The One Port

User’s Guide

Versalynx Corporation
8950 Carley Circle
San Diego, CA 92126

Document Number: 00066-380
Published in October 2000.
Printed in the United States of America.

Notices

Copyright © 2000 Versalynx Corporation. All rights reserved.

This document is protected by federal copyright law. No part of this document may be photocopied, reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written permission of Versalynx Corporation.

Limitation of Liability

This document is provided without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Versalynx may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time. In addition, Versalynx reserves the right to revise this document and to make changes in content from time to time without obligation on the part of Versalynx Corporation to provide notification of such revision or changes.

Trademarks

Any trademarks mentioned in this manual are acknowledged to be the property of the trademark owner.

Acknowledgments

The One Port’s operating system is based on Xinu. Xinu is a small multitasking operating system designed by Douglas Comer at Purdue University and developed with support from the National Science Foundation. The price of The One Port does not include any charge to the purchaser for Xinu itself.

FCC Compliance

WARNING: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with this instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case, the user will be required to correct the interference at personal expense.

Any changes or modifications made to this equipment that are not expressly approved by Versalynx Corporation may void the user’s authority to operate this equipment.

CE

The One Port conforms to the following European Union Council Directives and Standards:

Emissions:	EN 55022, Class A
Immunity:	IEC 801-2, IEC 801-3, IEC 801-4

Limited Warranty

Versalynx warrants all of its products to be in good working order and free from defects in materials and workmanship for a period of ONE YEAR from the date of shipment from Versalynx or its authorized suppliers. This warranty does not apply to products that have been damaged or made inoperable as a result of accident, misuse, abuse or modification. This warranty is not transferable.

If a product does not operate as warranted, Versalynx will, at its option, replace or repair the defective product at no charge. A replaced product becomes the property of Versalynx. Any replaced or repaired product carries either a 90 day warranty or the remainder of the initial warranty period, whichever is longer.

When returning a product, obtain a Return Material Authorization (RMA) number and forward the product transportation prepaid. Versalynx will not accept responsibility for a product that is improperly packaged and/or damaged in shipment.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO VERSALYNX PRODUCTS. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS WHICH VARY FROM STATE TO STATE.

IN NO EVENT SHALL VERSALYNX OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION: DAMAGES FOR LOSS OF BUSINESS PROFITS, LOST SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PRODUCT, NOTWITHSTANDING THE FACT THAT VERSALYNX MAY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Contents

Appendixes

List of Figures

List of Tables

Introduction 1

Product Overview

The One Port is a compact single-port terminal server offering a full complement of industry standard network protocols and system features. It connects a single RS-232 device a TCP/IP-based Ethernet network (see Figure 1 below).

The One Port can operate as a terminal server, a print server, or a remote access server. Its most popular usage, however, is to "network-enable" a serial device. The One Port connects a legacy serial device (or a more modern serial device that does not have a network port) to a network so that the device can be monitored and controlled remotely.

The One Port is easy to install and to configure. You can have it operational within minutes and, after using it for a short period of time, you will have mastered the majority of its simple yet powerful and familiar commands.

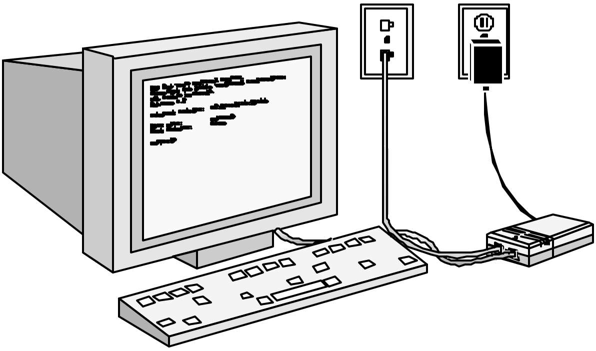

Figure 1. *The One Port Links a Terminal to a Network*

Product Description

The One Port is completely self-contained and requires no additional external software to be loaded onto a network server or into The One Port itself.

The One Port's features are listed below.

