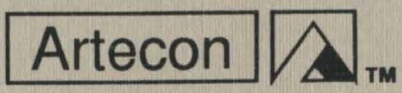
ArtePort Software
User's Manual
Version 1.6

for
SunOS 4.1.2, 4.1.3
and
5.1, 5.2 and 5.3(Solaris 2.1, 2.2 and 2.3)



ArtePort Software User's Manual

Version 1.6

for
SunOS 4.1.2, 4.1.3
and
5.1, 5.2 and 5.3
(Solaris 2.1, 2.2 and 2.3)

Artecon Inc.
P.O. Box 9000
Carlsbad, CA 92018-9000
(619) 931-5500

P/N 83-53011703 Rev 1.1-12/93

Standard Warranty Information

Artecon Standard Limited Warranty is a Return To Factory Warranty. Artecon reserves the right to repair, replace, or upgrade at our option with new or reconditioned product which is in good working order; which has been cleaned, lubricated adjusted and tested; and which has had defective or excessively worn parts replaced with new or serviceable used parts.

If one of our products operates improperly during the limited warranty period, we will service it free of charge, provided the complete product is shipped to our designated Service Center at the Customer's expense. This includes all brokerage, customs, handling and insurance charges which may be incurred during the shipping process. The complete defective product must be shipped in the original shipping container. Improper packaging may result in damage which is not covered under the limited warranty and will subsequently void said product limited warranty.

In the event of a System or Subsystem malfunction the customer or his agent should contact the Technical Support Department of Artecon for problem diagnosis. If the Technical Support Representative determines that the problem cannot be resolved on-site a Return Maintenance Authorization (RMA) will be issued.

Artecon will not accept any parcel without a valid RMA number printed on both the exterior of the Artecon shipping container, and on the shipping label. Please include a brief description of the problem encountered, in writing, along with the name of your firm's pertinent technical person.

Artecon reserves the right to replace defective product with new or reconditioned product prior to receiving said defective product. The customer must return the defective product within 10 days of receipt of the replacement product. The customer will be invoiced for all appropriate charges and upon receipt will be credited for the amount invoiced.

Table of Contents

List of Figures

List of Tables

Overview

ArtePort software includes an OpenWindows GUI software package that lets users easily connect any serial or parallel device to SPARC systems. The software provides all the functions of the UNIX *stty* command and additional features not found in *stty*. ArtePort software also includes a STREAMS-based driver that interfaces between the kernel and the Artecon SBus boards.

This document describes the installation and usage of the Artecon ArtePort Software for Sun Workstations. The purpose of this manual is to provide instructions on how to install and use the ArtePort Software with SBus serial/parallel boards attached to a Sun Workstation.

This document contains:

- **Chapter 1** describes Installing the ArtePort Software.
- **Chapter 2** describes Using ttytool Utility Programs.
- **Chapter 3** describes Configuration.
- **Chapter 4** describes Troubleshooting.
- **Appendix A** describes RS232 Signals.
- **Appendix B** describes Sample Printcap Entries.
- **Appendix C** describes Parallel Printcap.
- **Appendix D** describes Modem Setup.
- **Appendix E** describes Solaris Modems.

Chapter 1.0 Installing the ArtePort Software

ArtePort Version 1.6 is now supported on the following Sun Architecture/OS configurations

Table 1-1. Sun Architecture/OS Configurations

Architecture	SunOS Version
sun4c	4.1.2 - 4.1.3 and 5.1 - 5.3 (Solaris 2.1-2.3)
sun4m	4.1.2 - 4.1.3 and 5.1 - 5.3 (Solaris 2.1-2.3)
sun4d	5.1- 5.3 (Solaris 2.1-2.3)

1.1 Description

The ArtePort Software consists of both a device driver and an OpenWindows GUI utility (**ttytool**). ArtePort uses a loadable device driver so that no kernel reconfiguration is necessary. Please refer to **Chapter 2** for more information on **ttytool**.

ArtePort is released to the customer as a single QIC tape or as a 3.5" high density floppy disk containing binaries for relevant platforms.

1.1.1 SunOS 4.1.2 - 4.1.3 (Solaris 1.X) Media:

The table of contents of the Customer Release Tape is as follows

Table 1-2. Customer Release Tape Table of Contents:

Tape File	Format	Description
1	dd	Copyright
2	tar	CMX, README, manpages and install script
3	tar	ArtePort binaries

The Customer Release Floppy contains all of the above files in a single bar(1) file.

1.1.2 SunOS 5.1 - 5.3 (Solaris 2.1 - 2.3)

ArtePort for Solaris 2.1 - 2.3 is released in a **package** format. Refer to the Sun Manual Application Packaging and Installation Guide for more information on this format.

1.2 Software Installation .

*The ArtePort SBus card **should** be installed before the ArtePort software installation begins.*

Prior to installing the SBus Card, FIRMLY seat ALL socketed CHIPS!
Place board on a flat, ESD-safe work area with connectors hanging over the edges (and while wearing your ESD wrist strap) FIRMLY push down on the big, square Cirrus chips.

Become super user and start the installation.

```
# su root
```

During installation, the script displays the default values within square brackets. The script accepts the default value when you press the <Return> key. If you decide to choose an answer other than the default value, you need to type in the answer at the prompt and press the <Return> key.

.2.1 SunOS 4.1.2 - 4.1.3 (Solaris 1.X)

```
# extract_unbundled
```

If you are installing the software using an Artecon floppy drive, you must first extract the new version of the "extract_unbundled" command from the release floppy diskette. This version of "extract_unbundled" will recognize the Artecon floppy drives in addition to the devices supported by the normal "extract_unbundled (8) command.

 # bar xvfZT /dev/afd<unit> extract_unbundled

***Where** <unit> is the unit number of the floppy drive (i.e. 0, 1, 2, or 3).*

 # ./extract_unbundled

This installation script will query you for the drive location and device name. It then automatically extracts and installs the correct files for your system.

1.2.2 SunOS 5.1 -5.3 (Solaris 2.1-2.3)

 # pkgadd -d /dev/<device>

where <device> is the device you are installing from (e.g. `rfd0a` when installing from floppy, `rmt/0` when installing from tape)

The installation script will ask you where you would like the different files to be placed and then installs them.

1.3 Using the ArtePort Software

The ArtePort device driver is now loaded and ready for use. If you are running SunOS 4.1.X, the system does not need to be rebooted (assuming the ArtePort SBus card has already been installed). If you are using Solaris 2.X, do a boot *-r*.

Refer to **Chapter 2** for information on using **ttytool**, an OpenWindows GUI used to help configure ArtePort serial ports.

The ArtePort device names are shown in **Table 1-3**.

Table 1-3. ArtePort Device Names

	Dial-in Device	Dial-out Device
SunOS 4.1.2 - 4.1.3	/dev/tty<x><y>	/dev/cu<x><y>
Where <x> is the ArtePort board: A=1st, B=2nd, C=3rd, etc. Where <y> is the Port number: 0, 1, 2...f		
SunOS5.1 - 5.3	/dev/term/<x>	/dev/cua/<x>
Where <x> is the Port number: 0, 1, 2,...		

For dial-in devices (ie device on A<y>), the driver waits to see carrier detect (CD) go high before responding. Because of this, if you open a port and CD is not (or does not go) high, the port is hung.

Therefore, we suggest using cuA<y> to test the ports.

Chapter 2.0 Using the ttytool Utility Program

2.1 Introduction

ttytool features a graphical user interface (GUI) designed to allow the user to easily test serial or parallel devices connected to a system. It gives complete control over the terminal interface and provides a terminal window to communicate with the new device. **ttytool** provides all the functions of the UNIX *stty(1)* command and additional features not found in *stty(1)*. The information and parameters are in a clear and easy to understand format.Before trying to start **ttytool**, ensure that the following lines are near the top of your. *.cshre* file (or .login for Solaris 2.X),

setenv OPENWINHOME=/use/openwin.

setenv LD_LIBRARY_PATH $OPENWINHOME/lib:/usr/lib.

2.2 ttytool - GUI

You can bring up **ttytool** by typing the following command in an OpenWindows Shell Tool or Command Tool:

```
% ttytool
```

When first run, it will display a window showing a set of default parameter settings, as shown in **Figure 2-1**. By entering a device name in the "Device:" field and clicking on "Get", the program will read the current device settings and display them. The device parameters will not be changed until the "Apply" button is pressed.

For example:

SunOS 4.1.2 - 4.1.3 enter `/dev/cuA3`

SunOS5.1 - 5.3 enter `/dev/term/cua/3`

If CD is not high the port will hang. We suggest using cuA (instead of ttya) for testing purposes and/or determining the desired parameter settings.

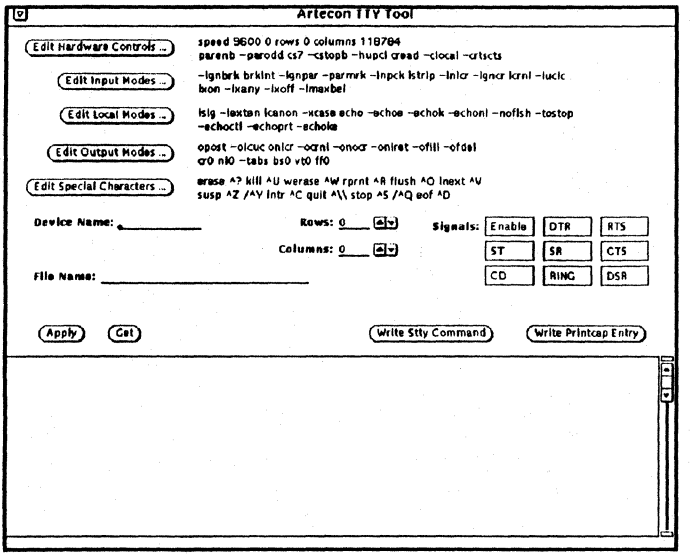


Figure 2-1. ttytool Graphical User Interface

The bottom terminal sub-window is connected to the device. The information typed in the window is sent to the device. Also, the information from the device (subject to line buffering) is sent to the window.

When the **ttytool** window is closed, the GUI will disappear from the screen and an icon will appear, as shown in **Figure 2-2**. But the port that was opened is still attached to it. To close the port, select quit.

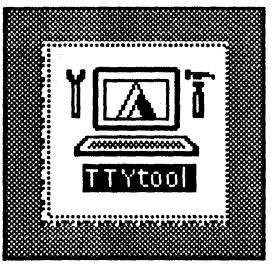


Figure 2-2. ttytool Icon

Pop-up windows allow detailed display and editing of all parameters. The following pop-up windows are:

- Edit Hardware Controls

- Edit Input Modes
- Edit Local Modes
- Edit Output Modes
- Edit Special Characters

2.2.1 Edit Hardware Controls

These options and their default settings are shown in **Figure 2-3.** Refer to **Table 2-1** for Hardware Control information.

