl:hawk REQUEST=yes
COMMANDS=rmail:/usr/bin/rnews
READ=/  WRITE=/
```

In addition to the names as specified above, there can be full pathnames of commands. For example, this line specifies that command *rmail* use the default path:

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

The default paths for your computer are */bin /usr/sbin*, */usr/bsd*, and */usr/bin*. When the remote computer specifies *rnews* or */usr/bin/rnews* for the command to be executed, */usr/bin/rnews* is executed, regardless of the default path. Likewise, */usr/local/lp* is the *lp* command that is executed.

Note: Including the ALL value in the list means that any command from the remote computer(s) specified in the entry is executed. If you use this value, you give the remote computer full access to your computer. *Be careful*. This value allows far more access than normal users have.

This string illustrates the greater access:

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

Two points about this string should be noted. The ALL value can appear anywhere in the string, and the pathnames specified for *rnews* and *lp* will be used (instead of the default) if the requested command does not contain the full pathnames for *rnews* or *lp*.

The VALIDATE option should be used with the COMMANDS option whenever potentially dangerous commands like *cat* and *uucp* are specified with the COMMANDS option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (*uuxqt*).

VALIDATE

The VALIDATE option is used in conjunction with the COMMANDS option when specifying commands that are potentially dangerous to your computer's security. It is used to provide a certain degree of verification of the caller's identity. The use of the VALIDATE option requires that privileged computers have a unique login and password for UUCP transactions. An important aspect of this validation is that the login and password associated with this entry be protected. If an outsider gets that information, that particular VALIDATE option can no longer be considered secure. (VALIDATE is merely an added level of security on top of the COMMANDS option, though it is a more secure way to open command access than ALL.)

Give careful consideration to providing a remote system with a privileged login and password for UUCP transactions. Giving another system these privileges is like giving anyone on that computer a normal login and password on your computer. Therefore, if you cannot trust everyone at the remote site, do not provide that system with a privileged login and password.

LOGNAME

The LOGNAME option ensures that remote stations attempting to log in to your computer have login privileges. The following LOGNAME entry specifies that if one of the remote computers that claims to be *eagle*, *owl*, or *hawk* logs in to your computer, it must have used the login *uucpfriend*:

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

As can be seen, if an outsider gets the uucpfriend login and password, marauding is trivial.

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

Each remote computer has its own "spool" directory on your computer. These spool directories have write permission given only to the UUCP programs. The execution files from the remote computer are put in its spool directory after being transferred to your computer. When the *uuxqt* daemon runs, it can use the spool directory name to find the MACHINE entry in the *Permissions* file and get the COMMANDS list or, if the computer name does not appear in the *Permissions* file, the default list is used.

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

```
MACHINE=eagle:owl:hawk REQUEST=yes  \
COMMANDS=rmail:/usr/bin/rnews \
READ=/  WRITE=/
LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/  WRITE=/
```

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

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

MACHINE Entry for "Other" Systems

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

