Chapter 1

Terminals and Modems

This chapter describes some of the software and hardware considerations for setting up terminals and modems. This information also applies to most generic serial devices. It does not cover specialized devices (such as the Dial & Button Box and the Spaceball). For more information on these devices, please contact Silicon Graphics. "Dumb" serial printers are also connected using the TTY subsystem, but they are covered in Chapter 2, "Printers."

Serial communication between users and serial devices, such as modems and terminals, is supported by the TTY subsystem. This chapter tells you how to administer the TTY subsystem with respect to terminals and modems. It includes information about:

- How the TTY subsystem works. See "About the TTY Subsystem" on page 2.
- Connecting an ASCII terminal. See "Installing an ASCII Terminal" on page 4.
- Connecting a modem. See "Installing a Modem" on page 9.
- Information on managing the TTY subsystem. See "Managing the TTY Subsystem" on page 18.
- Serial port cabling and pin signals for the standard serial ports. See "Serial Port Pin Signal Tables" on page 20.

To connect peripherals that are not covered in this chapter, refer to the documentation that accompanies the peripheral.

About the TTY Subsystem

A series of four processes (init(1M), getty(1M) or uugetty(1M), login(1), and one of sh(1), csh(1), or ksh(1)) connect a user to the IRIX system. *init* is invoked as the last step in the boot procedure and is responsible for spawning general processes. It spawns a process for each serial line that a user may log in on, as defined by instructions in the file */etc/inittab*. The *getty* or *uugetty* commands in */etc/inittab* require a <*line*> argument, which is the name of a special file in the */dev* directory. For a description of other arguments that may be used with *getty* and *uugetty*, see the getty(1M) and uugetty(1M) reference pages.

A user attempting to make a connection generates a signal that is routed by the hardware to the *getty* process (derived from "get a tty") for one of the TTY line files in */dev. getty* responds by sending an entry from the file */etc/gettydefs* down the line. The *gettydefs* entry used depends on the <*speed*> argument used with the *getty* command. (In the SYNOPSIS of the getty(1M) reference page, the argument name is <*speed*>, but it is really a pointer to the <*label*> field of a *gettydefs* entry.) If no <*speed*> argument is provided, *getty* uses the first entry in *gettydefs*. Among the fields in the *gettydefs* entry is the login prompt (described in "Checking Line Settings Using IRIX Shell Commands" on page 18).

On receiving the login prompt, the user enters a login name. *getty* starts *login*, using the login name as an argument. *login* issues the prompt for a password, evaluates the user's response, and if the password is acceptable, calls in the user's shell as listed in the */etc/passwd* entry for the login name. If no shell is named, */bin/sh* is furnished by default.

/bin/sh executes */etc/profile* and then executes the user's *.profile*, if it exists. The *.profile* or *.login* file often contains *stty* commands that reset terminal options that differ from the defaults. */bin/csh* executes */etc/cshrc*, *.cshrc*, and *.login*. The connection between the user and the IRIX system has now been made.

Terms

The following terms are used when describing serial devices and the TTY system:

TTY
: The near-classic abbreviation for teletypewriter. The term covers the whole area of access between the IRIX system and peripheral devices, including the system console. It shows up in commands such as *getty* and *stty*, in the names of device special files such as */dev/ttyd1*, and in the names of files such as */etc/gettydefs*, which is used by *getty*.

TTY line
: The cable connecting the computer to the serial device.

port
: The connector on the computer where the TTY line attaches.

line settings
: A set of characteristics for data transmission over the line.

baud rate
: The speed at which data is transmitted over the line. A part of the line settings.

mode
: The characteristics of the terminal interface, and a part of line settings. The TTY line and the terminal must be working in the same mode before communication can take place. Described in termio(7).

hunt sequence
: A series of line settings such as different baud rates. During the login sequence, a user looking for a compatible connection to the computer can go from one setting to the next by sending a BREAK signal. Also referred to as a rotary.

terminal options
: Selectable settings that define the way a given terminal operates. Described in termio(7).

Installing an ASCII Terminal

This section describes the procedures for connecting and configuring an ASCII terminal. The *diagnostics terminal*, or *console*, is the ASCII terminal connected to a serial port on the I/O panel of a server. It can be connected to any of these ports, depending on your system:

- To the alternate console port on a graphics system.
- To the console port (labeled port 1) or port 2 on a non-graphics system.
- To any of the four serial ports on larger systems such as the CHALLENGE® M or Onyx™.

The messages produced by the power-on diagnostics appear on the screen of this terminal.

Connecting the Terminal Hardware

Connect an appropriate cable from the ASCII terminal to the serial port on the computer I/O panel. See "Cabling the Serial Ports" on page 22 for information about port cabling and pin signals.

Configuring the Terminal Software

This section tells you how to configure IRIX software to use an ASCII terminal with your computer system.

During configuration, you need to:

- Edit the file */etc/ttytype* to associate the TTY port you have chosen with a file that describes your terminal settings. These terminal descriptor files reside in the directory */usr/lib/terminfo*.
- Update the file */etc/inittab* to allow a login to that port.