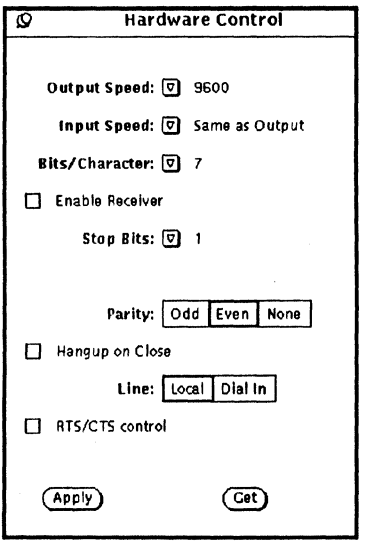


Figure 2-3. Hardware Control

Table 2-1. Editing Hardware Controls

Options	Description
Output Speed (cbaud)	Baud rate.
Input Speed (cibaud)	Input baud rate, if different from output rate.
Bits/Character (csize)	Character size.
Enable Receiver (cread)	Enable Receiver.
Stop Bits (cstopb)	Send one or two stop bits.
Parity (parenb, parodd)	Parity enable (odd, even, or none)
Hang-up on Close (hupcl)	Hang-up on last close.
Line (clocal)	Local line or Dial in.
RTS/CTS Control (crtscts)	Enable RTS/CTS flow control.

The CBAUD bits specify the baud rate.

The zero baud rate is used to hang up the connection. If it is specified, the modem control lines will cease to be asserted. Normally, this will disconnect the line. If the CIBAUD bits are not zero, they specify the input baud rate, with the CBAUD bits specifying the output baud rate; otherwise, the output and input baud rates are both specified by the CBAUD bits. The values for the CIBAUD bits are the same as the values for the CBAUD bits, shifted left IBSHIFT bits. For any particular hardware, impossible speed changes are ignored.

The CSIZE bits specify the character size in bits for both transmission and reception. This size does not include the parity bit, if any. If CSTOPB is set, two stop bits are used, otherwise one stop bit. For example, at 110 baud, two stop bits are required.

If PARENB is set, parity generation and detection is enabled and a parity bit is added to each character. If parity is enabled, the PARODD flag specifies odd parity if set, otherwise even parity is used.

If CREAD is set, the receiver is enabled. Otherwise no characters will be received.

If HUPCL is set, the modem control lines for the port will be disconnected when the last process with the line open closes it or terminates.

If CLOCAL is set, a connection does not depend on the state of the modem status lines. Otherwise modem control is assumed.

If CRTSCTS is set, and the terminal has modem control lines associated with it, the Request To Send (RTS) modem control line will be raised, and output will occur only if the Clear To Send (CTS) modem status line is raised. If the CTS modem status line is lowered, output is suspended until CTS is raised. Some hardware may not support this function, and other hardware may not permit it to be disabled; in either of these cases, the state of the CRTSCTS flag is ignored.

The initial hardware control value after open is CS7, CREAD, PARENB.

2.2.2 Edit Input Modes

These options and their default settings are shown in **Figure 2-4**. Refer to **Table 2-2** for Input Mode information.

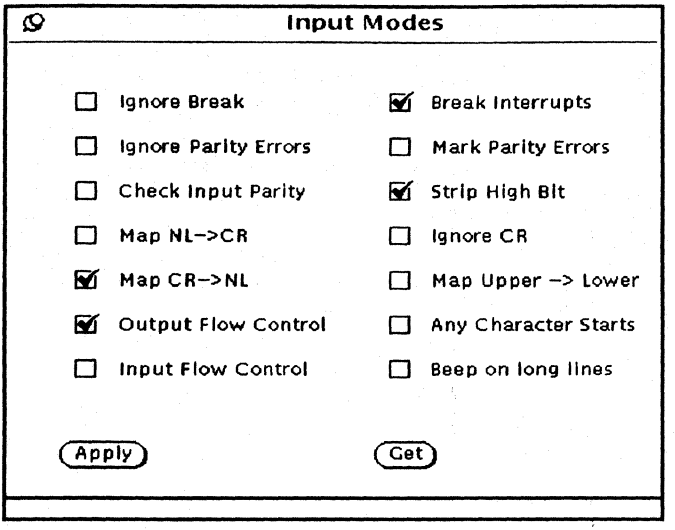


Figure 2-4. Input Modes

Table 2-2. Editing Input Modes

Options	Description
Ignore Break	Ignore break condition.
Ignore Parity Errors	Signal interrupts on break.
Check Input Parity	Enable characters with parity errors.
Map NL-CR	Map NL to CR on input.
Map CR-NL	Map CR to NL on input.
Output Flow Control	Enable start/stop output flow control.
Input Flow Control	Enable start/stop input flow control.
Break Interrupts	Signal interrupt on break.
Mark Parity Errors	Mark parity errors.
Strip High Bit	Strip character.
Ignore CR	Ignore CR

Options	Description
Map Upper- Lower	Map upper-case to lower-case on input.
Any Character Starts	Enable any character to restart output.
Beep on long lines	Echo BEL on input line too long.

If IGNBRK is set, a break condition (a character framing error with data all zeros) detected on input is ignored, that is, not put on the input queue and therefore not read by any process.Otherwise, if BRKINT is set, a break condition will generate a SIGINT and flush both the input and output queues.If neither IGNBRK nor BRKINT is set, a break condition is read as a single ASCII NUL character (0).

If IGNPAR is set, characters with framing or parity errors (other than break) are ignored. If neither IGNPAR nor PARMRK is set, a framing or parity error (other than break) is read as a single ASCII NUL character (0).

If INPCK is set, input parity checking is enabled. If INPCK is not set, input parity checking is disabled. This allows output parity generation without input parity errors.

If ISTRIP is set, valid input characters are first stripped to 7 bits, otherwise all 8 bits are processed.

If INLCR is set, a received NL character is translated into a CR character.

If IGNCR is set, a received CR character is ignored (not read). Otherwise if ICRNL is set, a received CR character is translated into a NL character.

If IUCLC is set, a received upper-case alphabetic character is translated into the corresponding lower-case character.

If IXON is set, start/stop output control is enabled. A received STOP character will suspend output and a received START character will restart output.The STOP and START characters will not be read, but will merely perform flow control functions. If IXANY is set, any input character will restart output that has been suspended.

If IXOFF is set, the system will transmit a STOP character when the input queue is nearly full, and a START character when enough input has been read that the input queue is nearly empty again.

If IMAXBEL is set, the ASCII BEL character is echoed if the input stream overflows. Further input will not be stored, but any input already present in the input stream will not be disturbed.

If IMAXBEL is not set, no BEL character is echoed, and all input present in the input queue is discarded if the input stream overflows. The initial input control value is BRKINT, ICRNL, IXON, ISTRIP.

2.2.3 Edit Local Modes

These options and their default settings are shown in **Figure 2-5**. Refer to **Table 2-3** for Local Mode information.

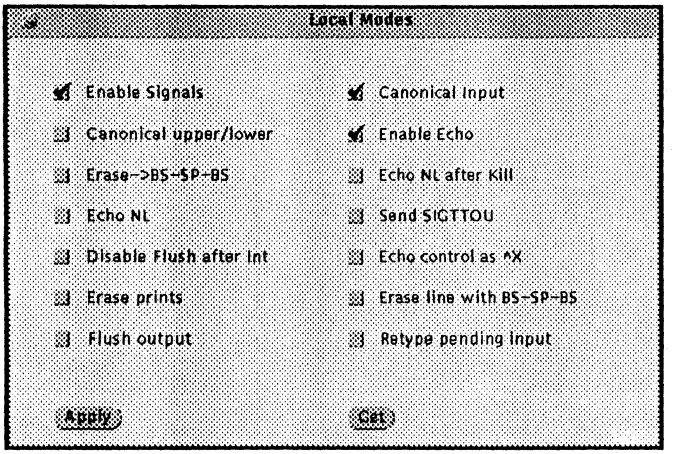


Figure 2-5. Local Modes

Table 2-3. Editing Local Modes

Options	Description
Enable Signals (isig)	Enable Signals.
Canonical upper/ lower (xcase)	Canonical upper/lower presentation.
Erase-BS-SP-BS (echoe)	Echo erase character as backspace (BS)-space(SP)-backspace (BS).
Echo NL (echonl)	Echo NL, even if echo is not set.
Disable Flush after Int (no fish)	Disable flush after interrupt or quit.
Erase prints (echoprt)	Echo erase character as character erased.
Flush output (flusho)	Output is being flushed.
Canonical Input (Icanon)	Canonical input (erase and kill processing).
Enable Echo (echo)	Enable echo.
Echo NL after Kill (echoke)	Echo NL after kill character.
Send SIGTTOU (tostop)	Send SIGTTOU for background output. Stop background jobs that attempt to write to the terminal.
Echo control as ^X (echoctl)	Echo control characters as ^char, delete as ^?.
Erase line with BS-SP-BS (echok)	BS-SP-BS erase entire line on line kill.
Retype pending input (pending)	Retype pending input at next read or input character.

If ISIG is set, each input character is checked against the special control characters INTR, QUIT, and SUSP. If an input character matches one of these control characters, the function associated with that character is performed. If ISIG is not set, no checking is done. Thus these special input functions are possible only if ISIG is set.

If ICANON is set, canonical processing is enabled. This is affected by the IEXTEN bit (see Special Characters above). This enables the erase, word erase, kill, and reprint edit functions, and the assembly of input characters into lines delimited by NL, EOF, EOL, and EOL2. If ICANON is not set, read requests are satisfied directly from the input queue. A read will not be satisfied until at least MIN characters have been received or the timeout value TIME has expired between characters. This allows fast bursts of input to be read efficiently while still allowing single character input.

The time value represents tenths of seconds. See the Non-canonical Mode Input Processing section for more details.

If XCASE is set, and if ICANON is set, an upper-case letter is accepted on input by preceding it with a character, and is output preceded by a character.

If ECHO is set, characters are echoed as received.If ECHO is not set, input characters are not echoed.

If ECHOCTL is not set, all control characters (characters with codes between 0 and 37 octal) are echoed as themselves. If ECHOCTL is set, all control characters other than ASCII TAB, ASCII NL, the START character, and the STOP character, are echoed as ^ X.

When ICANON is set, the following echo functions are possible:

If ECHO and ECHOE are set, and ECHOPRT is not set, the ERASE and WERASE characters are echoed as one or more ASCII BS SP BS, which will clear the last character(s) from a CRT screen.

If ECHO and ECHOPRT are set, the first ERASE and WERASE character in a sequence echoes as a backslash (\) followed by the characters being erased. Subsequent ERASE and WERASE characters echo the characters being erased, in reverse order. The next non-erase character types a slash (/) before it is echoed.

If ECHOKE is set, the kill character is echoed by erasing each character on the line from the screen (using the mechanism selected by ECHOE and ECHOPRT).

If ECHOK is set, and ECHOKE is not set, the NL character will be echoed after the kill character to emphasize that the line will be deleted. Note: an escape character or an LNEXT character preceding the erase or kill character removes any special function.

If ECHONL is set, the NL character will be echoed even if ECHO is not set. This is useful for terminals set to local echo (so-called half duplex).

If ECHOCTL is not set, the EOF character is not echoed, unless it is escaped. Because EOT is the default EOF character, this prevents terminals that respond to EOT from hanging up. If ECHOCTL is set, the EOF character is echoed; if it is not escaped, after it is echoed, one backspace character is output if it is echoed as itself, and two backspace characters are echoed if it is echoed as ^ X.