HARDWARE COMPONENTS

- Motorola 12 MHz 68HC000 CPU
- SMC91C96 IEEE 802.3 Ethernet controller
- 64K bytes static RAM; 2K bytes NVRAM
- 256K bytes EPROM
- Two LED status indicators:
  - power
  - link integrity
- Two serial interfaces using RJ-45 connectors:
  - 10BASE-T (twisted-pair Ethernet)
  - RS-232C
- RS-232C serial communications:
  - 50 to 115,200 bps
  - software (XON/XOFF) and hardware flow control
  - full modem control

SOFTWARE MODULES

- ARP, DHCP, DNS, ICMP, RARP, RIP, SNMP, TCP/IP and UDP Internet protocols
- Rlogin, TELNET, reverse TELNET, PPP, SLIP and LPD applications
- Easy-to-use network access, system configuration, and security commands

PHYSICAL CHARACTERISTICS

Height:	1 3/16 inches (30 mm)
Width:	2 5/8 inches (67 mm)
Depth:	4 15/16 inches (125 mm)
Weight:	6 ounces (170 grams)

EXTERNAL WALL-MOUNT POWER SUPPLY

- Specifications vary according to country model

Getting Started 2

Parts List

The One Port comes with an external wall-mount power supply and this manual. You will need to provide the interface cables or purchase them from Versalynx (see Figure 2 below). The One Port uses a Level 5 Unshielded Twisted-Pair (UTP) modular cable to connect to an Ethernet network and an RS-232 cable to connect to a terminal (see Chapter 5, “Cable Specifications” for cabling information).

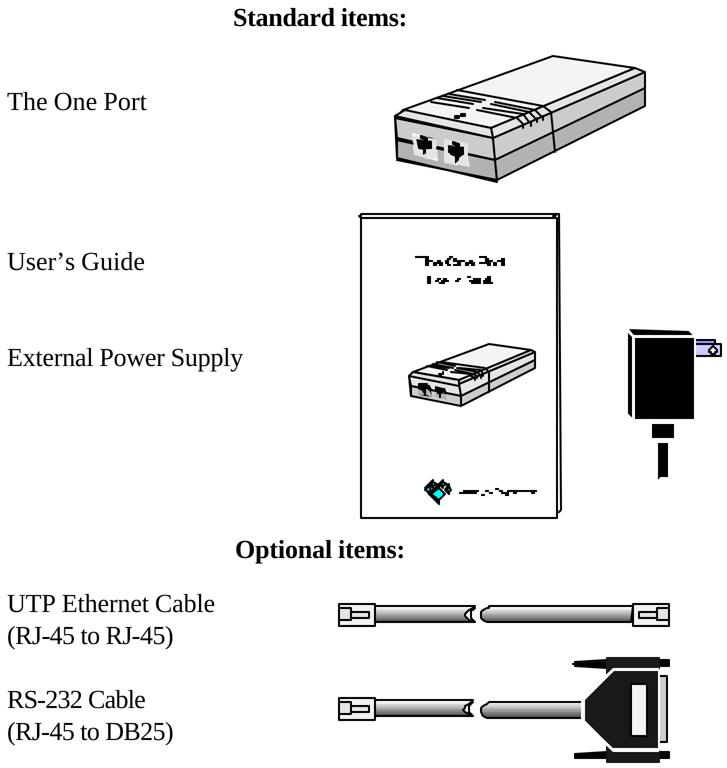

Figure 2. *System Components*

Placement of The One Port

Place The One Port near your terminal, a Local Area Network (LAN) connection and a power outlet. Make sure there is approximately two inches of unobstructed space on the top and on each side of the unit to allow for proper air flow.

Installing The One Port

To install The One Port:

1. Insert the RJ-45 connector of the RS-232 cable into The One Port's modular outlet labeled "Terminal" and the DB25 connector into your terminal's modem port. Turn on the terminal. Set the input and output speeds to 9,600 baud and the character size to eight data-bits with one stop-bit and no parity-bit.

2. Connect one end of the twisted-pair cable into The One Port's modular outlet labeled "TPE" and the other end into your LAN outlet or device (usually a network hub or computer).

Since telephone and Twisted-Pair Ethernet (TPE) network outlets and plugs look very similar, be sure to connect The One Port into your Local Area Network and not into the public telephone network (especially if they are on the same wall plate). You may damage both The One Port and the telephone network by connecting The One Port into the telephone outlet.

3. Plug the power cord into The One Port's power jack and into a standard AC outlet. Turn the power switch to the ON ("I") position.