Note: The utilities described in this section are distributed as part of the *eoe2.sw.terminfo* package. You must have installed this package on your system in order to use these utilities. See IRIX Admin: Software Installation and Licensing for information and specific instructions on installing this package.

The */usr/lib/terminfo* directory contains files that describe different terminal models, their capabilities, and how they operate. For most ASCII terminal models, you do not need to edit this database.

Note: If your terminal is not in the database, or if it does not work properly after you have configured the software, you may need to write a terminal description. Refer to the optional *Topics in IRIX Programming* and the tset(1), stty(1), and terminfo(4) reference pages. Additionally, your terminal may support a compatible terminal mode (such as VT100). Check your terminal documentation for compatibility modes.

The directory */usr/lib/terminfo* is divided into numerical and alphabetical subdirectories. Each subdirectory contains entries for terminals whose names begin with that character. For example, */usr/lib/terminfo/v* contains the entry for the Visual 50. The entry name listed in the subdirectory is *v50am*. Use the *fgrep* command to search for the terminal model name that corresponds to the entry name.

To find the entry name for your terminal and to configure software for an ASCII terminal, follow these steps:

1. Log in as root or become the superuser by entering the *su* command.
2. Change directories to */usr/lib/terminfo*, type:

   `cd /usr/lib/terminfo`

3. Find the entry name for your terminal. Issue the *fgrep* command with a string that you suspect could make up part of your terminal name.

   `ls -R | fgrep -i` *string*

   If this fails, examine the subdirectories of */usr/lib/terminfo*, which contain all the terminal entries.

4. Once you find the terminal name as it appears in */usr/lib/terminfo*, issue the *infocmp* command to determine the model name of your terminal. For example, for a Visual 50, issue this command:

   `infocmp -I v50am`

   You see a display that begins with this line:

   ```
   v50am|visual50 (v50 emulation) with automatic margins,
   ```

   The data in the first field (`v50am`) is the model name of your terminal.

Tip: If it looks like there are many terminal names that could apply, use this example shell script to help speed up the search. This example uses the string `wy60*` to help locate all the Wyse60 models. Substitute your own string, as appropriate.

```
sh
for i in wy60*
do
                infocmp -I $i
done
```

Note: If you choose an incorrect model name, you can change it later; the line still works, but screen-based commands do not display correctly.

5. Edit */etc/ttytype* and enter the model name.

   This file associates the model name with the port where the terminal is connected. In the line that contains the port you are using, replace `du` with the model name of your terminal.

   An */etc/ttytype* might look like this example:

```
iris-ansi systty
?v50am    ttyd1
?v50am    ttyd2
?v50am    ttyd3
?v50am    ttyd4
?v50am    ttyd5
?v50am    ttyd6
?v50am    ttyd7
?v50am    ttyd8
?v50am    ttyd9
?v50am    ttyd10
?v50am    ttyd11
?v50am    ttyd12
```

   Hint: The optional question mark (?) at the beginning of a line in */etc/ttytype* causes *tset* to display the model name as supplied (for example, v50am here) and prompts for the model name you are currently using. If you are not using the model name as provided, type in the name that you are using; otherwise, press **<Enter>** to accept the default. This provides an easy way to switch terminal settings if you use more than one type of terminal.

   tset is normally called in your login startup script (*.login* or *.profile*). *tset* commands use information from */etc/ttytype* and */usr/lib/terminfo* to initialize the terminal. These files also provide information on setting environment variables so that editors and other programs know how to communicate with the terminal. See tset(1) for detailed information.

6. Edit */etc/inittab* so that you can log in. */etc/inittab* provides information about how the computer ports behave. This is a sample from an */etc/inittab* file:

```
t1:23:respawn:/etc/getty -s console ttyd1 co_9600 # port 1
t2:23:off:/etc/getty -N ttyd2 co_9600 # port 2
t3:23:off:/etc/getty -N ttyd3 co_9600 # port 3
t4:23:off:/etc/getty -N ttyd4 co_9600 # port 4
```

   Here are two example entries, with an explanation of each field in the entries:

```
t1:23:respawn:/etc/getty -s console ttyd1 co_9600
```

```
t2:23:off:/etc/getty -N ttyd2 co_9600
```

   t1 t2 uniquely identifies the entry.

   23 defines the *init* run level in which this entry is to be processed. A 23 means this entry is to be processed in run levels two and three. Refer to init(1M) for more information about run levels.

   off means do not perform the action on the process field of *init*.

   respawn means start the program in the process field of *init* and restart it every time the process terminates. See inittab(4) for a description of all possible actions.

   /etc/getty -s console ttyd1 co_9600
   runs the *getty* process on the port labeled 1 at the baud rate and with the options specified in the *co_9600* entry in the */etc/gettydefs* file. The **-s console** option instructs *getty* that the login shell generated is a system console, and therefore receives system error messages.