If NOFLSH is set, the normal flush of the input and output queues associated with the INTR, QUIT, and SUSP characters will not be done.

If TOSTOP is set, the signal SIGTTOU is sent to a process that tries to write to its controlling terminal if it is not in the distinguished process group for that terminal. This signal normally stops the process. Otherwise, the output generated by that process is output to the current output stream. Processes that are blocking or ignoring SIGTTOU signals are excepted and allowed to produce output.

If FLUSHO is set, data written to the terminal will be discarded. This bit is set when the FLUSH character is typed. A program can cancel the effect of typing the FLUSH character by clearing FLUSHO.

If PENDIN is set, any input that has not yet been read will be reprinted when the next character arrives as input.

The initial line-discipline control value is ISIG, ICANON, ECHO.

2.2.4 Edit Output Modes

These options and their default settings are shown in **Figure 2-6.** Refer to **Table 2-4** for Output Mode information.

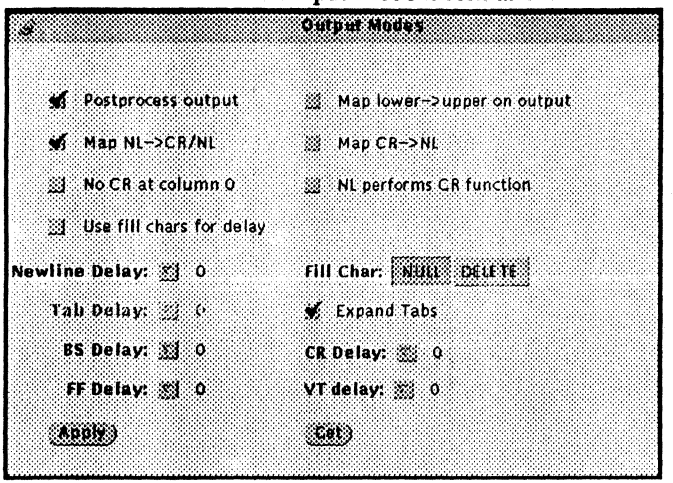


Figure 2-6. Output Modes

Table 2-4. Editing Output Modes

Options	Description
Postprocess Output	Postprocess output.
Map NL-CR/NL	Map new line (NL) to carriage return (CR) on output.
No CR at column 0	No CR output at column 0.
Use fill chars for delay	Use fill characters for delay.
Newline Delay	Select new-line delays.
BS Delay	Select backspace delays.
FF Delay	Select form-feed delays
Map lower- upper on output	Map lower case to upper case on output.
Map CR-NL	Map CR to NL on output.
NL performs CR function	NL performs CR function.
Fill Char	Fill is DEL, else is NUL.
Expand Tabs	Expand tabs to spaces.
CR Delay	Select carriage-return delays.
VT Delay	Select vertical tab delays.

If OPOST is set, output characters are post-processed as indicated by the remaining flags, otherwise characters are transmitted without change.

If OLCUC is set, a lower-case alphabetic character is transmitted as the corresponding upper-case character. This function is often used in conjunction with IUCLC.

If ONLCR is set, the NL character is transmitted as the CR-NL character pair. If OCRNL is set, the CR character is transmitted as the NL character. If ONOCR is set, no CR character is transmitted when at column 0 (first position). If ONLRET is set, the NL

character is assumed to do the carriage-return function; the column pointer will be set to 0 and the delays specified for CR will be used. Otherwise the NL character is assumed to do just the line-feed function; the column pointer will remain unchanged. The column pointer is also set to 0 if the CR character is actually transmitted.

The delay bits specify how long transmission stops to allow for mechanical or other movement when certain characters are sent to the terminal. In all cases a value of 0 indicates no delay. If OFILL is set, fill characters will be transmitted for delay instead of a timed delay. This is useful for high baud rate terminals that need only a minimal delay. If OFDEL is set, the fill character is DEL, otherwise NUL.

If a form-feed or vertical-tab delay is specified, it lasts for about 2 seconds.

New-line delay lasts about 0.10 seconds.If ONLRET is set, the RETURN delays are used instead of the NEWLINE delays. If OFILL is set, two fill characters will be transmitted.

Carriage-return delay type 1 is dependent on the current column position, type 2 is about 0.10 seconds, and type 3 is about 0.15 seconds. If OFILL is set, delay type 1 transmits two fill characters, and type 2, four fill characters.

Horizontal-tab delay type 1 is dependent on the current column position. Type 2 is about 0.10 seconds. Type 3, specified by TAB3 or XTABS, specifies that TAB characters are to be expanded into SPACE characters. If OFILL is set, two fill characters will be transmitted for any delay.

Backspace delay lasts about 0.05 seconds. If OFILL is set, one fill character will be transmitted. The actual delays depend on line speed and system load.

The initial output control value is OPOST, ONLCR, XTABS.

2.2.5 Edit Special Characters

Special characters have special functions on input and/or output. These options and their default settings are shown in **Figure 2-7**. Refer to **Table 2-5** for Special Characters information.

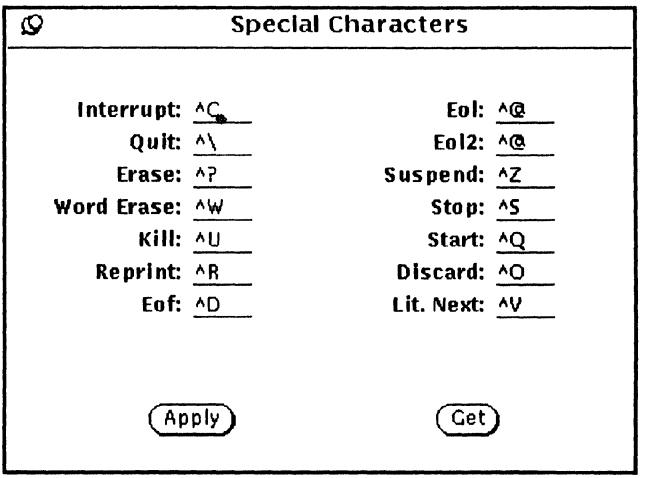


Figure 2-7. Special Characters

Table 2-5. Editing Special Characters

Options	Description
Interrupt	(CTRL-C or ASCII ETX) generates a SIGINT signal, which is sent to all processes in the distinguished process group associated with the terminal. Normally, each such process is forced to terminate the program, but arrangements may be made either to ignore the signal or to receive a trap to an agreed-upon location; see sigvec(2).
Quit	(CTRL-\or ASCII FS) generates a SIGQUIT signal, which is sent to all processes in the distinguished process group associated with the terminal. Its treatment is identical to the interrupt signal except that, unless a receiving process has made other arrangements, it will not only be terminated but a core image file (called core will be created in the current working directory.
Erase	(Rubout or ASCII DEL) erases the preceding character. It will not erase beyond the start of a line, as delimited by a NL, EOF, EOL, or EOL2 character.

Options	Description
Word Erase	(CTRL-W or ASCII ETB) erases the preceding word. It will not erase beyond the start of a line, as delimited by a NL, EOF, EOL, or EOL2 character.
Kill	(CTRL-U or ASCII NAK) deletes the entire line, as delimited by a NL, EOF, EOL, or EOL2 character.
Reprint	(CTRL-R or ASCII DC2) reprints all characters that have not been read, preceded by a NEWLINE.
Eof	(CTRL-D or ASCII EOT) may be used to generate an end-of-file from a terminal. When received, all the characters waiting to be read are immediately passed to the program, without waiting for a NEW-LINE, and the EOF is discarded. Thus, if there are no characters waiting, which is to say the EOF occurred at the beginning of a line, zero characters will be passed back, which is the standard end-of-file indication.
Eol	(ASCII LF) is the normal line delimiter. It can not be changed; it can, however, be escaped by the Lit. Next character.
Eol2	(ASCII NUL) is an additional line delimiters, like NL. It is not normally used.
Suspend	(CTRL-Z or ASCII SUB) is used by the job control facility to change the current job to return to the controlling job. It generates a SIGTSTP signal, which stops all processes in the terminal's process group.
Stop	(CTRL-S or ASCII DC3) can be used to temporarily suspend output. It is useful with CRT terminals to prevent output from disappearing before it can be read. While output is suspended, STOP characters are ignored and not read.

Options	Description
Start	(CTRL-Q or ASCII DC1) is used to resume output that has been suspended by a STOP character. While output is not suspended, START characters are ignored and not read.
Discard	(CTRL-O or ASCII SI) causes subsequent output to be discarded until another DISCARD character is typed, more input arrives, or the condition is cleared by a program.
Lit.Next	(CTRL-V or ASCII SYN) causes the special meaning of the next character to be ignored; this works for all the special characters mentioned above. This allows characters to be input that would otherwise get interpreted by the system (Example: KILL, QUIT)

The character values for Interrupt, Quit, Erase, WErase, Kill, Reprint, Eof, Eol, Eol2, Suspend, Stop, Start, Discard, and Lit.Next may be changed to suit individual tastes. If the value of a special control character is 0, the function of that special control character will be disabled. The Erase, Kill, and Eof characters may be escaped by a preceding character, in which case no special function is done. Any of the special characters may be preceded by the Lit.Next character, in which case no special function is done. If IEXTEN is added to the local modes (this is the default), then all of the special characters are in effect. If IEXTEN is cleared from the local modes, then only the following POSIX.1 compatible specials are seen as specials: Interrupt, Quit, Erase, Kill, Eof, Nl, Eol, Suspend, Stop, Start, and Cr.

2.2.6 Device Name Field

Type in the name of the ArtePort port you would like to work with.

Example:

SunOS 4.1.2 - 4.1.3: /dev/cuA0

(for SB-300P, use /dev/cuAd)

First port on the first ArtePort board.

SunOS5.1 - 5.3: /dev/cua/1

First port on the first ArtePort board.

(cua/0 is the parallel port for all except the SB-1600)

2.2.7 Rows and Columns Fields

Rows and **Columns** correspond to the size of the terminal's screen. These numbers are used by a number of screen oriented programs such as vi(1).

2.2.8 Signals Field

Each of the fields in this section correspond to a signal on the port. Not all ports support all signals. Clicking on "Get" button will cause the current status of the signals to be displayed. Some of the signals may be set. Clicking on the "Apply" button will cause **ttytool** to set the signals according to the values indicated. The signals will then be read and current values displayed.

2.2.9 File Name Field

Type in the name of a file which will hold the output of the Write **stty** Command or Write Printcap Entry functions.

2.2.10 Apply Button

Click this button to apply any changes you have made in **ttytool**. The changes will be made to the port currently shown in the Device Name Field.

2.2.11 Get Button

Click this button to get the current settings of the port shown in the Device Name Field. The current settings will be reflected in the different **ttytool** windows.

2.2.12 Colors Button

Change colors of buttons and other items within the ttytool screen. The color button will only appear on a color monitor. **Figure 2-1** illustrates an example of a ttytool on a monochrome monitor.

2.2.13 Write *stty* Command

Writes a *stty* command script to the file in "File Name". This can be executed to set the port's values.