The One Port's power status LED should now be on and the following welcome banner should be displayed on your terminal:

```
The One Port terminal server.
Copyright (c) 1994-2000, Versalynx
Corporation.
All rights reserved.
Release 3.8

Ethernet address:     00:20:D0:23:55:99

Host name:            oneport
Host address:         none

oneport%
```

If the welcome banner is not displayed or if the power status LED is not on, refer to Chapter 6, "Troubleshooting," for possible problem areas.

The link integrity LED should also be on. It lights when The One Port is connected to an operational network (and the connecting network device is sending link test pulses) or when the link state of The One Port is turned off. See the "TPE Link Test" section of Chapter 3 for additional information.

Configuring The One Port

After The One Port is turned on and connected to a network, you will need to configure several system parameters. The One Port's user commands, described in Appendix A, set the system parameters and provide network access. The commands are written in **bold** type. The configuration steps are listed below.

1. Assign The One Port an Internet Protocol (IP) address and (optionally) change its host name. The initial settings are `0.0.0.0` (none) and `oneport`, respectively. Consult your system administrator for an IP address that is appropriate for your network. The **ifconfig** command sets the IP address (refer to the **hosts** command to change The One Port's host name). For example, the command

```
oneport% ifconfig 192.10.160.3
```

sets The One Port’s IP address to `192.10.160.3`.

Alternatively, The One Port can acquire its IP address from a server using either the Dynamic Host Configuration Protocol (DHCP) or the Reverse Address Resolution Protocol (RARP). After it is turned on, The One Port broadcasts DHCP and/or RARP request messages (see the **snet** command, the “DHCP” section in Chapter 3, and the “RARP” section in Chapter 4). The server replies with a message containing The One Port’s IP address. Therefore, if there is a DHCP or RARP server on the network, you do not need to use the **ifconfig** command.

You can also configure The One Port’s IP address with the ping command. First, add The One Port’s Ethernet address into the host computer’s ARP table and then ping the IP address to be assigned to The One Port. For example, on a PC enter the commands:

```
arp -s 192.10.160.3 00-20-D0-XX-XX-XX
ping 192.10.160.3
```

Replace xx-xx-xx with the unit’s serial number (if the number only has five digits, precede the number with a zero). This is The One Port’s Ethernet address. It is also given in the welcome banner. Unlike an IP address configured with the **ifconfig** command, an address set with ping is not stored across a power cycle.

2. Now add networked computers (“remote hosts” or “hosts” for short) into the host database. These hosts are normally computers you are going to access with The One Port. You add hosts into the database with the **hosts** command. This command simply associates an IP address with the name of a computer or network device. For example, to add the host named “sun4” (which, in this example, has IP address `192.10.160.200`), you would enter:

```
oneport%   hosts   add   192.10.160.200
sun4
```

The One Port also supports the Domain Name Service. See Chapter 3 for details.

3. Use the **ping** command to verify that The One Port can communicate to network hosts (or devices). To test the accessibility of the “sun4” host, you would type either:

   ```
   oneport% ping sun4
   ```

   or

   ```
   oneport% ping 192.10.160.200
   ```

   The One Port responds with “`sun4 is alive`” if The One Port, the network, and the host are properly configured.

4. To access hosts on a different but attached network, you will need to provide The One Port with the IP address of the gateway that connects the two networks. You can do this with the **route** command:

   ```
   oneport% route add net default sun4 1
   ```

   In this example, The One Port will send all packets not destined for the local network (signified by the keyword `default`) to the gateway (`192.10.160.200` or “sun4”) to be routed appropriately.

5. You may also want to change some of your terminal’s characteristics—such as increasing the baud rate. You can change The One Port’s baud rate with the **stty** command:

   ```
   oneport% stty 38400
   ```

   At the completion of the **stty** command, change your terminal’s speed to the new baud rate.

6. After you have The One Port configured for your application, you can prevent the configuration from being reset to the factory defaults with the **lockmem** command:

   ```
   oneport% lockmem on
   ```

   You should, however, have the configuration settings exactly as you want them *before* running this command. All settings are saved in non-volatile memory; **lockmem** simply prevents them from being reset to the factory defaults after The One Port is turned off and on several times. (See the “Non-Volatile Memory” section of Chapter

3 and the **lockmem** command page in Appendix A for details.)

Advanced Configuration

The One Port can execute one or more user commands immediately after it is turned on. The **initcmd** command defines these commands. This is useful, for example, if you want The One Port to automatically establish a connection with a remote host. The terminal appears directly connected to the host computer since no user commands need to be entered at the keyboard.