```
MACHINE=OTHER \
   COMMANDS=rmail:rnews:/usr/bin/Photo:/usr/bin/xp
```

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

Combining MACHINE and LOGNAME Entries

It is possible to combine MACHINE and LOGNAME entries into a single entry where the common options are the same. For example, the two entries that follow share the same REQUEST, READ, and WRITE options:

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

These two entries can be merged:

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

MYNAME

The MYNAME option is used to override the name of the local computer, when the local computer identifies itself to the remote computer. This facility is useful when a computer is replaced or renamed, and its neighbors need to process old traffic to the old name.

The Poll File

The *Poll* file (*/etc/uucp/Poll*) contains information for polling remote computers. Each entry in the *Poll* file contains the name of a remote computer to call, followed by a **<Tab>** character (a space won't work), and finally the hours at which the computer should be called. The format of entries in the *Poll* file is

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

For example, the following entry provides polling of computer *eagle* every four hours:

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

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

The Sysfiles File

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

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

The format of the *Sysfiles* file is:

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

The *w* parameter is replaced by *uucico*, *cu*, or both separated by a colon; *x* is one or more files to be used as the *Systems* file, with each filename separated by a colon and read in the order presented; *y* is one or more files to be used as the *Dialers* file; and *z* is one or more files to be used as the *Devices* file. Each file is assumed to be relative to the */etc/uucp* directory, unless a full path is given. A backslash-carriage return (**\<Return>**) can be used to continue an entry to the next line.

Here is an example using a local *Systems* file in addition to the usual *Systems* file:

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

If this line is in */etc/uucp/Sysfiles*, then both *uucico* and *cu* will first look in */etc/uucp/Systems*. If the station they're trying to call doesn't have an entry in that file, or if the entries in the file fail, then they'll look in */etc/uucp/Local_Systems*.

When different *Systems* files are defined for *uucico* and *cu* services, your station will store two different lists of stations. You can print the *uucico* list by using the *uuname* command, or the *cu* list by using the *uuname* ***-c*** command.

Other UUCP Files

Three files, in addition to those described in the preceding subsections, have an impact on the use of basic networking facilities. In most cases, the default values are fine and no changes are needed. If you want to change the default values, however, use any standard IRIX text editor (*ed*, *vi*, or *jot*).

Maxuuxqts — This file defines the maximum number of *uuxqt* programs that can run at once. The default number is two.

Maxuuscheds — This file defines the maximum number of *uusched* programs that can run at once. The default number is two.

unknown — This file is a program that executes when a station that is not in any of the *Systems* files starts a conversation. The program logs the conversation attempt and refuses the connection. If you change the permissions of this file so it cannot execute (*chmod 000 unknown*), your station will accept any conversation requests.

UUCP Administrative Files

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

TM (temporary data file)

These data files are created by UUCP processes under the spool directory (for example, */var/spool/uucp/X*) when a file is received from another computer. The directory X has the same name as the remote computer that is sending the file. The names of the temporary data files have the following format:

```
TM.pid.ddd
```

pid is a process-ID and *ddd* is a sequential, three-digit number starting at 0.

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

LCK (lock file) — Lock files are created in the */var/spool/locks* directory for each device in use. Lock files prevent duplicate conversations and multiple attempts to use the same calling device. The names of lock files have this format:

`LCK..`*`str`*

str is either a device or computer name. These files may remain in the spool directory if the communications link is unexpectedly dropped (usually because of a computer crash). Lock files are ignored (removed) after the parent process is no longer active. Each lock file contains the process ID of the process that created the lock.

C. (work file) — Work files are created in a spool directory when work (file transfers or remote command executions) has been queued for a remote computer. The names of work files have the following format:

`C.`*sysnxxxx*

sys is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the four-digit job sequence number assigned by UUCP. Work files contain the following information:

- full pathname of the file to be sent or requested
- full pathname of the destination or user or filename
- user login name
- list of options
- name of associated data file in the spool directory. If the *uucp* ***-c*** or *uuto* ***-p*** option was specified, a dummy name (**D.0**) is used
- mode bits of the source file
- login name of the remote user to be notified upon completion of the transfer

D. (data file) — Data files are created when the command line specifies that the source file should be copied to the spool directory. The names of data files have the following format:

`D.`*systmxxxxyyy*

systm is the first five characters in the name of the remote computer and *xxxx* is a four-digit job sequence number assigned by UUCP. The four-digit job sequence number may be followed by a subsequence number, *yyy*, used when several *D.* files are created for a work (*C.*) file.

X. (execute file)

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

`X.`*sysnxxxx*

sys is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is a four-digit sequence number assigned by UUCP. Execute files contain the following information:

- requester's login and computer name
- name of file(s) required for execution
- input to be used as the standard input to the command string
- filename for a file to receive standard output from the command execution, together with the hostname of the computer on which the file resides
- command string
- option lines for return status requests

Setting Up UUCP

Setting up UUCP involves five steps:

1. Determine the *remote* and *local* stations.
2. Make the physical connection
3. Configure the *local* (calling) station
4. Configure the *remote* (called) station
5. Test the UUCP connection

Determining the Remote and Local Stations

Typically, the *local* station is the station that initiates the UUCP connection. The *remote* station is the station that responds to UUCP connection requests. However, with the arrival of *uugetty* (a program that allows bi-directional line usage), the distinction between the local and remote station is usually only the station name.

For our example, *japan* is the local station and *us* is the remote station.

Making the Physical Connection

UUCP supports physical connections for TCP/IP local area network connections, direct links, or telephone lines. This example assumes a direct link. The procedure for running UUCP over telephone line or local area networks is similar, requiring minor adjustments to the various configuration files.

A direct link constitutes a connection between two Data Terminal Equipment (DTE) devices. The devices must be fooled into thinking they are communicating with a Data Communication Equipment (DCE) device. The way to get around this is with a null modem.

The minimum pinning configuration shown in Table 7-3.

Table 7-3 Three-Wire Null-Modem Pinning Configuration

IRIS A	IRIS B
2 Transmit Data	3 Receive Data
3 Receive Data	2 Transmit Data
7 Signal Ground	7 Signal Ground

Attach the null modem cable to serial port two (*ttyf2*) on the local and remote workstations.

Note: For more information on cables and modems, see IRIX Admin: Peripheral Devices, Chapter 1, "Terminals and Modems."

Configuring the Local Station

The remote station name in our example is *us*, and the local station name is *japan*. There are two steps in configuring the local station:

1. Update standard system files.
2. Modify the UUCP configuration files.

Updating Standard System Files

The three system files that you need to be concerned with are

- */etc/passwd*
- */etc/group*
- */etc/inittab*

/etc/passwd

To ensure proper security and access, you need to ensure that the user entries for *uucp* and *nuucp* are both present and correct. The *uucp* entry in the *passwd* file is for ownership purposes, and the *nuucp* entry is for remote UUCP access. Ensure that your password file has both entries and that they are the same as the following example. If the *uucp* and *nuucp* entries don’t match the following, edit those accounts so they do match.

```
uucp:*:3:5:UUCP Owner:/usr/lib/uucp:/bin/csh

nuucp::10:10:Remote UUCP
User:/var/spool/uucppublic:/usr/lib/uucp/uucico
```

In the above example, the *passwd* entry for *nuucp* is split across two lines due to formatting constraints. In the actual file, the entry appears on a single line.

On a newly installed station, neither *uucp* nor *nuucp* has a password. It is a good idea to put a “*” in the password field for uucp, since no one should log in as *uucp*. You need to assign *nuucp* a valid password that matches the password you assign for *nuucp* in the Systems file. (See “The Systems File” on page 138. For example, assign *nuucp* the password “secret.”)

```
New password: secret
Re-enter new password: secret
```

/etc/group

Check this file to ensure that there are valid groups for both *uucp* and *nuucp*. Compare your uucp and nuucp group entries with the following. If there is a discrepancy, correct it now.

```
uucp::5:uucp
nuucp::10:nuucp
```

/etc/inittab

This sample entry is for the local station. It allows calls to be initiated on port two, but does not allow incoming calls on the port. Edit your */etc/inittab* entry for "t2" as follows:

```
t2:23:off:/usr/lib/uucp/uugetty -Nt 60 ttyf2 co_9600 # port 2
```

For complete information on the *uugetty* command, see the uugetty(1M) reference page. As usual, any time you make a change to the */etc/inittab*, you must tell init to read the file again with the *telinit q* command. Issue the following command:

```
/etc/telinit q
```

Modifying the UUCP Configuration Files

The UUCP configuration files to be modified are:

- */etc/uucp/Systems*
- */etc/uucp/Devices*
- */etc/uucp/Dialers*
- */etc/uucp/Permissions*

/etc/uucp/Systems

The *Systems* file contains information describing the station(s) that the local station knows about. Add the following line to the bottom of the file:

```
us Any systemx 9600 unused ogin:--ogin: nuucp ssword: \ secret
```

Note: The *Systems* file is read only, so if you are using *vi* you must force this change to be written out by exiting *vi* with the **:wq!** option.

The first field specifies the name of the station that can call (the remote station). The second field indicates that the specified station can call at any time. The third field tells *uucp* the name of the device to use (**systemx**). The third field must match one of the first field entries found in */etc/uucp/Devices*. The forth field specifies the transfer speed (9600). The fifth field is normally used for a phone number, but unused for direct links. The rest of the line handles the login sequence; it is the chat script negotiated between the local station and the remote station. This chat script is very important for a successful *uucp* connection.

/etc/uucp/Devices

The *Devices* file contains information about the physical connection between the two stations. Remove the pound sign from the *systemx* device entry so it looks like the following:

```
# ---A direct connection to a system
systemx ttyf2 - Any direct
```

Note: If you have another direct connection to a station on another port, copy the *systemx* device entry and modify the port number accordingly.

The first field in the *Devices* file links the device to the *Systems* file (third field entry). The second field tells *uucp* which port to access. The third field is used with an Automatic Call Unit (ACU). Direct links use a dash in the third field. The fourth field specifies the line speed. The "Any" entry allows the speed to be determined by the */etc/inittab* file for that particular device. The fifth field contains the dialer name. It must be a valid entry in the */etc/uucp/Dialers* file.

/etc/uucp/Dialers

This file contains the chat script for the *uucp* device. Because this is a direct connection, the chat script is picked up from the *Systems* file. However, there still has to be a valid dialers entry for the direct connection. Verify that the *Dialers* file has an entry for the "direct" dialer. Enter the following command:

```
grep direct /etc/uucp/Dialers
```

The system responds with

```
direct
#   The following entry is for use with direct connections
uudirect ""   "" \r\d in:--in:
```

/etc/uucp/Permissions

The *Permissions* file controls remote *uucp* access with regard to remote users and stations. See “The Permissions File” on page 142 for descriptions of all options. For this example, edit the Permissions file to look like the following:

```
#dent"@(#)uucp:Permissions2.2"
# This entry for public login.
# It provides the default permissions.
# See the Basic Networking Utilities Guide for more information.
LOGNAME=nuucp MACHINE=us READ=/var/spool/uucp/uucppublic \
WRITE=/var/spool/uucppublic REQUEST=yes SENDFILES=yes \
COMMANDS=rmail
```

Note: This entry must be interpreted as a single line, even if it expands more than one physical line.

This entry specifies that the user, *nuucp*, is allowed to log in from the remote station (*us*). The *nuucp* user on *us* may read any files that reside in */var/spool/uucp/uucppublic* directory and write to the general public directory */var/spool/uucppublic*. The users on *us* may make requests. Users on *japan* can send files.

Configuring the Remote Station

The remote station name in our example is *japan*. The local station name in our example is *us*. There are two steps in configuring the remote station:

1. Update standard system files.
2. Modify the UUCP configuration files.

Updating Standard System Files

The three system files that you need to be concerned with are

- */etc/passwd*
- */etc/group*
- */etc/inittab*

/etc/passwd

To ensure proper security and access, you need to ensure that the user entries for *uucp* and *nuucp* are both present and correct. The *uucp* entry in the passwd file is for ownership purposes and the *nuucp* entry is for remote UUCP access. Ensure that your password file has both entries and that they are the same as the following example. If the *uucp* and *nuucp* entries don't match the following, edit them so they do match.