2.2.14 Write Printcap Entry

Writes a sample printcap entry to the file in "File Name". This may be easily edited into a real printcap entry thus bypassing all the bit twiddling usually associated with configuring serial printers

This button is not present in the SunOS 5.1-5.3 (Solaris 2.1-2.3) version of ttytool. SunOS 5.1 - 5.3 does not use a printcap file.

2.3 Help Windows

Pop-up windows that allow you to view on-line help are called Help windows. You display a Help window by moving the pointer to the object for which you want help and pressing the <Help> key on the keyboard. The <F1> key can also be used as the Help key. A pinned Help window is displayed, as shown in the example in **Figure 2-8**. The object (or a portion of it if the object is large) at the pointer location is displayed in the magnifying glass of the Help window along with explanatory help text.

*If no help is displayed, enter the following command at the system prompt before executing "**ttytool**":*

setenv HELPPATH $OPENWINHOME/lib/help

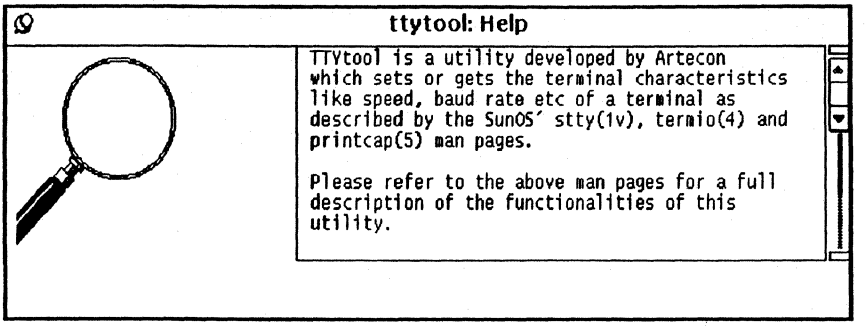


Figure 2-8. Example of Help Window

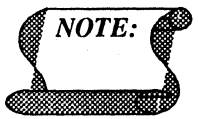
Chapter 3.0 Configuration

3.1 Overview

Teletypes consisted of an upper-case-only printer, a keyboard and a paper tape reader/punch.

When computers came along, the Teletype became a cheap and readily available terminal. It gave the programmer access to a keyboard and printer. It also gave him a paper tape reader/punch for program storage. Almost every computer had a RS232 serial interface for hooking up a Teletype.

The RS232 standard uses the term "Data Terminal Equipment' (DTE) for a Teletype-like device and "Data Communication Equipment" (DCE) for modems. The ArtePort device uses the Sun standard of DTE. Standards concerning DTE's and DCE's are confusing.



Throughout this chapter the term "computer" refers to the ArtePort serial ports.

3.2 Serial Connections Made Simple

Connecting a terminal, modem, or serial printer is a simple procedure. The RS232 standard defines the serial interface. This section will give you some background and practical advice to help you easily connect serial devices to your ArtePort card.

3.3 DCE or DTE

The basic tool for debugging communications problems is the breakout box. It is a small box with several LED's that show the status of the important communication signals, as shown in **Figure 3-1.**

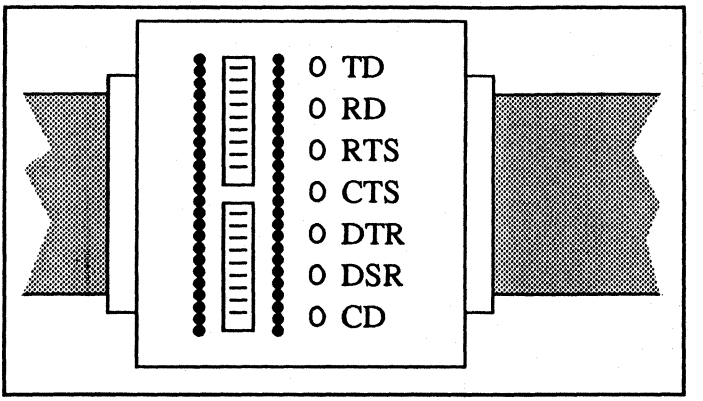


Figure 3-1. DCE or DTE

Sun systems are wired up as DTEs. A DCE-type device can be wired indirectly. If you want to connect a DTE you must use a null modem cable, refer to *Section 3.4.3*.

To determine if you have a DCE or DTE, connect one end of your breakout box to the device. Leave the other end unconnected. If LED #2 (TD) lights you have a DTE. If LED #3 (RD) lights you have a DCE. If both LEDs light, you have a broken device. If no LEDs light, you have a cable problem or a broken device.

3.4 Physical Connection

The first step in connecting a terminal, printer, or modem to your computer is to string a cable between the two devices. There are two types of connections, the three wire connection and full modem control.

3.4.1 Three Wire Cable

Three wire uses a light cheap cable. No modem control signals are passed between the terminal and the computer. Turning off the terminal does not log you off the computer. This type of connection should not be used for modems. There are two types of three wire connections: straight-through and with a twist. The straight-through cable is shown in **Figure 3-2.**

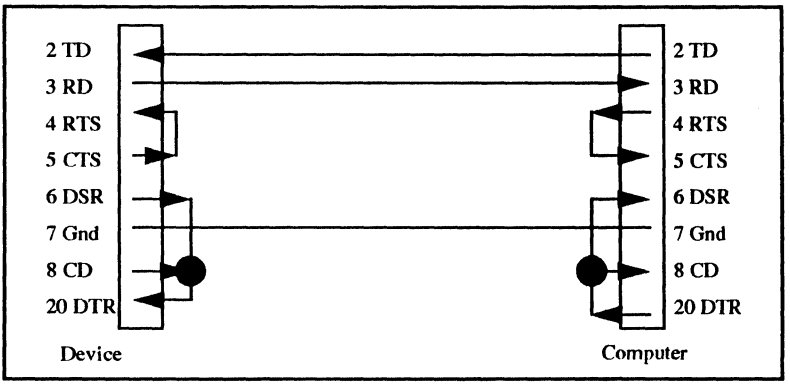


Figure 3-2. Straight Through Cable

A three wire with a twist contains a built in null modem, as shown in Figure 3-3.

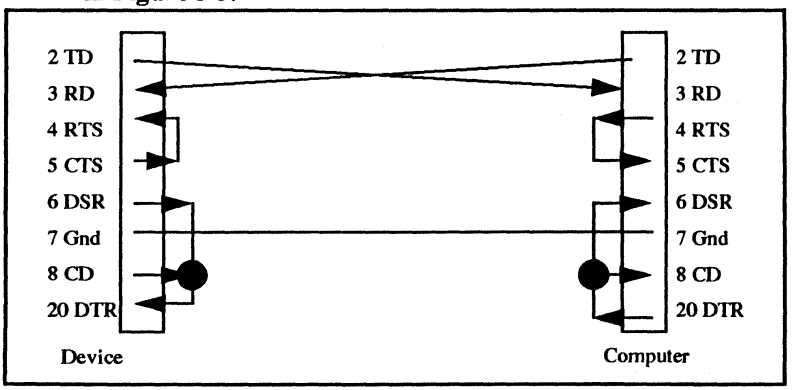


Figure 3-3. Twist Cable

To determine if you need a straight-through cable or one with a twist, put your breakout box on the computer and see which data signal (2 or 3) is used. Then check the device in a similar manner. If both devices use the same data signal, you need a twist cable. If different data signals are used, then a straight-through cable is needed.

3.4.2 Full Modem Cable

The full modem cable passes all seven signals from the computer to the device. The cable must contain 8 wires (one for ground). The advantage of this connection is to allow you to connect a modem, or other devices that require full modem control, to the computer. The wiring diagram for a full modem cable is shown in **Figure 3-4.**

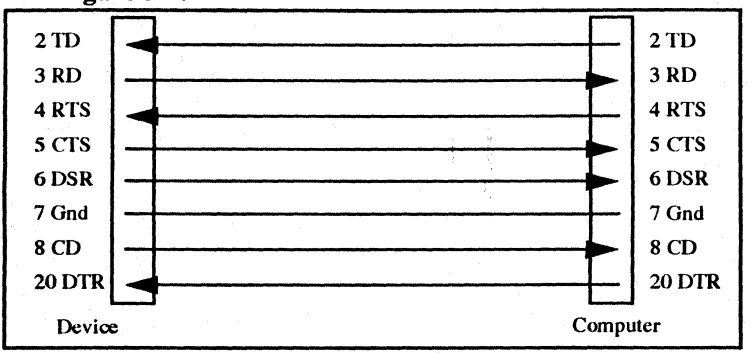


Figure 3-4. Full Modem Cable

3.4.3 Null Modem Cable

If the device you are adding is a DTE, you will need a null modem or the three wire "twist" cable described in Section 3.4.1. A null modem will switch certain signals so that a DTE's signals will look similar to a DCE's signals. The wiring diagram for a null modem cable is shown in Figure 3-5.

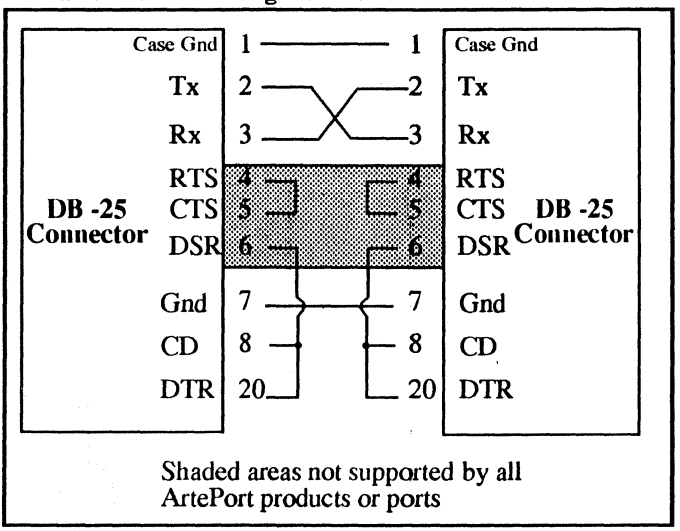


Figure 3-5. Null Modem Cable

3.5 Communications Parameters

After the device is connected, set the communications parameters. These include speed (baud rate), data bits, parity, and stop bits. The program **ttytool** is useful in getting the communications parameters. Run the program and set the Device Name to whichever port you are trying to connect. Click on the "Get" button to establish the device. Now try setting them according to the information in your device manual.

The safest settings are shown in **Table 3-1.**

Table 3-1. Communication Parameters

Settings	Parameters
Data Bits	8
Parity	None
Stop Bits	2

Try various speeds until the device responds correctly. Common speeds are 1200, 2400, 9600 and 300. After setting the parameters, click on the "Apply" button (any one, they are all the same) and try typing text into the terminal subwindow at the bottom of the ttytool main window. What you type, should appear on the device. Some devices, like printers, must receive an entire line before they print anything.

3.5.1 Parity

Parity is a primitive error detection method. Standard ASCII characters contain 7 bits of data. In even parity, an extra bit is added to make the total number of bits come out even.

If you send no parity data to an even parity device, some of the characters will get parity errors. If some of the data gets through, but some does not, then you probably have a parity problem.