Following the examples of the previous sections, an appropriate **initcmd** would be:

```
oneport% initcmd "rlogin sun4"
```

☞ Use the **initcmd** command only after configuring The One Port as described in the "Configuring The One Port" section of this chapter. Refer to **initcmd** in Appendix A for additional details.

System Features 3

This chapter describes the system features of The One Port terminal server.

Status Indicators

The One Port has two status Light-Emitting Diodes (LEDs) located at the top of the unit. The LED nearest the connector outlets is the power indicator. It lights when power is applied to the unit and when the power switch is in the ON (“I”) position. The other LED is the network link integrity indicator and lights when either the link state is on and The One Port is receiving link test pulses, or when the link state is off (see the “TPE Link Test” section below).

TPE Link Test

Network devices communicating on a twisted-pair Ethernet network should send link test pulses every 50 to 150 milliseconds as defined in the IEEE 802.3 10BASE-T specification. Unfortunately, not all devices follow the specification. To ensure compatibility with both conforming and non-conforming devices, The One Port’s link integrity test state can be enabled or disabled as needed in your network environment (see the **link** command in Appendix A).

To successfully exchange network packets, both The One Port and the connecting network device must have the same link test state. Set the link state of The One Port equal to the link state of the connecting device.

The One Port is factory configured with the link integrity test enabled.

Non-Volatile Memory

The One Port has 2K bytes of non-volatile memory that maintains the following system parameter settings when the unit is off:

- the terminal settings (**stty**)
- the protocol settings (**snet**)
- the host database including The One Port's name and IP address (**hosts, ifconfig**)
- the user name (**usrname**)
- the **initcmd** command(s)
- the link test state (**link**)
- the terminal lock state and password (**lock**, **passwd**)
- the hand-entered IP routing table entries (**route**)
- the domain name and server (**dname**, **dserver**)
- SNMP information (**snmpcom, snmpcon, snmploc, snmptrap**)
- the silent modes (**silent**)
- the memory lock state (**lockmem**)

The system parameter settings are reset to the factory default values by:

- executing the "**system -r**" command. After the completion of the command, turn The One Port off and on to enable the default parameter settings.

- turning The One Port off and on five consecutive times without keyboard input and without taking longer than five seconds between iterations. (Note: the **lockmem** command disables this reset method.)

  On occasion it may be necessary to reset the system parameters in this manner to exit a locked or incompatible state between The One Port and a connecting device. For example, if The One Port's baud rate is set to a value not supported by the attached terminal, The One Port and the terminal can no longer transfer useful data. The different baud rates make it impossible to use the **stty** command to reset The One Port's baud rate to the previous value. To correct this situation, you would set the terminal's baud rate equal to The One Port's default baud rate of 9,600 (see **stty**) and reset The One Port five times.

Command Line Interface

The One Port has a simple command line interface with the following characteristics:

- The **help** command displays a list of the available user commands. The syntax and usage of each command is provided in Appendix A. Command names are written in **bold** type in this manual.
- String arguments containing space or tab characters must be enclosed in single- or double-quotation marks.
- A period entered as the first character on a line and followed immediately by a carriage return repeats the previous command.
- The prompt is always set to The One Port's name (see the **hosts** command or the `sysName` MIB-II object in Appendix C) followed by the percent sign and space characters (e.g., "`oneport% `").

TELNET

The **telnet** command establishes a connection with a remote host using the TELNET terminal protocol. After connection establishment, **telnet** accepts characters entered at the keyboard. It sends the characters to the remote host either one character-at-a-time or collects them into lines and sends them line-by-line, depending on the mode setting (see "mode" in Table 1 below).

Command Mode

Telnet connects directly to a host when the host name or IP address is given on the command line; otherwise, **telnet** enters command mode and displays the TELNET prompt (“`telnet>` ”). Table 1 below describes the available commands and their arguments (you only need to type enough characters to uniquely identify a command). Arguments in square brackets are optional.