   /etc/getty -N ttyd2 co_9600
   runs the *getty* process on the port labeled 2 at the baud rate and with the options specified in the *co_9600* entry in the */etc/gettydefs* file. The **-N** option instructs *getty* to honor the presence of the */etc/nologin* file, which does not allow remote logins over the network.

   To enable you to log in to the terminal connected to the port labeled 2**,** find this line:

```
t2:23:off:/etc/getty -N ttyd2 co_9600
```

   Change it to:

```
t2:23:respawn:/etc/getty -N ttyd2 co_9600
```

7. If you don't plan to run the terminal at 9600 baud, replace `co_9600` in */etc/inittab* with the correct entry name from */etc/gettydefs*.

   /etc/inittab refers to */etc/gettydefs* for information about the terminal line settings. In the example from */etc/inittab* above, `co_9600` refers to the name of an entry in */etc/gettydefs*; it defines a 9600 baud console setting.

   To see what entries are defined in */etc/gettydefs*, examine the file, or see "Checking Line Settings Using IRIX Shell Commands" on page 18. To make a new entry, see "Creating and Testing Line Settings" on page 19.

8. Inform *init* of the change to */etc/inittab*, and start a *getty* process for the port:

   `telinit q`

9. Power on the terminal. The workstation sends a login prompt to the terminal screen. Press **<Enter>** if the login prompt doesn't appear.

Note: If the default line speed set in */etc/inittab* is incorrect, the prompt may be garbled or may not appear. You should choose another line speed.

Setting Terminal Options

The TTY system described thus far establishes a basic style of communication between the user's terminal and the IRIX operating system. Once a user has successfully logged in, he or she may prefer terminal options other than the default set.

The *stty* command controls terminal options. Many users add an *stty* command to their *.profile* or *.login* file so the options they want are automatically set as part of the login process. Here is an example of a simple *stty* command:

```
stty cr0 nl0 echoe -tabs erase '^H'
```

The options in the example mean:

cr0 nl0 — No delay for carriage return or new line. Delays are not used on a video display terminal, but are necessary on some printing terminals to allow time for the mechanical parts of the equipment to move.

echoe — Erases characters as you backspace.

-tabs Expand tabs to spaces when printing.

erase '^H' Change the character-delete character to a Ctrl-H. The default character-delete character is the pound sign (#). Most terminals transmit a Ctrl-H when the **<Backspace>** key is pressed. Specifying this option makes **<Backspace>** delete the character that was just typed.

Installing a Modem

This section describes how to install a modem onto a serial port. It describes how to set up the hardware and software to support a UUCP modem connection.

Topics in this section are:

- "Before You Begin" on page 10 provides a list of prerequisites that you should be aware of.
- "Connecting the Modem Hardware" on page 11 describes the hardware connections that you should make before configuring the software.
- "Setting Up the Modem Software" on page 12 describes the software configuration procedure.

Silicon Graphics is compatible with most industry-standard modems. This section uses the US Robotics Sportster and Telebit modems as examples for the installation procedure. If you are using another type of modem, substitute its information where applicable.

You can use the standard system software to set up a modem for three different modes:

- As a dial-in modem—other users can call your modem to log in to your system.
- As a dial-out modem—you use the modem to call a remote modem to log in to the system to where the remote modem is connected.
- As a dial-in/dial-out modem—other users can call your modem and you can call other modems.

If you want to set up your modem for a SLIP or PPP connection, complete the modem configuration steps in the chapter and then refer to the SLIP or PPP configuration procedures in IRIX Admin: Networking and Mail.

Before You Begin

There are a few prerequisites you should be aware of before installing the modem. Review the following points before installing the modem.

- Choose a modem configuration script. Silicon Graphics provides a set of predefined configuration scripts for the many popular modems to help with your modem installation. These scripts describe the correct jumper settings for your modem and contain the expected AT commands and values for your modem to operate correctly. If your modem is not listed, you can use the *fix-hayes* script as a template and customize it to your modem's specifications.

  Table 1-1 shows the modem types and their configuration scripts.

  Table 1-1 Modem Configuration Scripts

Modem	Configuration Script
Telebit T2500, T1600, QBlazer, T3000, and WorldBlazer	*fix-telebit*
ZyXEL U-1496	*fix-zyxel*
Intel 14.4ex	*fix-intel*
DSI 9624 models	*fix-dsi*
US Robotics (also called USR)	*fix-usr*
Hayes ACCURA (also called Hayes14)	*fix-hayes*

- Check that the UUCP software is installed. The *eoe2.sw.uucp* subsystem must be installed on your system. It is shipped with your IRIX system software distribution, but it is not installed by default. You can determine whether *eoe2.sw.uucp* is installed by using the command:

  `versions | grep uucp`

  Check the *versions* output for the following line:

  ```
  I eoe2.sw.uucp uucp utilities
  ```

  If the line above is not present, *eoe2.sw.uucp* is not installed. See IRIX Admin: Software Installation and Licensing for instructions on installing a subsystem and install *eoe2.sw.uucp*.