Some printers use the high bit to indicate graphics or italic characters. So if your output appears as: "αβcδefγη" or "*abcdefgh*". Parity should be set to none and the characters size to eight.

3.5.2 Data bits

The safest setting for most devices is 8 bit data. Sending 8 bit data to a 7 bit device is not a significant problem. The computer sends 8 bits, the device reads 7, throws away 1 and works about 10% slower.

3.5.3 Stop bits

Stop bits are used to signal the end of a character much like a space is used to signal the end of a word. Extra stop bits allows extra space between characters and will work slower.

3.5.4 Flow control

Computers can send characters faster than modems can receive and re-transmit them. Flow control is used to tell the computer to slow down. If the flow control is not set up correctly, the modem can hang the port or overwrite data previously received.

The two main flow control methods are "XON/XOFF" and "RTS/CTS."

If you are using XON/XOFF you **cannot** *use the Xmodem, Ymodem or Zmodem file transfer protocols. These protocals can have the XON or XOFF character as part of the packet header.*

XON/XOFF are two control characters that are a holdover from the old Teletype days. They are used to control data being sent by the computer. The modem will send the XOFF character to indicate that it can not handle any more data. When it's able to handle additional data, it will send an XON character.

RTS stands for Request To Send. RTS is a signal generated by the computer to indicate that it's ready to accept data. CTS stands for Clear To Send. CTS is generated by the device to indicate that it's ready for data. Some printers misuse DTR as a type of CTS signal.

For printers that do not support RTS/CTS, connect the printer's DTR line (20) to the ArtePort's CD line (8). Since Artecon's driver monitors Carrier Detect (CD), when CD goes low (ie printer dropped DTR) the port stops sending data. When CD goes high again, transmission is resumed.

3.5.5 Hints

- When starting out, use the `/dev/cuA<x>` (or `/dev/cua/<x>` for Solaris 2.1) entry instead of the typical `/dev/ttyA<x>` (or `/dev/term/<x>` for Solaris 2.1). The second entry will wait for carrier to come up before opening the device. If there is a problem with carrier this will freeze your program. Get data going back and forth from the device before worrying about the modem control signals.

- When setting up a new type of terminal or printer for the first time, try it out next to the computer. You can change setup parameters or dip switches on the device quickly and not have to run into the next room every time you want to try something new.
- Some computer installations have "lost cables". That happens when there is a bunch of cables going from the computer into a wall and out in some remote office.

  When confronted by a situation like this, short pins 2 and 3 of one of the remote cables. Then go into the computer room and use a ohm meter to find the cable with the shorted pins. Now label the cable and repeat the process.

3.5.6 Example of Printer Installation

The following examples are only useful for SunOS 4.1.2 - 4.1.3 users. SunOS 5.1 - 5.2 (Solaris 2.1 - 2.3) uses a different facility for configuring printers. Refer to the Sun document which covers configuring printers for more details.

We will explain how to set up a serial printer on line, as shown in **Figure 3-6.** Printers are generally more difficult to set up than terminals.

We unpack the printer, plug it in and turn it on. (We do not connect it to the computer yet.)

We attach a full modem (straight through) cable to the printer and plug it into a breakout box. LED 2 (TD) lights indicating that the printer is wired as a DTE. Since the computer is also wired as a DTE we will need a null modem.

We connect the printer to the computer using a full modem cable and a null modem cable.

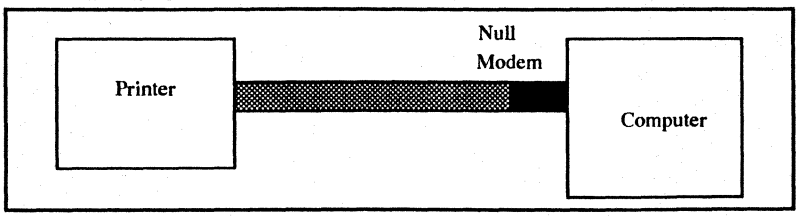


Figure 3-6. Serial Printer on-line

Next we run the program ttytool. We fill in the "Device Name" entry with `/dev/cuA4` and click on the "Get" button.

We click on the "Edit Hardware Control" button to bring up the "Hardware Controls" pop-up window. The printer is set for 9600 baud according to the DIP switches. We set the communication parameters to Output Speed:9600, Stop Bits: 2, Bits/Character: 8, and Parity: None.

To test the printer, we type a few lines of "abcdefgh" in the ttytool terminal window. Our printer prints "!#^%!@%#^%&^%&&^#@!$%#$%". This is not right. Looking in our printer manual we discover that we have set the wrong DIP switches. We correct the problem and try again.

This time the printer works correctly except everything prints on the same line. Our end-of-line character is <Return>. We need a <line feed> character to space the paper up. <line feed> is <Control-J>, so typing <Control-J> a few times will cause the paper to feed up. Normally, the conversion of end-of-line into <Return><line feed> is done in the printer filter so we are not worried. Just for fun, we try a <form feed> character and the printer does try to eject a page, but the paper jams.

The number of stop bits is cut down to one and we try again. Still works.

Just for fun, we try 7 data bits and even parity. Now "abcdefgh" comes out as "abcdefgh". So the high bit is used for underline (an undocumented feature of the printer). We set the parameters back to 8 data bits, no parity.

We fill in the "File Name" entry with "/tmp/printcap.add" and click on "Write Printcap Entry" button.

`/etc/printcap` is edited to add the new entry. The new printer is named "accounting" and has a spool directory of `/var/spool/accounting`.

We set up the spool directory with the commands:

```
% mkdir /var/spool/accounting
% chmod 775 /var/spool/accounting
% chown daemon /var/spool/accounting
% chgrp daemon /var/spool/accounting
% chmod g+s /var/spool/accounting
```

The line printer spooler is started using the command:

```
% lpc start accounting
```

We test the printer with the command:

```
% lpr -Paccounting /etc/fstab
```

and it works.

The project is complete.

3.5.7 Simple Hardware Flow control for Serial Printers

In this example, we are attaching a serial printer to port "/dev/ttyA2" on an ArtePort SB-1600. The SB-1600 does not have "normal" hardware flow control signals (RTS/CTS) so we have to substitute.

1. Make sure that the device is labeled as "remote" in /etc/ttytab. In the following line from /etc/ttytab, you can see we have changed "local" to "remote".

```
ttyA2 "/usr/etc/getty std.9600"
unknown off remote secure
```

2. When you reboot the system, soft carrier will be turned off as a result of the above change. If you do not wish to reboot now, you can turn off soft carrier with the following command:

```
# ttysoftcar -n /dev/ttyA2
```

3. The following /etc/printcap entry is a good starting point. The printer behaves as desired using this entry. Refer to **Appendix B** for other printcap entries.

To prevent Artecon's SW driver from post-processing a printfile (which causes problems when trying to print a postscript file on a postscript printer), add the line/entry to the printcap file:

:ms=-opost:

```
# ArtePort Serial Printer Test
testlp:\
    :lp=/dev/ttyA2:\
```

```
:br#9600:\
:sd=/var/spool/testlp:\
:lf=/var/adm/lpd-errs:
```

We did come across one limitation. When we included the following line in our `/etc/printcap` *entry, the spooler would be cleared of all files when the printer was powered off.*

```
:of=/usr/lib/lpf:
```

4. Finally, you must use a special cable to achieve the desired results, as shown in **Figure 3-7.**

For this example, we assume that the DTR signal from the printer goes open (low) when you power the printer off. If this is not true for your particular printer, determine a signal (i.e. DSR, CD, RTS, CTS) that does exhibit this behavior and substitute it for DTR in the pinout.

As you can see, we wire DTR on the printer side to CD on the SB-1600 side. The ArtePort driver monitors CD and will cease data flow when it goes open (low). Data flow will resume when CD goes high (i.e. when the printer is powered on). These 5 wires are all that are needed and should work on any given serial printer. Also, many printers drop DTR to signify "Buffer Full", ie stop sending data until buffer has been emptied.

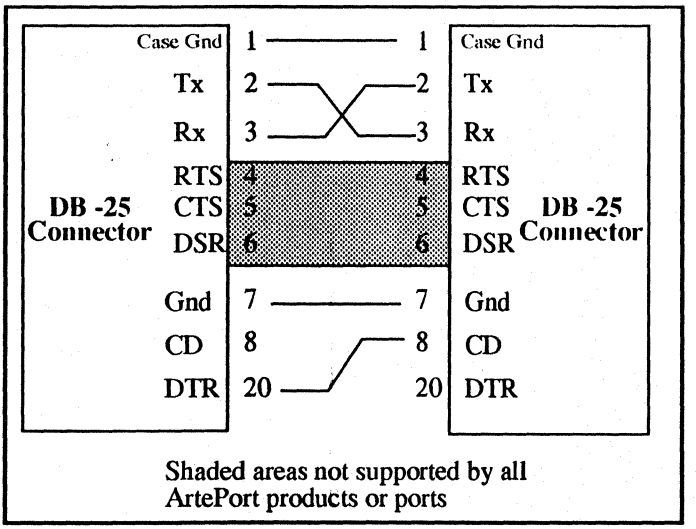


Figure 3-7. Serial Printer Cable

NOTE: *Occasionally, when you spool a file to a printer that is currently powered off, the following line will appear on the console:*

```
lpd[1453]: testlp:
ioctl(TIOCEXCL): No such device or address. The
easy fix is to execute the following command:
        # lpc restart testlp
```

where "testlp" is the name of the printer.

This will restart the spooler from where it left off.

3.6 Adding a Terminal

NOTE: *This example is for SunOS 4.1.2 - 4.1.3 users. Refer to the Sun document which covers adding devices and drivers for more information on adding modems to SunOS 5.1 - 5.3 systems.*

The following information explains how to add a terminal to your system.

You must first connect the device to the computer.

Edit the file `/etc/ttytab`. Each line in this file corresponds to a device. A typical entry looks like:

```
ttyA5 "/usr/etc/getty std.9600"  vt100\
 off local secure
```

Where:

`ttyA5` is the name of the device

"`/usr/etc/getty std.9600`" is the command used to start the login process on the terminal. There are more complex options available for the command *getty(8)*; see the *getty(8)* manual page for more information on this command.

NOTE: *Since the Artecon driver starts getty when CD goes high, make sure you have the appropriate cable setup, i.e. straight through, twist or null-modem*

`vt100` is the terminal type. Change this to the correct terminal type.

`off` indicates that this port is not being used. Change this entry to `on`.

`secure` indicates that the terminal is in a secure area and can be used for root logins. Delete the word `secure` if you wish to disable root logins on this device

After you edit the `/etc/ttytab` file you must notify the operating system of the changes. Execute the following command as root:

```
# kill -1 1
```

At this point, the terminal should have a login prompt.

3.7 Adding a Modem

This example is for SunOS 4.1.2 - 4.1.3 users. Refer to the Sun document which covers adding devices and drivers for more information on adding modems to SunOS 5.1 - 5.2 systems.

The following information explains how to add a modem to your system.

You must first connect the device to the computer.