Command	**Argument(s)**	**Description**
?		Print a command summary.
close		Close an open connection.
display	[set or toggle param]	Display the set and toggle values.
mode	?	Print mode command help.
	character	Send text immediately to the host (character-at-a-time mode).
	line	Send text to the host in lines (line-by-line mode); echo characters locally.
open	*host* [*port*]	Open a connection to *host*. Use the default TELNET server port unless *port* is specified.
quit		Close an open connection and exit.
send	?	Print send command help.
	ao	Send the TELNET AO (abort output) sequence. AO flushes the remote system’s output to the terminal.
	ayt	Send the TELNET AYT (are you there) sequence.

Table 1. *TELNET Commands*

Command	Argument(s)	Description
send (cont.)	brk	Send the TELNET BRK (break) sequence.
	ec	Send the TELNET EC (erase character) sequence. EC erases the last input character.
	el	Send the TELNET EL (erase line) sequence. EL erases the current input line on the remote system.
	escape	Send the TELNET escape character.
	ga	Send the TELNET GA (go ahead) sequence.
	ip	Send the TELNET IP (interrupt process) sequence. IP aborts the currently running process on the remote host.
	nop	Send the TELNET NOP (no operation) sequence.
	synch	Send the TELNET SYNCH sequence. SYNCH flushes any data not yet read.
set	?	Print set command help.
	echo *char*	Toggle between local and remote echoing when in line-by-line mode. The default character is ^E.
	eof *char*	Send *char* if it is entered as the first character of a line and if TELNET is in line-by-line mode. The default character is The One Port's **eof** character—see **stty**.

Table 1. *TELNET Commands (cont.)*

Command	Argument(s)	Description
set (cont.)	erase *char*	Send the TELNET EC sequence if in localchars and character-at-time mode and the erase character is typed. The default character is The One Port's **erase** character— see **stty**.
	escape *char*	Set the character that puts TELNET into command mode. The default character is `^]`.
	flushoutput *char*	Send the TELNET AO sequence if in localchars mode and the flushoutput character is typed. The default character is `^O`.
	interrupt *char*	Send the TELNET IP sequence if in localchars mode and the interrupt character is typed. The default character is The One Port's **intr** character—see **stty**.
	kill *char*	Send the TELNET EL sequence if in localchars and character-at-time mode and the kill character is typed. The default character is The One Port's **kill** character—see **stty**.
	quit *char*	Send the TELNET BRK sequence if in localchars mode and the quit character is typed. The default character is `^\`.
status		Display information about the connection state, mode setting, local or remote echo, local or remote signal catching and the escape character.

Table 1. *TELNET Commands (cont.)*

Command	Argument(s)	Description
toggle	?	Print toggle command help.
	autoflush	Flush output on interrupt character reception. When TELNET AO, IP or BRK sequences are sent to the remote host (in response to the receipt of a corresponding interrupt character) and The One Port is in autoflush and localchars mode, TELNET does not display the data on your terminal (i.e., it is flushed) until the remote host acknowledges the TELNET sequence. The default setting is true.
	autosynch	Send (as urgent data) the TELNET AO or BRK sequences followed by a SYNCH sequence after an interrupt or quit character (respectively) has been typed if TELNET is in autosynch and localchars mode. The remote host should discard all previously typed data until both sequences are received and processed. The default setting is false.
	binary	Send and receive terminal data in binary (eight-bit) mode. The default setting is false.
	binaryin	Receive data from the terminal in binary mode. The default setting is false.
	binaryout	Send data to the terminal in binary mode. The default setting is false.

Table 1. *TELNET Commands (cont.)*

Command	Argument(s)	Description
toggle (cont.)	crmod	Map input CR to CR-LF. This applies only to characters received from the remote host. The default setting is false.
	localchars	Recognize control characters locally. The appropriate TELNET sequences are sent upon receipt of the flush, interrupt, quit, erase and kill characters. The default setting is true for line (line-by-line) mode and false for character (character-at-a-time) mode.
	localflow	Handle XON/XOFF flow control locally. The default setting is true.
	options	Display option negotiation. The default setting is false.

Table 1. *TELNET Commands (cont.)*

Negotiated Options

The TELNET protocol uses “negotiated options” to establish the services to be provided by each side of the network connection. Option negotiation has four commands: do, don’t, will and won’t.

One side of the connection (called a party) requests the other party to perform a given service with the “do” command (or, conversely, to not perform a service with the “don’t” command). The receiving party accepts or rejects a “do” request with a “will” or “won’t” response, respectively, and must always accept “don’t” requests.