- Check the baud rate of the modem. The installation procedure examples assume you are using a 28,800 or 14,400 baud modem. Be sure to substitute your modem baud rate where applicable.
- Check your serial port interface speed. The interface speed is the maximum speed of your computer's serial port, generally 38,400 bps. Check your system *Owner's Guide* to be sure. Use the maximum speed common to both modem and computer.
- If your modem supports hardware flow control (RTS/CTS) be sure to use it and disable support for software flow control (XON/XOFF).
- Determine how you want to use your modem: dial-in only, dial-out only, or dial-in/dial-out.

Connecting the Modem Hardware

To set up a modem, you must first make the following physical connections. With power to the modem off, connect the modem to:

- A serial port on your system I/O panel. Be sure to use the Silicon Graphics modem cable (model number X5-25TO8, P/N 018-8109-00X) or one created to the Silicon Graphics serial port specifications. See your system *Owner's Guide* or "Cabling the Serial Ports" on page 22 for cabling information.

  Caution: Do not connect your system to the modem with a cable for an IBM®-PC/AT or an Apple Macintosh®. It will not function correctly. You must use a Silicon Graphics cable or one made to the specifications for the serial port pinouts as described in serial(7) and "Cabling the Serial Ports" on page 22.

- A power source. Connect the power cord that came with the modem to the modem and an AC outlet.
- A telephone jack. Attach the phone cable supplied with the modem into the telephone jack and the correct phone jack on the back of the modem.

Setting Up the Modem Software

The major tasks for setting up the modem software are fairly similar, despite how you choose to use your modem.

Table 1-2 provides an overview of the major tasks for each of the three usage modes. An “x” in the table cell denotes the task is performed.

Table 1-2 Overview of Modem Software Setup

Task	Dial-in Only	Dial-out Only	Dial-in/ Dial-out
Turn off the port in */etc/inittab*	x	x	x
Execute *telinit q* to reinitialize the port	x	x	x
Run *fix-modem* configuration script	x	x	x
Edit *Devices* file		x	x
Chown device to *uucp*		x	x
Update action field in */etc/inittab*	x		x

Modem Software Setup, All Modes

The installation procedure in this section is used to start the software setup for the three usage modes. Follow the procedure until you reach the point where you are directed to the section that is specific for the usage mode you have chosen. Then continue with that section. For example, if you are configuring your modem for dial-in only, complete the procedures in this section and the “Dial-in Mode” section.

1. Log in as root.

2. Edit */etc/inittab* to turn off the port you will be using for your modem so you can configure it correctly.

   Find the line for the port. The line begins with `t`*portnumber*, where *portnumber* is the number of a serial port. For example, if you connected your modem to port 2, look for a line similar to the following:

   ```
   t2:23:respawn:/etc/getty -N ttyd2 co_9600 #port 2
   ```

   Change respawn to **`off`**. This tells the system not to restart the */etc/getty* program when it exits. If it is already off, continue to the next step. The line should look like this when you are finished:

   ```
   t2:23:off:/etc/getty -N ttyd2 co_9600 #port 2
   ```

3. Write and exit the */etc/inittab* file.

4. Inform *init* about the change to *inittab* with the following command:

   ```
   telinit q
   ```

5. Run the modem configuration script for your modem type (see Table 1-1). Select the appropriate argument for its usage mode from Table 1-3 for the port you are configuring.

   Table 1-3 Determining the Usage Mode for Modem Configuration

Dial-in	Dial-out	Dial-in/Dial-out
-i	**-o**	**-io**

   Note: Be sure to view the entire configuration script for information and command arguments specific to your modem type before running the script.

   For example, if you want to configure a US Robotics Sportster modem for dial-in/dial-out, connected to port 2, use the *fix-usr* command and specify a dial-in/dial-out argument (**-io**) and the port number to which the modem is connected. For example, enter the command:

   ```
   /etc/uucp/fix-usr -io 2
   ```

You can also specify the optional modem model type and the modem speed, using the command:

/etc/uucp/fix-usr -io -m SPORT -s 38400 2

As another example, to configure a Telebit modem for dial-in only mode, use the *fix-telebit* command and specify a dial-in modem (**-i**), the model of the modem (tb+, t1000, t1600, t2500) and its speed, and the port number to which the modem is connected.

In this example, if a T2500 modem is connected to port 2 at 19,200 baud, dial-in only, enter the following command:

/etc/uucp/fix-telebit -i -m t2500 -s 19200 2

- If you are configuring a dial-out modem, continue with the section, “Dial-out Mode” on page 14.
- If you are configuring a dial-in modem, continue with the section, “Dial-in Mode” on page 15.
- If you are configuring a dial-in/dial-out modem, continue with the section, “Dial-out Mode” on page 14, followed by “Dial-in Mode” on page 15.

Dial-out Mode

This section describes additional steps for configuring dial-out only and dial-in/dial-out modems. Be sure you have completed the steps in “Modem Software Setup, All Modes” on page 12 before starting the setup in this section.

You turn on dial-out modem software by editing the */etc/uucp/Devices* file. You also have to change ownership of the port to uucp.