Edit the file `/etc/ttytab`. Each line in this file corresponds to a device. A typical entry looks like:

```
ttyA5 "/usr/etc/getty std.9600" vt100\
off local secure
```

Where:

`ttyA5 is the name of the device`

`"/usr/etc/getty std.9600"` is the command used to start the login process on the modem. There are more complex options available for the command *getty(8)*; see the *getty(8)* manual page for more information on this command.

`vt100` is the terminal type. Change this to `unknown`

`off` indicates that this port is not being used. Change this entry to `on`.

`remote` indicates that the device does support carrier.

`secure` indicates that the terminal is in a secure area and can be used for root logins. Delete the word `secure` if you wish to disable root logins on this device

After you edit the `/etc/ttytab` file you must notify the operating system of the changes. Execute the following commands as root:

```
# kill -1 1
# ttysoftcar -a
```

At this point the modem should be able to accept incoming calls.

Outgoing calls may be placed using the "call" device name `/dev/cuA<number>`.

Where:

number is the port that the modem is connected to.

See **Appendices D and E** for modem setup.

3.8 Adding a Parallel Printer

This example is for SunOS 4.1.2 - 4.1.3 users. SunOS 5.1 - 5.2 uses a different procedure for configuring printers. Refer to the appropriate Sun document which covers adding devices and drivers for more information on configuring printers.

The device name for the parallel port is `/dev/ttyAc`. The parallel driver provides output editing functions identical to the ones provided for serial ports. Refer to the `termios(4)` man page and the ArtePort Hardware Manual for more information.

The following is a minimal /etc/printcap entry that should work for most parallel printers. You should embellish the entry as necessary for your site.

To disable postprocessing of printfiles (which causes problems when printing postscript files on a postscript printer), add:

```
:ms=-opost:\
```

```
lp|Panasonic|Panasonic 1595:\
        :lp=/dev/ttyAc:\
        :br#9600:\
        :ms=-parenb,cs8,-cstopb,clocal,ignbrk:\
        :sd=/var/spool/pan:\
        :lf=/var/adm/printer_errs:\
        :of=/usr/lib/lpf:
```

See **Appendix C** for sample printcap entries for parallel port.

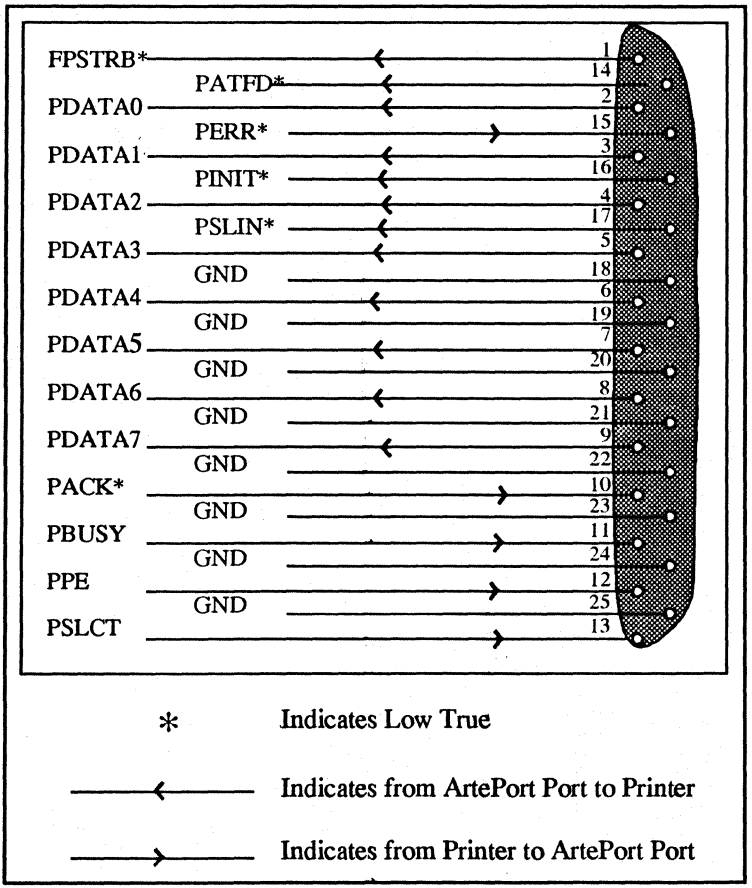


Figure 3-8. Parallel DB-25 Connector

Chapter 4.0 Troubleshooting

4.1 Introduction

This section provides reference information for troubleshooting the ArtePort Software. Refer to **Table 4-1.**

Table 4-1. Troubleshooting

Error	Remedy
Printer prints nothing	Check the cabling to see if you need a null modem. If not, check the communications settings, particularly the speed.
Printer prints garbage	Check communication speed.
Some characters come out correctly, others do not.	Check the parity setting.
Printer starts out fine, then prints garbage.	Check the flow control.
Modem does not hang-up on close	The ArtePort driver drops DTR for a short period of time. Some modems are expecting a longer DTR drop before they will hang up the line. There is usually a modem register (s10 on a Telebit) that can be changed to affect this behavior. For example: On a Telebit: ATS10 = 3 will lower the time to 0.3 seconds.

Appendix A RS232 Signals

RS232 uses a 25 pin connector. Although all 25 pins are defined, only the following 10 are used for asynchronous communications, as shown in **Table A.**

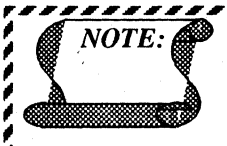
Not all signals are connected for all ports. For more information, refer to the ArtePort SBus Hardware User's Manual

Table A. RS232 Signals

Pin	Signal	Description
1	GND	Protective Ground
2	TD	Transmit Data stream. Data from DTE(Computer) to DCE
3	RD	Receive Data stream. Data from DCE to DTE(Computer)
4	RTS	Request To Send. This is a signal from the DTE(Computer) to the DCE that tells the DCE that the DTE(Computer) is ready to send data.
5	CTS	Clear To Send. Signal from the DCE to the DTE(Computer), telling the DTE(Computer) that it is OK to send data.
6	DSR	Data Set Ready. Signal from the DCE to the DTE(Computer), telling the DTE(Computer) that the DCE has been turned on and is ready to communicate.
7	GND	Signal Ground. In most cases, Signal Ground (7) and Protective Ground (1) are wired together. The difference between the two is of interest only to the purist.
8	CD	Data Carrier Detect (also known as DCD). A signal from the DCE indicating that it is connected to another DCE. "Carrier" is the "I'm alive" signal sent between two modems.
20	DTR	Data Terminal Ready. Signal from the DTE(Computer) to the DCE, telling the DCE that the DTE(Computer) is on and the initialization of the communications port is complete.
22	RI	Ring Indicator. Signal from the DCE indicating that the phone is currently ringing

Appendix B Sample Printcap Entries

Sample /etc/printcap entries:



NOTE: *For the parallel port, use: /dev/ttyAc*

```
hp4|hp4m|New HP LaserJet 4M by FAX Printer:\
  :lp=/dev/ttyAe:\
  :br#9600:\
  :fc#0177777:fs#06021:xc#0177777:xs#040040:\
  :ms=-parity,onlcr:\
  :mx#0:\
  :pw#80:pl#60:\
  :af=/usr/adm/lpd-errs:\
  :lf=/usr/spool/hp4/ilog:\
  :if=/AI/printers/hp4:\
  :st=/usr/spool/hp4/status:\
  :sd=/usr/spool/hp4:\
  :sh:
DMP|CALCOMP|52424P:\
  :lp=/dev/ttyAc:\
  :sh:\
  :sf:\
  :mx#0:\
  :sd=/var/spool/calcomp:\
  :lf=/var/spool/calcomp/log:
ps|postscript|PostScript|postscript emulation on TI microlaser:\
  :lp=/dev/ttyAc:\
  :sd=/var/spool/ps:\
  :br#9600:\
```

```
:mx#0:\
:sh:\
:df=/usr5/wp/slib/wppscript:\
:nf=/var/spool/ps/wpps:\
:gf=/usr5/wp/slib/wppscript:\
:cf=/usr3/wp/shbin/wpp:
```

lplhplhpllhewlett packard emulation on TI microlaser:\

```
:lp=/dev/ttyA4:\
:sd=/var/spool/hp:\
:br#9600:\
:mx#0:\
:sh:\
:nf=/var/spool/hp/hpwp:\
:gf=/usr5/wp/slib/wppscript:\
:cf=/usr3/wp/shbin/wpp;
```

This is a sample of printcap entries used by various printers/ plotters. All entries have been commented out to avoid complaints from the line printer daemon about printers that don't really exist.

```
# # DecWriter over a tty line.
# lp|ap|arpa|ucbarpa|LA-180 DecWriter III:\
#    :br#1200:fs#06320:tr=\f:\
#    :if=/usr/lib/lpf:lf=/usr/adm/lpd-errs:
# # typical remote printer entry
# ucbvax|vax|vx|ucbvax line printer:\
#    :lp=:rm=ucbvax:sd=/usr/spool/vaxlpd:\
#    :lf=/usr/adm/lpd-errs:
# varian|va|Benson Varian:\
#    :lp=devva0:sd=usrspoolvad:mx#2000:pl#58:\
#    :px#2112:py#1700:tr=\f:\
#    :of=/usr/lib/vpf:if=/usr/lib/vpf:\
#    :tf=/usr/lib/rvcat:cf=/usr/lib/vdmp:\
#    :gf=/usr/lib/vplotf:df=/usr/local/dvif:\
```

```
#    :vf=/usr/lib/vpltdmp:\
#    :lf=/usr/adm/lpd-errs:

# versatec|vp|Versatec plotter:\
#    :lp=/dev/vp0:sd=/usr/spool/vpd:\
#
:sb:sf:mx#0:pw#106:pl#86:px#7040:py#2400:\
#    :of=/usr/lib/vpfW:if=/usr/lib/vpsf:\
#    :tf=/usr/lib/vcat:cf=/usr/lib/vdmp:\
#    :gf=/usr/lib/vplotf:vf=/usr/lib/vpltdmp:\
#    :lf=/usr/adm/lpd-errs:
#    :tr=\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
# \n\n\n\n\n\n\n:
```

Appendix C Parallel Printcap

```
#Printcap Entry from Landscape
hpps|New HP LaserJet III Printer PS:\
   :lp=/dev/ttyAc:\
   :br#9600:\
  :fc#0177777:fs#06021:xc#0177777:xs#040040:\
   :ms=-parity,onlcr:\
   :pw#80:pl#60:\
   :af=/usr/adm/lpd-errs:\
   :lf=/usr/spool/hpps/ilog:\
   :if=/AI/printers/hpiiips:\
   :st=/usr/spool/hpps/status:\
   :sd=/usr/spool/hpps:\
   :sh:
#-------
#Printcap Entry from Landscape(Word Pecfect)
hpwp|laser|hpiii|New HP LaserJet III Printer:\
   :lp=/dev/ttyAc:\
   :br#9600:\
  :fc#0177777:fs#06021:xc#0177777:xs#040040:\
   :ms=-parity,onlcr:\
   :pw#80:pl#60:\
   :af=/usr/adm/lpd-errs:\
   :if=/AI/printers/hpiiiwp:\
   :lf=/usr/spool/hpwp/ilog:\
   :st=/usr/spool/hpwp/status:\
   :sd=/usr/spool/hpwp:\
   :sh:

#Printcap Entry from Landscape
```

```
hp|laser|hpiii|New HP LaserJet III Printer:\
    :lp=/dev/ttyAc:\
    :br#9600:\

:fc#0177777:fs#06021:xc#0177777:xs#040040:\
    :ms=-parity,onlcr:\
    :mx#0:\
    :pw#80:pl#60:\
    :af=/usr/adm/lpd-errs:\
    :lf=/usr/spool/hp/ilog:\
    :if=/AI/printers/hpiii:sb:\
    :st=/usr/spool/hp/status:\
    :sd=/usr/spool/hp:\
    :sh:
```

Appendix D Modem Setup

The 1st section below is a sample copy /etc/remote file (that is used to tip or dial out on a modem. The 2nd section is a copy of gettytab file which is used (along with /etc/ttytab) when someone dials into a system & controls the login process.