A party offers to perform an option with the “will” command and receives a “do” or “don’t” response. A request to enter into a mode that is already set is not acknowledged.

The One Port sends the following requests and offers upon connection establishment:

DO	TELOPT_SGA	(suppress go ahead)
DO	TELOPT_ECHO	(echo)
WILL	TELOPT_TTYPE	(terminal type)
WILL	TELOPT_TSPEED	(terminal speed)
DO	TELOPT_STATUS	(status)

It responds to requests and offers as indicated in Table 2 below.

	Remote Party's Request or Offer			
Option	**do**	**don’t**	**will**	**won’t**
TELOPT_BINARY	will	won’t	do	don’t
TELOPT_ECHO	won’t	won’t	do	don’t
TELOPT_LFLOW	will	won’t	don’t	don’t
TELOPT_SGA	will	won’t	do	don’t
TELOPT_STATUS	won’t	won’t	do	don’t
TELOPT_TM	will	won’t	no reply	no reply
TELOPT_TSPEED	will	won’t	don’t	don’t
TELOPT_TTYPE	will	won’t	don’t	don’t
All others	won’t	won’t	don’t	don’t
Note: If The One Port is already in the requested mode, the request is not acknowledged.				

Table 2. *Responses to* TELNET *Options*

Connecting to a Host

The default port number is 23.

The **telnet** command does not perform the normal initial option negotiation when the *port* argument is present on the command line unless the port number is preceded with a hyphen (-) character.

telnet attempts to connect to a host up to five times with a delay progression of one, two, four and eight seconds between each attempt. These delay times do not include any possible TCP retransmission time-out delays.

Rlogin

The **rlogin** command establishes a login session with a remote host. After the connection has been established, the remote host rlogin server invokes the *login* program. Unlike **telnet**, you may then be able to log on without providing a password.

Password Validation

The One Port's user name (see **usrname**) is used by **rlogin** (unless the **-l** command line option is specified). If there is a user account on the remote host with an identical user name and The One Port's host name is in the remote computer's trusted host name file (e.g., `hosts.equiv` or `.rhosts`), then you can log on to the remote host without supplying a password. If The One Port's name is not in the file or the user name is not found, then the remote host's *login* process prompts you for a password.

Implementation Specifics

The One Port's implementation of rlogin has the following characteristics:

- The terminal type and window size settings (see **stty**) are sent to the remote host upon request.

- All character echoing is performed by the remote host. Flow control and interrupt characters entered at the terminal are sent to the remote host for processing.

- Only the disconnect escape sequence (“~.”) of the rlogin protocol sequence set is supported. The escape character (the default is the tilde character) can be set with the **-e** command line option. Using the disconnect escape sequence breaks the network connection without warning the remote host.

- The contact port number is 513.

- **rlogin** attempts to connect to a host up to five times with a delay progression of one, two, four and eight seconds between each attempt. These delay times do not include any possible TCP retransmission time-out delays.

- The processing of urgent data is treated differently depending on the TCP “urgdata” and the rlogin “flshurg” settings (see the **snet** command).

- The Kerberos authentication method is not supported.

Host-Initiated Connections

The One Port accepts one host-initiated (incoming) connection. It can be a connection to the command line interface (CLI TELNET) or a raw TCP socket connection for pass-through data.

CLI TELNET

Hosts can TELNET to The One Port (through port 23) to access the command line interface. All commands to configure terminal, network, and other system parameters are available; network and terminal access commands are disabled (namely, **lock**, **passwd**, **ppp**, **rlogin**, **slip**, and **telnet**)

The One Port must have an IP address before you can access the command line via TELNET.

The originating (host) TELNET should operate in local echo and line-at-a-time modes for best results.

CLI TELNET can be disabled with the **snet** command. To prevent users from accessing the command line from TELNET enter:

```
oneport% snet tcp icmd off
```

The default state is on.

You can test terminal settings modified during a CLI TELNET session with the **echo** command. The echo command sends text data to both the network and the terminal ports when run under CLI TELNET.

Raw Socket Connection

A raw TCP socket connection has no TELNET option negotiation, canonical character processing, or access to the command line interface. Data from the host is transmitted to the RS-232 (“Terminal”) connector and input characters are sent to the host. (See also the “RCON” section later in this chapter.)

The operational characteristics of a raw socket connection are modified through configuration settings.