1. Edit the */etc/uucp/Devices* file to specify the port to which the modem is connected and to set the correct baud rate for the modem. Also, add a “Direct” line for use in debugging problems. The lines have the following forms:

   ```
   ACU ttynportnumber null baud 212 x dialer
   Direct ttydportnumber - baud direct
   ```

   - If your modem runs at 2400 baud or lower, the *n* field should be **m**.
   - If it runs faster than 2400 baud and supports hardware flow control, the *n* field should be **f**.
   - All other cases can use **d**.

The value for *portnumber* is the number of the serial port to which the modem is connected. The *baud* variable is the baud rate at which you will run the modem. The *dialer* variable is the model tag, as listed in the first field of the */etc/uucp/Dialers* file.

Note: If your modem type is not listed in the *Dialers* file, you may want to obtain the most up-to-date version of the *Dialers* file. It is available online from Silicon Graphics with the *ftp* utility.

For example, for a US Robotics modem connected to serial port 2 that can accept and send data at 38400 baud (using RTS/CTS hardware flow control), add the following lines to the *Devices* file:

```
ACU ttyf2 null 38400 212 x usr
Direct ttyd2 - 38400 direct
Direct ttyf2 - 38400 direct
```

For a Telebit modem connected to serial port 2 and running at 19200 baud, add the following lines:

```
ACU ttyf2 null 19200 212 x telebit
Direct ttyd2 - 19200 direct
Direct ttyf2 - 19200 direct
```

Save and exit the *Devices* file.

2. Change the ownership of the device files. For example, if you are using serial port 2, give the following command (substituting the appropriate *tty* device as determined above):

chown uucp /dev/tty[*dfm*]**2**

- If you are configuring your modem for dial-in/dial-out mode, continue with the section “Dial-in Mode” on page 15.

When you have finished the above process, try to dial out through the modem to verify the dial-out setup, as described in “Dialing Out to Another Modem” on page 17. For more information on dialer options, see the file */etc/uucp/Dialers*.

Dial-in Mode

Be sure you have completed the steps in “Modem Software Setup, All Modes” on page 12. For dial-in/dial-out mode, be sure you have also completed the steps in “Dial-out Mode” on page 14.

Complete the configuration by editing the file */etc/inittab*. Perform the following steps:

1. Edit the file */etc/inittab* again so the system recognizes that the port is used for the modem (as opposed to a terminal). This editing also sets the interface speed on the port and enables user logins through the port.

   Find the line for the port to which the modem is connected. The line begins with `t`*portnumber*, where *portnumber* is the number of a serial port. For example, if you connected your modem to port 2, look for a line similar to the following:

   ```
   t2:23:off:/etc/getty -N ttyd2 co_9600 #port 2
   ```

   Make the following changes to this line:

   - Change off to **`respawn`**.
   - Change */etc/getty* to */usr/lib/uucp/uugetty*.
   - Add the dial-in **-i** option with the modem model argument. Pick the modem model variable from Table 1-4 (or check the */etc/uucp/Dialers* file for additional variables):

   Table 1-4 inittab Dial-in Options by Modem Type

Modem Type	-i Argument
Telebit T2500, T1600, QBlazer, T3000, and WorldBlazer	telebitin,conn
ZyXEL U-1496	zyin,conn
Intel 14.4ex	intelin,conn
DSI 9624 models	dsiin,conn
US Robotics (also called USR)	usrin,conn
Hayes ACCURA (also called Hayes14)	hayesin,conn

   - Change ttyd2 to **`ttyf2`**.
   - Change co_9600 to **`dx_38400`** (this is the serial port interface speed. It lets the modem answer and connect at 38400 and slower baud rates.) Do not set this value to a speed slower than your modem.
   - Change the comment at the end of the line to note that this port is now used for a modem; include its phone number for reference.

For example, if you connected a US Robotics modem to serial port 2, and found a line similar to this line in */etc/inittab*:

```
t2:23:off:/etc/getty -N ttyd2 co_9600 #port 2
```

You would change it to read:

```
t2:23:respawn:/usr/lib/uucp/uugetty -Nt60 -iusrin,conn ttyf2
dx_38400 #Modem 555-2345
```

If you are setting up a Telebit modem to serial port 2, and found a line similar to this one:

```
t2:23:off:/etc/getty -N ttyd2 co_9600 #port 2
```

You would change it to read:

```
t2:23:respawn:/usr/lib/uucp/uugetty -Nt60 -itelebitin,conn ttyf2
dx_19200 #Modem 555-3456
```

2. Write and exit the */etc/inittab* file.
3. Inform *init* about the change to */etc/inittab* with the following command:

```
telinit q
```

To test the dial-in setup, have someone try to dial in to the system through this modem.

To dial out through the modem to verify the dial-out setup, continue with the section "Dialing Out to Another Modem."