```
uucp:*:3:5:UUCP Owner:/usr/lib/uucp:/bin/csh
nuucp::10:10:Remote UUCP User:/var/spool/uucppublic:/usr/lib/uucp/uucico
```

On a newly installed station, neither *uucp* nor *nuucp* has a password. It is a good idea to put an asterisk (*) in the password field for *uucp*, since no one should log in as *uucp*. You need to assign *nuucp* a valid password that matches the password you assign for *nuucp* in the Systems file. (See "The Systems File" on page 138.) Assign *nuucp* the password "secret" with the command:

`passwd nuucp`

`New password:` **`secret`**

`Re-enter new password:` **`secret`**

/etc/group

Check this file to ensure that there are valid groups for both *uucp* and *nuucp*. Compare your *uucp* and *nuucp* group entries with the following example. If there is a discrepancy, correct it now.

```
uucp::5:uucp
nuucp::10:nuucp
```

/etc/inittab

This sample entry is for the remote station. It allows calls to be received on serial port 2, but does not allow outgoing calls on the port. Edit your */etc/inittab* entry for "t2" as follows:

```
t2:23:respawn:/usr/lib/uucp/uugetty -Nt 60 ttyf2 co_9600#pt 2
```

For complete information on the *uugetty* command, see the uugetty(1M) reference page. As usual, any time you make a change to the */etc/inittab*, you must use the *telinit q* command to tell *init* to read the file again. Issue the following command:

`/etc/telinit q`

Modifying the UUCP Configuration Files

The UUCP configuration files to be modified are

- */etc/uucp/Systems*
- */etc/uucp/Permissions*

/etc/uucp/Systems

The *Systems* file contains information describing the station(s) that the remote station knows about. Add this line to the bottom of the *Systems* file:

```
japan Never
```

Note: The permission of the *Systems* file is read-only. If you edit this file, you may have to use a forced write in order to save your changes. Consult the *jot* or *vi* reference page for more information.

The first field specifies the name of a station that can call the local station. The second field indicates the times this station can make the call. The value **Never** in this field indicates that this station that this station may receive calls, but never initiate them.

/etc/uucp/Permissions

The *Permissions* file controls remote *uucp* access with regard to remote users and stations. See "The Permissions File" on page 142 for descriptions on all options. For this example, edit the *Permissions* file to look like the following:

```
#ident"@(#)uucp:Permissions2.2"
# This entry for public login.
# It provides the default permissions.
# See the Basic Networking Utilities Guide for more
                                                 information.
LOGNAME=nuucp MACHINE=japan READ=/var/spool/uucp/uucppublic\
WRITE=/var/spool/uucppublic REQUEST=yes SENDFILES=yes \
COMMANDS=rmail
```

Note: This entry must be interpreted as a single line, even if it expands to more than one physical line.

This entry specifies that the user, *nuucp*, is allowed to log in from the local station (*japan*). The *nuucp* user on *japan* may read any files that reside in */var/spool/uucp/uucppublic* directory and write to the general public directory */var/spool/uucppublic*. The users on *japan* may request files. The users on *us* can send files.

Setting Up UUCP on a TCP/IP Connection

In many cases, you may decide to use the UUCP tools over conventional TCP/IP network connections. There are entries in the *Devices* file provided for this, but you must make some changes in the */usr/etc/inetd.conf* and */etc/uucp/Systems* files. Follow these steps:

1. Edit the */usr/etc/inetd.conf* file on the remote host and find this line:

   ```
   #uucp stream tcp nowait root /usr/lib/uucp/uucpd uucpd
   ```

   Remove the leading hashmark (#) to uncomment the line, and use these commands to make the change take effect:

   ```
   /etc/init.d/network stop
   /etc/init.d/network start
   ```

   This change tells the remote system to run the *uucpd* daemon when a request comes in for UUCP transfer.

2. Add a line similar to the following to your local */etc/uucp/Systems* file:

   ```
   remotehost Any TCP Any
   ```

   The name *remotehost* should be replaced with the name of the remote host you will be calling.

3. Run the */etc/uucp/genperm* command as **root** on the local host to generate the UUCP *Permissions* file.

4. Enter the following command to check that everything is set up correctly:

   ```
   /usr/lib/uucp/uucheck -v
   ```

   There is a great deal of output. You should see output similar to the following for the specific entry you made:

   ```
   When we call system(s): (remotehost)
    We DO allow them to request files.
    They can send files to
    /var/spool/uucppublic (DEFAULT)
    They can request files from
    /var/spool/uucppublic
    /usr/lib/mail
    /usr/people/ftp
   ```