- The default TCP port number is 3001. Use the **snet** command to change the port number. For example:

  ```
  oneport% snet tcp port 3000
  ```

- The One Port can be configured to disallow host-initiated connections. For example:

  ```
  oneport% snet tcp icon off
  ```

- The remote host can set several serial modes on The One Port before sending and receiving data. If the first 16-bit data word received by The One Port contains the value 0xDEED, The One Port’s terminal modes are set according to the value of the subsequent data word. The format of the serial configuration data words is:

Byte:	1	2	3	4
Fields:	Magic Number		Serial Settings	
Data:	0xDE	0xED	See Below	

The configuration data must be contained in the first four bytes of the first network packet after connection establishment; otherwise, the serial modes are left unchanged.

The serial mode bytes are defined as:

byte 3								byte 4							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ifc	any	ofc	hfc	csize		sbits		0	parity		baud				

The ifc values are:

Input S/W flow control	Value
Disabled	0x0
Enabled	0x1

The any (any character will restart output after output has been suspended) values are:

Any character resume	Value
Disabled	0x0
Enabled	0x1

The ofc values are:

Output S/W flow control	Value
Disabled	0x0
Enabled	0x1

The hfc values are:

H/W flow control	Value
Disabled	0x0
Enabled	0x1

The csize values are:

Character Size	Value

5	0x0
6	0x1
7	0x2
8	0x3

The sbits values are:

Stop Bits	Value
1	0x0
1.5	0x1
2	0x2
Invalid	0x3

The parity values are:

Parity	Value
None	0x0
Even	0x1
Odd	0x2
Invalid	0x3

The baud values are (be sure to enable hardware flow control when using baud rates above 9,600):

Baud Rate	Value
50	0x00
75	0x01
110	0x02
134	0x03
150	0x04
200	0x05
300	0x06
600	0x07
1200	0x08
1800	0x09
2400	0x0A
4800	0x0B
9600	0x0C
19200	0x0D
38400	0x0E
57600	0x0F
115200	0x10

Serial-Line Extender

The One Port can connect to another One Port (through the "TPE" modular outlet) to act as a serial-line extender. The One Port initiating the connection uses the **telnet** command (in raw mode). The **telnet** *port* parameter has the following modifiers to support raw mode:

r Run in raw mode without the initial TELNET option negotiation.

d Like the **r** modifier except the serial configuration data words are sent to the remote host (using the current **stty** settings).

The raw mode connection uses the following terminal modes as set by the **stty** command: software flow control (`ixon`, `ixoff`, `ixany`), hardware flow control (`rtscts`), modem control (`clocal`), baud rate, character size, stop bits, and parity generation. All other terminal modes are preset by **telnet** and cannot be modified.

A break sequence terminates the connection.

For example, to establish a connection to a remote One Port, you could enter:

```
oneport% telnet 198.211.125.220 d3001
```

This command establishes a raw connection at port number 3001 and sends the serial configuration words (so that the terminal modes of the remote One Port are the same as those of the host One Port).

LAN-to-LAN Connections

The One Port can serve as a link between two LANs when connected to another One Port (via the "Terminal" modular outlet). Prior to connecting The One Ports, you need to enter the destination network address on each unit with the **slip** command. For example:

```
oneport1% slip 128.10.0.0
oneport2% slip 128.22.0.0
```

(`oneport1` is on the 128.22 network; `oneport2` is on the 128.10 network.)

You can also use the default network address (although The One Port will send *all* network packets to the RS-232 port instead of just those packets intended for the destination network). For example:

```
oneport1% slip 0.0.0.0
oneport2% slip 0.0.0.0
```

In addition, the **stty** settings should be identical (e.g., the baud rate, character size, etc.) for each unit and, if an RS-232 cable is used between the units (as opposed to a modem), it must swap the signal lines (i.e., act as a null modem).

Implementation Specifics

The LAN-to-LAN link has the following operational restrictions and characteristics:

- The connecting networks (or subnetworks) cannot be the same.
- The connecting networks do not need to be of the same class.
- All incoming packets (from the “Terminal” port) that are not destined for the network to which The One Port is connected are dropped.
- Automatic modem connection, dialing, and chat scripts are not currently supported.