Dialing Out to Another Modem

The *cu* utility dials your modem. Before you attempt to dial out, make sure your local modem is connected to both your system and a working telephone line. Also, make sure the modem is turned on. The *cu* syntax to dial the number 1-800-555-1234 is:

```
cu 18005551234
```

Refer to the cu(1) reference page for complete information on *cu*. If everything is working, you should hear the modem dialing, and after a short time you should see the following message on your console:

```
Connected
```

- If no login prompt is displayed, press **<Enter>** once.
- If a prompt is displayed, continue and log in to the remote system.
- To disconnect, press the **<Enter>** key, the tilde key (~), a period (.), and the **<Enter>** key again.
- If you cannot connect, refer to the troubleshooting tips in "Troubleshooting Your Modem Setup" in Chapter 5.

Managing the TTY Subsystem

This section covers the following topics:

- "Checking Line Settings Using IRIX Shell Commands"
- "Creating and Testing Line Settings"
- "Modifying Line Characteristics"

Checking Line Settings Using IRIX Shell Commands

The */etc/gettydefs* file contains information used by the *getty* command to establish the speed and terminal settings for a line. The general format of the *gettydefs* file is:

label# initial-flags # final-flags #login-prompt #next-label

The following example shows a few lines from a *gettydefs* file:

```
dx_9600# B9600 # B9600 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #dx_9600
dx_4800# B4800 # B4800 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #dx_4800
dx_2400# B2400 # B2400 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #dx_2400
dx_1200# B1200 # B1200 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #dx_1200
du_1200# B1200 # B1200 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #du_300
du_300#  B300  # B300  SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #du_2400
du_2400# B2400 # B2400 SANE TAB3 HUPCL #\r\n\n$HOSTNAME login: #du_1200
```

The entries beginning with `dx` are typically used for terminals or modems; those beginning with `du` are typically for low-speed modems. See gettydefs(4) for more information on the fields of each entry.

These entries form a single, rotary hunt sequence; the last field on each line is the label of the next line. The next-label field for the last line shown points back to the first line in the sequence. The object of the hunt sequence is to link a range of line speeds. If you see garbage characters instead of a clear login prompt, press the **<Esc>** key to force *getty* to step to the next entry in the sequence. The hunt continues until the baud rate of the line matches the speed of the user's terminal.

The flag fields shown have the following meanings:

B300-B19200 The baud rate of the line.

HUPCL Hang up on close.

SANE A composite flag that stands for a set of normal line characteristics.

IXANY Allow any character to restart output. If this flag is not specified, only DC1 **<Ctrl-Q>** restarts output.

TAB3 Send tabs to the terminal as spaces.

For a description of all *getty* flags, see termio(7).

Creating and Testing Line Settings

Create new lines for the *gettydefs* file by following the example shown in "Checking Line Settings Using IRIX Shell Commands." Each entry in the file is followed by a blank line. After editing the file, run the command:

```
/etc/getty -c /etc/gettydefs
```

This causes *getty* to scan the file and print the results on your terminal. Any unrecognized modes or improperly constructed entries are reported.

Modifying Line Characteristics

You can modify TTY line characteristics using an IRIX editor, such as *vi*, to edit */etc/inittab*.

The */etc/inittab* file contains configuration instructions for the */etc/init* command. The general format of a line entry in the */etc/inittab* file is as follows:

identification:level:action:process

The four colon-separated fields are as follows:

identification A unique one or two character identifier for the line entry.

level The run-level(s) in which the entry is to be performed.

action How */etc/init* treats the process field (refer to the inittab(4) reference page for complete information).

process The shell command to be executed.

/etc/inittab contains several entries that spawn *getty* processes. The following example is a selection of such entries from a sample */etc/inittab*:

```
t1:23:respawn:/etc/getty -s console ttyd1 co_9600
t2:23:respawn:/etc/getty ttyd2 co_9600
```

There are at least three things you might want to do to an *inittab* entry for a TTY line:

- Change the action. Two actions that apply to TTY lines are "respawn" and "off" (see the inittab(4) reference page for complete information on this field).
- Add or change arguments to */etc/getty* in the process field. A frequently used argument is **-t***nn*. This tells *getty* to hang up if nothing is received within *nn* seconds. It's good practice to use the **-t** argument on dialup lines.
- Add or change comments. You can start your comments after a pound sign (#), and insert comments inline after a semi-colon (**;**).

Serial Port Pin Signal Tables

This section outlines the serial support on IRIS workstations and servers and provides instructions for connecting peripheral devices to the serial ports. For information on the optional serial ports, see the document that accompanies that hardware option.

Defining the Serial Interface

The workstation or server provides an RS-232 (or RS-423) or a DB-9 (9-pin) compatible serial interface. Additionally, some systems are equipped with the mini-DIN8 serial interface. These ports do not lay out the pins in a conventional pattern such as DB-9 or RS-232; instead, the plug is cylindrical. Adapter cables are available commercially or through Silicon Graphics to connect mini-DIN8 ports with DB-9 and RS-232 ports. Consult your system *Owner's Guide* for complete information on your system's serial ports. All serial data cables that you connect to the computer should be shielded. The computer can easily drive and receive signals on a 50-foot cable, and it typically drives and receives signals on a cable up to 200 feet long.