```
 Myname for the conversation will be MyName.
 PUBDIR for the conversation will be /var/spool/uucppublic.

Machine(s): (remotehost)
CAN execute the following commands:
command (rmail), fullname (/bin/rmail)
command (rnews), fullname (/usr/bin/rnews)
command (cunbatch), fullname (/usr/lib/news/cunbatch)
```

The *cu* command does not work for UUCP over a TCP connection. Use the */usr/lib/uucp/Uutry* command instead. *Uutry* is documented completely in the Uutry(1M) reference page and in "Testing With Uutry" on page 163.

Testing the UUCP Connection

There are two basic tools for testing a UUCP connection:

- the *cu* program
- the *Uutry* program

Testing With cu

The *cu* program is used to test the basic functionality of the UUCP connection. When you use *cu* directly, you are performing the login process as if you were the *uucp* programs. The *cu* command is also used for direct modem connections for terminal emulation. The **-d** option to *cu* is used for diagnostics and causes traces of information to be printed out to the standard output (your shell window). You should always use this mode when testing a UUCP connection.

The following command tests the physical connection to the remote station, UUCP configuration files, operating system files, and the *uucico* daemon on the remote station.

Note: The default permissions on devices (*/dev/ttyf2*) are set to 622. For *cu* to access your device, you need to change the permissions to 666.

1. The *cu* command must be run from the local (calling) station. Execute the *cu* command from the local station (*japan*) as follows:

   `/usr/bin/cu -d us`

   You get output similar to the following:

   ```
   conn(us)
   ```

```
Device Type us wanted
mlock ttyf2 succeeded
filelock: ok
fixline(5, 9600)
processdev: calling setdevcfg(cu, us)
gdial(direct) called
getto ret 5
device status for fd=5
F_GETFL=2,iflag=‘12045’,oflag=‘0’,cflag=‘6275’,lflag=‘0’,line=‘1’
cc[0]=‘177’,[1]=‘34’,[2]=‘10’,[3]=‘25’,[4]=‘1’,[5]=‘0’,[6]=‘0’,[7]=‘
0’,
call _mode(1)
Connected
_receive started
transmit started
```

2. When the system pauses, press **<Return>**.
3. When you see the login prompt, log in as *nuucp* and supply the password for *nuucp*. (In this case, the password is **secret**):

```
Break your connection with a tilde(~) dot(.) and a carriage return
(<CR>).
us login: nuucp
Password: secret
IRIX System V Release 3.3.2 us
Copyright (c) 1988,1989,1990 Silicon Graphics, Inc.
All Rights Reserved.
here=japan~[us].
call tilda(.)
call _quit(0)
call _bye(0)
Disconnected
call cleanup(0)
call _mode(0)
```

Testing With Uutry

Uutry is the program that tests the copy-in/copy-out program (*uucico*). *uucico* must be functioning properly before you can actually transfer data. Issue the *Uutry* command from the local station (*japan*) to the remote station (*us*):

/usr/lib/uucp/Uutry us

You should see output similar to the following:

```
/usr/lib/uucp/uucico -r1 -sus  -x5 >/tmp/us 2>&1&
tmp=/tmp/us
mchFind called (us)
conn(us)
Device Type us wanted
mlock ttyf2 succeeded
processdev: calling setdevcfg(uucico, us)
gdial(direct) called
getto ret 5
expect: (ogin:)
```

Note: The system may pause here for several minutes.