In the 1st section, cuD0 refers to port 0 of the 4th (A,B,C,D)Arteport board, cuD1 is port 1, etc. As you can see, the br# values set the baud rate.

For example, to connect to a modem on port 2 @ 38400, issue the following command:

cuD2

this opens the connection so that you can now issue the modem commands e.g. ATDT xxxxxx to get the modems attention & dial xxxxx.

NOTE: *In the following directions, I will refer to cuD0 as the 1st port on the 4th board. If you are using only one board, then the value you must use is cuA0 (ie 1st port on 1st board). Whenever cuD is used, substititue the correct value (e.g. cuA0, cuB1, cuC5, etc).*

To set-up a tip entry for 38400, add the following lines:

tip38|tip38400:tc-UNIX-38400

UNIX-38400:\

```
:el=^D^U^C^S^Q^O@:du:at=ventel:ie=#$%:\
oe=^D:br#38400:tc=dialers:
```

and change dialers from:

dialers:\

```
:dv=/dev/cua0,/dev/cua1:
```

to:

dialers:\

```
:dv=/dev/cuD0,/dev/cuD1,/dev/cuD2,/cuD3
```

To setup the port for incoming calls, 1st add line h from our gettytab file (3rd section) to your /etc/gettytab file. Next, modify the /etc/ttytab file and change std.9600 to std.38400 to setup the new incoming baud rate.

NOTE: *Since you are using SB-1600s, you CANNOT ust the rts/cts (hardware flow control) mode.*

Line j of our gettytab shows how you can add extra filters to the operation of the ports.

For additional information, I suggest you do.

man remote

and/or

man gettytab

at the Unix command prompt.

Section 1.0

```
cua0:dv=/dev/cua0:br#19200
cua1:dv=/dev/cua1:br#9600
cuD0:dv=/dev/cuD0:br#38400
cuD1:dv=/dev/cuD1:br#38400
cuD2:dv=/dev/cuD2:br#38400
#cuD3:dv=/dev/cuD3:br#9600

dialup1|Dial-up system:\
        :pn=2015551212%:tc=UNIX-1200:
hardwire:\
        :dv=/dev/ttyb:br#9600:\
        :el=^C^S^Q^U^D:ie=%$:oe=^D:
```

```
sun:\
        :dv=/dev/ttyb:br#2400:\
        el=^C^S^Q^U^D:ie=%$:oe=^D:\
        pn=918004774768:
a9600:\
        :dv=/dev/ttyb:r#19200:\
        el=^C^S^Q^U^D:ie=%$:oe=^D:
tip0|tip1200:tc=UNIX-1200:
tip300:tc=UNIX-300:
cu0|cu300:tc=UNIX-300:
cu1200:tc=UNIX-1200:
UNIX-300:\
        :el=^D^U^C^S^Q^O@:du:at=ventel:\
        ie=#$%:oe=^D:br#300:tc=dialers:
UNIX-1200:\
        :el=^D^U^C^S^Q^O@:du:at=ventel:\
        ie=#$%:oe=^D:br#1200:tc=dialers:
VMS-300|TOPS20-300:\
        :el=^Z^U^C^S^Q^O:du:at=ventel:\
        ie=$@:oe=^Z:br#300:tc=dialers:
VMS-1200|TOPS20-1200:\
        :el=^Z^U^C^S^Q^O:du:at=ventel:\
        ie=$@:oe=^Z:br#1200:tc=dialers:
dialers:\
      :dv=/dev/cua0,/dev/cua1:
f:\
      :dv=/dev/cuD0:br#9600:at=ventel:
e:\
      :dv=/dev/cuD1:br#9600:at=ventel:
d:\
      :dv=/dev/cuD2:br#9600:at=ventel:
c:\
      :dv=/dev/cuD3:br#9600:at=ventel:
```

b:\

:dv=/dev/cuBb:br#9600:at=ventel:

The attributes are:

dv	device to use for the tty
el	EOL maeks (default is NULL)
du	make a call flag (dial up)
pn	phone numbers (@=>'s search phones file; possibly taken from PHONES environment variable)
at	ACU type
ie	input EOF marks (default is NULL)
oe	output EOF string (default in NULL)
cu	call unit (deafult is dv)
br	baud rate (defaults to 300)
fs	frame size (default is BUFSIZ) -- used in buffering writes on receive operations
tc	to continue a capability

Section 2.0

@(#)gettytab 1.9 89/09/27 SMI; from UCB 5.7 2/16/86

Copyright (c) 1980 Regents of the University of California. All rights reserved. The Berkeley software License Agreement specifies the terms and conditions for redistribution.

Most of the table entries here are just copies of the old getty table, it is by no means certain, or even likely then any of them are optimal for any purpose whatever . Nor is it likely that more than a couple are even correct

The default gettytab entry, used to set defaults for all other entries, and in cases where getty is called with no table name

default:\

```
:ap:lm=\r\n%h (%t) login\72 :sp#9600:
```

#

Fixed speed entries

```
#The "std.NNN" names are known to the
special case portselector code in getty,
however they can be assigned to any table
desired. The "NNN-baud" names are known to
the special case autobaud code in getty,
and likewise can be assigned to any table
desired (hopefully the same speed).
```

alstd.110|110-baud:

```
:nd#1:cd#1:uc:sp#110:
```

blstd.134|134.5-baud:

```
:ep:nd#1:cd#2:fd#1:td#1:sp#134:ht:nl:
```

1lstd.150|150-baud:\

```
:ep:nd#1:cd#2:td#1:fd#1:sp#150:ht:nl:\
lm=\E\72\6\6\17login\72 :
```

clstd.300|300-baud:\

```
:nd#1:cd#1:sp#300:
```

dlstd.600|600-baud:\

```
:nd#1:cd#1:sp#600:
```

flstd.1200|1200-baud:\

```
:fd#1:sp#1200:
```

6lstd.2400|2400-baud:\

```
:sp#2400:ht:
```

7lstd.4800|4800-baud:\

```
:sp#4800:ht:
```

2lstd.9600|9600-baud\

```
:sp#9600:
```

glstd.19200|19200-baud:\

```
:sp#19200:
```

ilnonstd.19200|19200-baud:\

```
:sp#19200:ms=crtscts:
```

hlstd.38400|38400-baud:\

```
:sp#38400:
```

jlnonstd.38400|38400-baud:\

```
        :sp#38400:ms=crtscts:
#
# Dial in rotary tables, speed selection via 'break'
#
0|d300|Dial-300:\
        :nx=d1200:cd#2:sp#300:
d1200|Dial-1200:\
        :nx=d150:fd#1:sp#1200:
d150|Dial-150:\
        :nx=d110:lm@:tc=150-baud:
d110|Dial-110:\
        :nx=d300:tc=300-baud:
#
# Odd special case terminals
#
-|tty33|asr33|Pity the poor user of this beast:\
        :tc=110-baud:
4|Console|Console Decwriter II:\
        :co:nd@:cd@:rw:tc=300-baud:
e|Console-1200|Console Decwriter III:\
        :fd@:nd@:cd@:rw:tc=1200-baud:
i|Interdata console:\
        :uc:sp#0:
l|lsi chess terminal:\\
        :sp#300:
X|Xwindow|X window system\
        :fd@:nd@:cd@:rw:sp#9600:
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either
way)
#
D2400|Fast-Dial-2400\
```

```
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200\
        :nx=D300:fd@:tc=1200-baud:
5|D300|Fast-Dial-300\
        :nx=D2400:tc=300-baud:
#
# Wierdo special case for fast crt's with hardcopy devices
#
8|T9600|CRT with hardcopy:\
        :nx=T300:tc=9600-baud:
9|T300|CRT with hardcopy (300)\
        : nx=T9600:tc=300-baud:
#
# Plugboard, and misc other terminals
#
p|P9600|Plugboard-9600:\
        :nx=P300:tc=9600-baud:
q|P300|Plugboard-300:\
        :nx=P1200:tc=300-baud:
r|P1200|Plugboard-1200:\
  :nx=P9600:tc=1200-baud:
#
# XXXX Port selector
#
s|DSW|Port Selector:\
      :ps:sp#2400:
#
# Auto-baud speed detect entry for Micom 600. Special code in
getty will switch this out to one of the NNN-baud entries.
#
A|Auto-baud:\
        ab:sp#2400:f0#040:
```

Appendix E Solaris Modems

This message is being posted as a public service to everyone pulling out their hair over pmadm and ttyadm. Ths instructions in Sun hardcopy docs and Answerbook are misleading. The instructions below do work.

Configuring Terminals and Modems under Solaris 2.x.

A. Setting up a terminal with Solaris 2.x

NOTE: *1. All Single quotes in these directions are BACK QUOTES, not FORWARD QUOTES.*

2. Be sure and execute these commands as root, from a BOURNE shell (sh), not ksh, tcsh, or csh.

3. If, after following these instructions, you get terminal failures, then make sure that root's default shell is /sbin/sh and not sch Logging in as csh-using user and doing an su to root with a password-file shell of sh is not good enough. You must login as root, and root's default shell must be the Bourne shell.

A.1 BEFORE YOU BEGIN: SET UP THE TERMINAL

- Set the terminal for the speed you wish (example below assumes you are using 9600 baud)

- The only lines you really need in the terminal-to-Sun cable are 2, 3, and 7. Be sure to cross lines 2 and 3 in your cable.

- Set the terminal for Xon/Xoff flow control

- Set the terminal for 7 bits, Even Parity. (1 stop bit, if you have that setting.)

A.2 ENABLING SOLARIS FOR A TERMINAL:

- Save the following script as add_terminal - make it executable with "chmod u (plus sign) x add_terminal" (plus sign) so suninfo can wk

- Edit add_terminal to meet your particular needs

- Execute add_terminal

- If the terminal doesn't give you a log when you hit <return> try rebooting the machine once.