There are two types of serial interface equipment available: *Data Terminal Equipment* (*DTE*) and *Data Communications Equipment (DCE)*. The primary difference between DTE and DCE is the use of several pins on the connector. For example, DTE devices output on pin 2 and input on pin 3. DCE devices output on pin 3 and input on pin 2. You can connect a DTE interface directly to a DCE interface.

To connect either a DCE to a DCE, or a DTE to a DTE, use a *null modem* cable. A null modem cable allows equipment interfaces of the same type to be connected by swapping the appropriate input and output pins.

The serial ports for workstations and servers are all configured as DTE devices. Most terminals are also configured as DTE devices. Therefore, to connect a terminal to a workstation that has RS-232 or DB-9 connectors, use a null modem cable (if this connection does not work correctly, consult the documentation for your terminal to determine its requirements). To connect a modem to the workstation, use a cable that connects each pin of the serial port to the corresponding pin of the modem. No signals need to be swapped. Ensure that your cable is a straight connection before using it with your system. Connect other peripheral devices according to the configuration data provided with the device.

Silicon Graphics provides three special files that determine the signals that are used on each port. The special files beginning with *ttyd* are used for devices such as terminals; the files *ttymX* are used for modems that do not use hardware flow control; and *ttyfX* files are used for devices that can use hardware flow control on the RTS and CTS pins.

Cabling the Serial Ports

This section describes the cables typically used to connect the computer to terminals and modems. The serial ports are designed to connect directly to Data Communications Equipment (DCE) devices such as modems through a serial cable. Each wire on this cable connects the same pin of the computer socket to the device, that is, it has a pin-to-pin correspondence. Connecting to Data Terminal Equipment (DTE) devices such as terminals requires a different cable arrangement: a *null modem* cable. A null modem cable has input and output signals swapped between the two connectors.

DB-9 Serial Connector Cabling

Table 1-5 lists the pin definitions for an example of a 9-pin (DB-9) null modem cable, as shown in Figure 1-1. Connect pins that are shown separated by commas in the Terminal column together with a wire or jumper at that end of the cable. Then connect these joined pins to the pin shown in the 9-pin column at the 9-pin end of the cable.

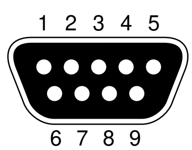

Figure 1-1 DB-9 Connector

Table 1-5 Pin Definitions for a Null Modem Cable

9-Pin	Signal	Terminal	Signal
2	Transmit Data	3	Receive Data
3	Receive Data	2	Transmit Data
4	Request To Send	5	Clear To Send
5	Clear To Send	4	Request To Send
7	Signal Ground	7	Signal Ground
8	Data Carrier Detect	6,8	Data Set Ready and Data Carrier Detect
9	Data Terminal Ready	20	Data Terminal Ready

Most terminals do not require the various handshaking lines such as *Clear To Send* or *Data Set Ready* and work with a three-wire null modem cable. To make one of these, you simply need to swap the signals for pins 2 and 3, and you need to connect pin 7 of the computer to pin 7 of the terminal. Table 1-6 lists the pin definitions for a three-wire null modem cable.

Table 1-6 Pin Definitions for a Three-wire Null Modem Terminal Cable

Computer	**Signal**	**Terminal**	**Signal**
2	Transmit Data	3	Receive Data
3	Receive Data	2	Transmit Data
7	Signal Ground	7	Signal Ground

Note that the pinout of these DB-9 connectors is different from that of the DIN connectors that may be next to them. These DIN connectors also have different pinouts from the mini-DIN8 connectors used on some systems and documented in "Mini-DIN8 Serial Connector Cabling." The DB-9 and DIN connectors are connected to the same internal port hardware.

Caution: Do not use a cable designed for an IBM PC/AT™ compatible 9-pin connector. It does not work correctly with your Silicon Graphics workstation.

For modem devices that use RTS/CTS hardware flow control, the following pinout allows "full flow control." This cable is required to implement */dev/ttyf** devices. This cable also supports */dev/ttym** devices. (When using *ttym* devices, ensure that the connected device does not require flow control signals.) Figure 1-2 shows the DB-9 to DB-25 RTS/CTS cable and Table 1-7 lists the correct pin-out for hardware flow control devices.