```
sendthem (^M)
expect: (ogin:)
^M^M^J^M^J^Jus login:got it
sendthem (nuucp^M)
expect: (ssword:)
nuucp^M^JPassword:got it
sendthem (secret^M)
Login Successful: System=us
msg-ROK
Rmtname us, Role MASTER, Ifn - 5, Loginuser - root
rmesg - 'P' got Pg
wmesg 'U'g
Proto started g
*** TOP ***  -  role=1, setline - X
wmesg 'H'
rmesg - 'H' got HY
PROCESS: msg - HY
HUP:
wmesg 'H'Y
cntrl - 0
send OO 0,exit code 0
Conversation Complete: Status SUCCEEDED
```

When you see "Status SUCCEEDED," *Uutry* has successfully tested *uucico*. Press **<Ctrl-C>** to break out of *Uutry*.

UUCP Error Messages

This section describes common error messages associated with the UUCP environment. UUCP error messages can be divided into two categories: ASSERT Error Messages and STATUS Error Messages.

ASSERT Error Messages

When a process is aborted, the station records ASSERT error messages in */var/spool/uucp/.Admin/errors*. These messages include the filename, the *sccsid*, the *line number*, and the text listed in Table 7-4. In most cases, these errors are the result of file system problems.

Table 7-4 Assert Error Messages

Error Message	Description/Action
CAN'T OPEN	An **open()** or **fopen()** failed.
CAN'T WRITE	A **write()**, **fwrite()**, **fprint()**, or other call failed.
CAN'T READ	A **read()**, **fgets()**, or other call failed.
CAN'T CREATE	A **create()** call failed.
CAN'T ALLOCATE	A dynamic allocation failed.
CAN'T LOCK	An attempt to make an LCK (lock) file failed. In some cases, this is a fatal error.
CAN'T STAT	A **stat()** call failed.
CAN'T CHMOD	A **chmod()** call failed.
CAN'T LINK	A **link()** call failed.
CAN'T CHDIR	A **chdir()** call failed.
CAN'T UNLINK	An **unlink()** call failed.
WRONG ROLE	This is an internal logic problem.

Table 7-4 (continued) Assert Error Messages

Error Message	Description/Action
CAN'T MOVE TO CORRUPT DIR	An attempt to move some bad *C.* or *X.* files to the */var/spool/uucp/.Corrupt* directory failed. The directory is probably missing or has wrong modes or owner.
CAN'T CLOSE	A **close()** or **fclose()** call failed.
FILE EXISTS	The creation of a *C.* or *D.* file was attempted, but the file already exists. This situation occurs when there is a problem with the sequence-file access. This usually indicates a software error.
NO UUCP SERVER	A TCP/IP call was attempted, but there is no server for UUCP.
BAD UID	The *uid* cannot be found in the */etc/passwd* file. The filesystem is in trouble, or the */etc/passwd* file is inconsistent.
BAD LOGIN_UID	The *uid* cannot be found in the */etc/passwd* file. The filesystem is in trouble, or the */etc/passwd* file is inconsistent.
ULIMIT TOO SMALL	The **ulimit** for the current user process is too small. File transfers may fail, so transfer will not be attempted.
BAD LINE	There is a bad line in the *Devices* file; there are not enough arguments on one or more lines.
FSTAT FAILED IN EWRDATA	There is something wrong with the Ethernet media.
SYSLST OVERFLOW	An internal table in *gename.c* overflowed. A big or strange request was attempted.
TOO MANY SAVED C FILES	An internal table in *gename.c* overflowed. A big or strange request was attempted.
RETURN FROM FIXLINE IOCTL	An *ioctl*, which should never fail, failed. There is likely a system driver problem.

Table 7-4 (continued) Assert Error Messages

Error Message	Description/Action
PERMISSIONS file: BAD OPTION	There is a bad line or option in the *Permissions* file. Fix it immediately.
BAD SPEED	A bad line-speed appears in the *Devices* or *Systems* file (Class field).
PKCGET READ	The remote station probably hung up. No action is required.
PKXSTART	The remote station aborted in a nonrecoverable way. This message can generally be ignored.
SYSTAT OPENFAIL	There is a problem with the modes of */var/spool/uucp/.Status*, or there is a file with bad modes in the directory.
TOO MANY LOCKS	There is an internal problem.