Use sample script file below:

```
#!/bin/sh
#
# add_terminal shell script. Must be run as root from bourne shell
# Change these parameters as needed for your particular needs.
# PARAMETER          MEANING
# ---------          -------
#-s ttya             termial is on ttya. you may want ttyb
#-d  /dev/term/a     actual port device. you may want
                     /dev/term/b
#l- 960F             speed/stty setting from /etc/ttydefs.
                     choose an entry that meets your
                     needs
#-T tvi925           your default terminal type for this
                     port. This is a terminfo termnal type
#-i  'terminal disabled'  message sent to the tty port if the port
                     is ever disabled
# Remove any existing port monitor on ttya
# (Change the "ttya" to "ttyb" if that's the port you're dealing with)
#
pmadm -r -p zsmon -s ttya
#
# Create the new port monitor
#
pmadm -a -p zsmon -s ttya -i root -fu -v 1 -m "`ttyadm -l 9600F\
- d /dev/term/a -T tvi925 -i 'terminal disabled' -s /usr/bin/login \
-S y'"
```

B. Setting Up a Bidirectional Modem with Solaris 2.x

B.1 BEFORE YOU BEGIN: SET UP THE MODEM

Be sure that the modem is configured for the following:

- Hardware DTR, that is, when the Sun drops DTR the modem should hang up and reset the modem.

- Hardware Carrier Detect, that is, the modem only raises the CD line when there is an active carrier signal on the phone connection, when carrier drops, either when the other end of the connection terminated or in the event the phone connection is broken, the Sun will be notified and act appropriately. The CD signal is also used for coordinating dial-in and dial-out use on a single serial port and modem.

- Respond with numeric result codes. (Usually this is ATV0)

- Sends BASIC result codes ONLY. (Sometimes this is ATX0 or ATQ0)

- Does not echo commands. (Usually this is ATE0)

- Modem is locked at a single speed setting between Sun and modem.

Speed may vary as needed between local and remote modem, but should remain constant between Sun and modem.

B.2 ENABLING SOLARIS FOR A BIDIRECTIONAL MODEM x

1. All single quotes in these directions are BACK QUOTES, not FORWARD QUOTES.

2. Be sure and execute these commands as root, from a BOURNE shell (sh), not ks, tcsh, or csh.

3. If, after following these instructions, you get modem failures, or if step #7 gives an error messge about "ttyadm: -V: invalid parameter", then make sure that root;s default shell is /sbin/sh and not csh Logging in as csh-using user and doing an su to root with a password-file shell of sh is not good enough. You must login as root, and roots' s default shell must be the Bourne shell.

1. Login as root and type:

   prompt# eeprom ttya-ignore-cd=true <carriage return>

   prompt# eeprom ttya-rts-dtr-off=true <carriage return>

   >> go to step 2

2. Reboot the system by typing:

   prompt# init 6

   >> Go to step 3

3. Make sure the modem is properly connected and configured.

Make sure the cable is ok.

>> Go to step 4

4. Is zsmon configured and running?

(The following instructions assume that the modem cable to serial port A.)

Login as root and type:

prompt# sacadm -l -t ttymon

If you get a message like:

"Invalid request, zsmon does not exist"

then the zsmon port monitor is not configured.

>> Go to step 5

If you get a result like:

PMTAG	PMTYPE	FLGS	RCNT	STAUTUS
zsmon	ttymon	-	0	ENABLED

COMMAND

/usr/lib/saf/ttymon

>> Go to step 6

5. To configure the zsmon port monitor

prompt# sacadm -a -p zsmon -t ttymon -c /usr/lib/saf/ttymon \

-v `ttyadm -V` -y "dial in/out on serial port a"

(Note: zsmon = PMTAG

"dial in/out on serial port a" = "any comment")

>> Go to step 6

6. Is ttymon configured and running?

Type:

prompt# pmadm -l -s ttya (where ttya is the SVCTAG)

a) If there is no service, you will get a result like: the

Service <a> does not exist.

>> Go to step 7.

b) If you got a result like:

PMTAG	PMTYPE	SVCTAG	FLGS ID
zsmon	ttymon	ttya	u root

<PMSPECIFIC>

/dev/term/a I - /

Then you need to remove the existing service by typing:

prompt# pmadm -r -p zsmon -s ttya

^ ^

| |

(PMTAG) (SVCTAG)

>> Go to step 7.

7. Create a new bidirectional port service. Type:

prompt# # pmadm -a -p zsmon -s ttya -i root -v `ttyadm -V-\
-fu -m\ "`ttyadm -d /dev/term/a -s /usr/bin/login -l contty\
3H -b -S n -m\ ldterm,ttcompat`" -y "dial in/out on serial\
port a"

>> Go to step 8.

zsmon - PMTAG

ttya - SVCTAG

/dev/term/a - <PMSPECIFIC>

contty3H - ttylabel, defined in /etc/ttydefs file (speed setting. cpmmtty3H=2400 baud. contty5H=19200 baud.)

-b set flag for bi-directional

-S n to turn software carrier off

"dial in/out on serial port a" - "any comment")

8. TIP and /etc/remote

Edit /etc/remote and find the entry beginning with "hardwire:". Change the portion which says:

:dv=/dev/??????:

to read

:dv=/dev/cua/a: (or /dev/cua/b if you're using the B port)

Change the portion which says:

:br#somenumber:

to read

:br#9600: (or 19200, or whatever baud rate you plan to use)

>> Go to step 9

9. Type:

prompt# tip hardwire

you should see "connected"

type:

ATE1V1<carriage return>

you should see "OK"

>> Go to step 10

If you don't see "OK", you are not communicating with the modem. See if the modem is plugged in, verify the baud rate of the modem and the hardwire entry in /etc/remote, verify the modem cable.

10. Test the dialin

Test your modem setup by dialing into the Solaris 2.x machine. If you get logged in, congratulate yourself!

>> Go to step 11

11. Further setup for using the modem

At this point, you have established basic outgoing computer-to-modem setup. If you plan to use uucp, then start setting up the /etc/uucp files. (This is left as an exercise for the reader.)

If you are interested in more ways to use tip with your modem, see Appendix A of this document.

1. The cuaa entry (below) allows you to type "tip cuaa" and talk directly to a modem on port a at 2400 baud.

2. "tip phonenumber" will look for the tip0 entry and use that definition. (In the example below, it will use tip0, which points to UNIX-2400, which sets up a dialup 2400 baud hayes-compatible modem. UNIX-2400 points to dialers, which references /dev/cua/a.

3. "tip mysystem" (below) looks up the "mysystem" entry, and dials the number 14155551212 via a 19200 baud connection on / dev/cua/a.

The following sample file is set up for

port a, 2400 baud hayes-compatible modem

port b, 9600 baud modem, may or may not be hayes compatible

/etc/remote:

```
============
cuaa:dv=/dev/cua/a:br#2400
mysystem:pn=14155551212:tc=UNIX-19200:
hardwire:\
    :dv=/dev/cua/b:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
tip300:tc=UNIX-300:
tip1200:tc=UNIX-1200:
tip0|tip2400:tc=UNIX-2400:
tip9600:tc=UNIX-9600:
tip19200:tc=UNIX-19200:
cu300:tc=UNIX-300:
cu1200:tc=UNIX-1200:
cu0|cu2400:tc=UNIX-2400:
UNIX-300:\
    el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
    :oe=^D:br#300:tc=dialers:\:
UNIX-1200:
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
    :oe=^D:br#1200:tc=dialers:
UNIX-2400:\
        :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
```

```
        :oe=^D:br#2400:tc=dialers:
UNIX-9600:
        :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
        :oe=^D:br#9600:tc=dialers:
UNIX-19200:\
        :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
        :oe=^D:br#19200:tc=dialers:
VMS-300|TOPS20-300:\
        :el=^Z^U^C^S^Q^O:du:at=hayes:ie=$@:\
        :oe=^Z:br#300:tc=dialers:
VMS-1200|TOPS20-1200:\
        :el=^Z^U^C^S^Q^O:du:at=hayes:ie=$@:\
        :oe=^Z:br#1200:tc=dialers:
dialers:\
        :dv=/dev/cua/a:
```

The attributes are:

dv	device to use for the tty
el	EOL maeks (default is NULL)
du	make a call flag (dial up)
pn	phone numbers (@=>'s search phones file; possibly taken from PHONES environment variable)
at	ACU type
ie	input EOF marks (default is NULL)
oe	output EOF string (default in NULL)
cu	call unit (deafult is dv)
br	baud rate (defaults to 300)
fs	frame size (default is BUFSIZ) -- used in buffering writes on receive operations
tc	to continue a capability

(This doc is revision 8/13/93)

ArtePort Notes:

Artecon™

July 15, 1994

READ THIS FIRST

ArtePort 1.8.1 Software

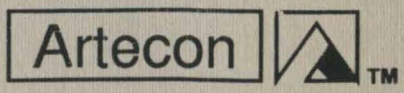
INSTALLATION CHANGES:

Solaris 2.X
Change install script so it will create cu devices as 0x20000+port instead of 0x10+port.

SunOS 4.1.X
Change install script so it will create cu devices as 0x80+port instead of 0x10+port.

The install script no longer is specific to the OS version. It will install based on the system architecture. The current supported architectures are Sun4c and Sun4m for SunOS 4.1.X. The support architectures are Sun4c, Sun4m and Sun4d for Solaris 2.X.

6305 El Camino Real
Carlsbad, CA 92009-1606
(619) 931-5500, FAX (619) 931-5527



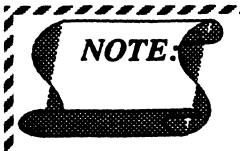
ArtePort 1.6 SW Manual
Made with Pride in Carlsbad, California, USA
P/N 83-53011703 Rev 1.1-12/93

Artecon™

July 15, 1994

READ THIS FIRST

ArtePort 1.8.1 Software



All references in the ArtePort software manual apply to ArtePort 1.8.1 software shipped. The Revision level change to ArtePort 1. 8.1 was initiated for software enhancements.

DRIVER CHANGES from 1.7 to 1.8.1:

PROM. ArtePort hardware with PROM dated prior to June 1994 doesn't work well with CD1400 rev G. The FCode PROM was doing chip reset on the 1400 upon bootup. However, due to timing differences on the SS5 and SS1000 hardware, chip will not reset correctly and will see chip status of 4 for failure to reset, instead of 0 for normal (status of 5 means the socket is not populated with CD1400 and is normal in 300S, 400P and 800P). ArtePort 1.8.1 will make the port work by doing reset in the driver and will work with older and newer PROMs.

SunOS 4.1.X only

Parallel signal STROBE has incorrect width in previous version of ArtePort. Thus some printers will fail because the STROBE* width is too short.

DRIVER CHANGES from 1.6(a) to 1.7 (included in 1.8):

Fix DTR dropping problem. The problem was manifest if there are background jobs around when a getty died. The port thought the background process is still having control of the port.

Changed callout port (cu devices) from bit 0x10 to 0x80 as Sun's convention. This allow more software compatibility.

Solaris 2.X

Changed callout port (cu devices) from bit 0x10 to 0x20000 as Sun's convention. This allow mor software compatibility.

Fix driver unloading problem after killing getty. Driver will now unload after killing all pror that was using a port.

6305 El Camino Real
Carlsbad, CA 92009-1606
(619) 931-5500, FAX (619) 931-5527

P/N 83-53011704, Rev 1.8.1 7/94