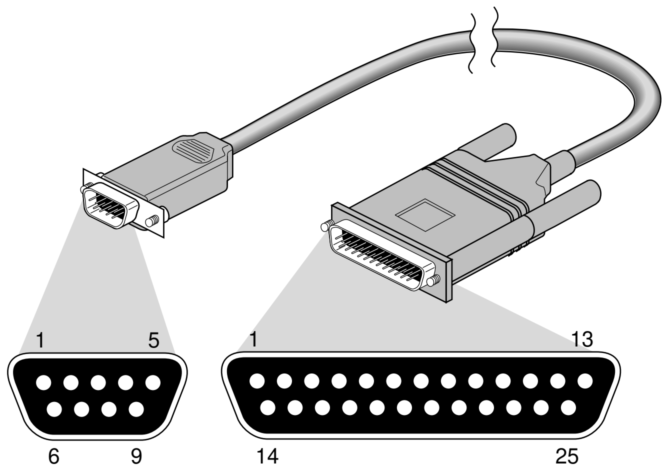

Figure 1-2 DB-9 to DB-25 RTS/CTS Modem Control Cable

Table 1-7 Pin Definitions for a DB-9 to DB-25 RTS/CTS Modem Control Cable

DB-9-Male	Signal	DB-25-Male	Signal
1	not used	1	not used
2	Transmit Data	2	Transmit Data
3	Receive Data	3	Receive Data
4	Request to Send*	4	Request to Send*
5	Clear to Send*	5	Clear to Send
6	not used	6	not used
7	Signal Ground	7	Signal Ground
8	Data Carrier Detect	8	Data Carrier Detect
9	Data Terminal Ready	20	Data Terminal Ready

* RTS and CTS are ignored (optional) if using */dev/ttym** but required if using */dev/ttyf**.

Note: This cable can be used with a null modem adapter for terminals and printers (see Table 1-5); however, it is recommended that you use this cable exclusively for modem connections. The IBM PC/AT to modem cable (“off the shelf cable”) does not work properly with your workstation. For additional information see the serial(7) reference page.

Mini-DIN8 Serial Connector Cabling

There are three basic cable configurations for the mini-DIN8 serial ports. See your *Owner's Guide* to determine if you have a mini-DIN8 port on your workstation or server. Depending on the cables used, some functionality may be sacrificed. Note that the pinout of these mini-DIN8 connectors is different from that of the DIN connectors on larger systems. These larger systems also have DB-9 connectors that are connected to the same internal port hardware.

For most dumb terminals you should use a commercially available cable, Macintosh SE® to ImageWriter1. This cable uses the normal three-wire connection and is used as a */dev/ttyd** device.

Figure 1-3 shows the mini-DIN8 to DB-25 Serial Terminal Cable and Table 1-8 lists the pin configuration.

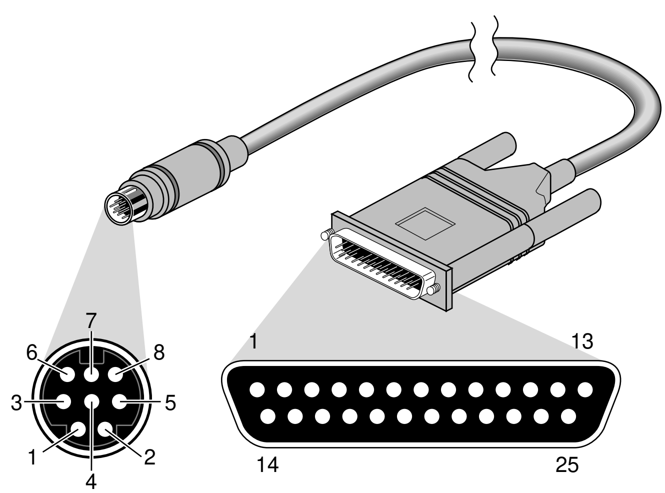

Figure 1-3 Mini-DIN8 to DB-25 Serial Terminal Cable

Table 1-8 Pin Definitions for a Mini-DIN8 to DB-25 Serial Terminal Cable

Mini-DIN8-Male	Signal	DB-25-Male	Signal
1	not used	1	not used
2	not used	2	Transmit Data
3	Transmit Data	3	Receive Data
4	Signal Ground	4	Request to Send
5	Receive Data	5	Clear to Send
6	not used	8	Data Carrier Detect*
7	Data Carrier Detect*	20	Data Terminal Ready
8	Signal Ground		

* */dev/ttym** devices should be used with this cable only if the system must notice when the terminal or printer is powered off.

Note: A Macintosh SE cable has other pins connected but they can be ignored.

For modem devices using RTS/CTS hardware flow control, the following pinout allows "full flow control." This cable is required to implement */dev/ttyf** devices. It also supports */dev/ttym** devices. Table 1-9 shows the pinout:

Table 1-9 Pin Definitions for a Mini-DIN8 RTS/CTS Modem Cable

Mini-DIN8-Male	Signal	DB-25-Male	Signal
1	Data Terminal Ready	20	Data Terminal Ready
2	Clear to Send*	5	Clear to Send*
3	Transmit Data	2	Transmit Data
4	Signal Ground	7	Signal Ground
5	Receive Data	3	Receive Data
6	Request to Send*	4	Request to Send*
7	Data Carrier Detect	8	Data Carrier Detect
8	Signal Ground	7	Signal Ground

* RTS and CTS are ignored (optional) if using */dev/ttym** but required if using */dev/ttyf**

Note: This cable is available from Silicon Graphics. Contact your sales representative. This cable can be used with a null modem adapter for terminals and printers (see Table 1-5); however, you should use this cable exclusively for modem connections. The commercially available cable that connects a Macintosh SE to a modem does not work properly with Silicon Graphics software.