XMV ERROR	There is a problem with some file or directory. The problem is likely caused by the spool directory, since the modes of the destinations should have been checked before this process was attempted.
CAN'T FORK	An attempt to fork and execute failed. The current job should not be lost, but will be attempted later (*uuxqt*). No action need be taken.

STATUS Error Messages

Status error messages are stored in the */var/spool/uucp/.Status* directory. This directory contains a separate file for each remote station that your station attempts to communicate with. These files contain status information on the attempted communication, indicating whether it was successful or not. Table 7-5 lists the most common error messages that can appear in these files.

Table 7-5 STATUS Error Messages

Error Message	Description/Action
OK	System status is normal
NO DEVICES AVAILABLE	There is currently no device available for the call. Make sure that there is a valid device in the *Devices* file for the particular station. Check the *Systems* file for the device to be used to call the station.
WRONG TIME TO CALL	A call was placed to the station at a time other than that specified in the *Systems* file.
TALKING	Self-explanatory.
LOGIN FAILED	The login for the given station failed. The problem could be a wrong login and password, wrong number, a very slow station, or failure in getting through the Dialer-Token-Pairs script.
CONVERSATION FAILED	The conversation failed after successful startup. This situation usually means that one side went down, the program aborted, or the line (link) was dropped.
DIAL FAILED	The remote station never answered. The problem could be a bad dialer or the wrong phone number.
BAD LOGIN/MACHINE COMBINATION	The station called you with a login or station name that does not agree with the *Permissions* file. This could be an attempt to breach system security.
DEVICE LOCKED	The calling device to be used is currently locked and in use by another process.

Table 7-5 (continued) STATUS Error Messages

Error Message	Description/Action
ASSERT ERROR	An ASSERT error occurred. Check the */var/spool/uucp/.Admin/errors* file for the error message and refer to "Other UUCP Files" on page 151.
SYSTEM NOT IN Systems	The station is not in the *Systems* file.
CAN'T ACCESS DEVICE	Typically, this message means that the permissions on the device file (*/dev/tty**) are not set correctly. Some programs set these permissions, and if terminated abnormally, do not reset them to correct states. Also, check the appropriate entries in the *Systems* and *Devices* files.
DEVICE FAILED	The attempt to open the device failed.
WRONG MACHINE NAME	The called station is reporting a name different from the one expected.
CALLBACK REQUIRED	The called station requires that it call your computer back to start a connection.
REMOTE HAS A LCK FILE FOR ME	The remote site has a LCK file for your computer. The remote station could be trying to call your computer. If they have an older version of UUCP, the process that was talking to your station may have failed earlier, leaving the LCK file. If the remote site has the new version of UUCP and they are not communicating with your computer, then the process that has a LCK file is hung.
REMOTE DOES NOT KNOW ME	The remote computer does not have the node name of your computer in its *Systems* file.
REMOTE REJECT AFTER LOGIN	The ID used by your computer to log in does not agree with what the remote computer was expecting.

Table 7-5 (continued) STATUS Error Messages

Error Message	Description/Action
REMOTE REJECT, UNKNOWN MESSAGE	The remote computer rejected the communication with your computer for an unknown reason. The remote computer may not be running a standard version of UUCP.
STARTUP FAILED	Login succeeded, but initial handshake failed.
CALLER SCRIPT FAILED	The problem indicated by this message is usually the same as that indicated by `DIAL FAILED`. However, if it occurs often, suspect the caller script in the *Dialers* file. Use *uutry* to check the caller script.