Example Configuration

To configure The One Port for LAN-to-LAN operation, you could enter the following commands:

```
oneport% stty rtscts clocal
oneport% initcmd “stty 115200; slip 0.0.0.0;
stty 9600”
oneport% silent -sane
```

The first command enables hardware flow control and does not allow modem signal changes to terminate the link. (Note that the link is only restarted when the power is cycled on The One Port.) The second command executes when the unit is turned off and on. It sets the transfer rate to 115200 baud; starts the connection; and lowers the baud rate to The One Port’s default baud rate when the connection terminates. The last command disables all standard output from The One Port.

In addition, you will need to add a default route to the host connected to The One Port's Ethernet connection specifying The One Port as a gateway. For example, with a Sun machine running SunOS, you would enter:

```
# route add net default oneport 6
```

Network Printing

The One Port can operate as a print server providing network hosts shared access to a serial printer. Two methods are available: **rcon** and LPD.

RCON

rcon, running on the host, establishes a TCP/IP connection to The One Port and transfers data between a host application and The One Port. Printing programs, such as `lp` or `lpr`, use the device created by **rcon** to forward files to the printer connected to The One Port's RS-232 ("Terminal") port. The **rcon** device allows printing programs to treat The One Port as a standard serial port instead of a network device. Therefore, users of the printing programs can use most of the features provided by these programs (such as fancy banner pages). Unfortunately, the **rcon** software needs to be installed and compiled on the host computer. The software tools (e.g., compiler) to accomplish this may not be installed or available. Contact Versalynx to obtain **rcon** source or object code. The **rcon** program syntax and documentation are found in Appendix B.

LPD

The One Port supports the widely used Internet printer protocol named Line Printer Daemon Protocol (LPD).

To configure The One Port to operate in a UNIX environment, you will need to add an entry in the printer database file (e.g., `/etc/printcap`). The entry should look something like the following:

```
# The One Port
op|The One Port:\
    :rm=oneport
```

The `rm` parameter is the name of The One Port as entered in the host name database file (e.g., `/etc/hosts`).

To print a file (`filename`) on a printer connected to The One Port, you could use the command:

```
lpr -Pop filename
```

The One Port's implementation of LPD has the following characteristics:

- Generally, the host sends the banner page after the file to be printed. Since The One Port does not queue print jobs, the banner page is printed last. If this is not desirable, run the file through a filter program first and then send it to the printer (or specify the filter program in the `printcap` file), or use the **rcon** program.

- The file will be printed using the tab expansion and line feed mapping as configured with `onlcr` and `tabs` parameters of the **stty** command.

- Hardware and software flow control are set as configured with the `rtscts`, `ixon`, `ixoff`, and `ixany` parameters of the **stty** command.

- The character size, parity generation, stop bits, and baud rate are as specified with the **stty** command.

Remote Network Access

The One Port can operate as a serial network or remote access server for a client host (such as a PC) that does not have the hardware or software necessary to connect directly to a TCP/IP-based network. The client establishes a serial connection (either directly or via a modem) to The One Port using the Serial Line Internet Protocol (SLIP) or the Point-to-Point Protocol (PPP) to access the network. The **slip** and **ppp** commands encapsulate IP datagrams for transmission over the serial connection between The One Port and the client host.

SLIP

Important aspects of The One Port’s implementation of SLIP are listed below:

- The maximum packet size is 1,006 octets.
- CSLIP (compressed SLIP), a version of SLIP used to compress TCP/IP datagram headers, is not supported.
- The **slip** command requires the user to provide (via the command line) the client IP address. **slip** displays both The One Port’s IP address and the client’s IP address for use by host chat scripts.

PPP

The One Port’s implementation of PPP has the following characteristics (in addition to those covered in the **ppp** command—see Appendix A) :

- TCP/IP datagram header compression is not supported.
- Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) are not supported.
- Old-style (RFC 1172) IP address negotiation is not supported.
- The maximum transmit unit (MTU) is 1,500 octets unless the client negotiates it to be smaller.
- A proxy ARP entry, for the client, is added to the ARP table; a route to the client is added to The One Port’s routing table.
- A connection can be initiated from a client connected to The One Port’s RS-232 (“Terminal”) port by invoking the **ppp** command; connections cannot be started from a device connected to the network port.

- The PPP counter and timer values are given below. The connection is closed if the counter values are exceeded.

Retransmission timeout (in seconds):	3
Maximum Terminate-Request transmissions:	2
Maximum Configure-Request transmissions:	10
Maximum Configure-Nak loops:	5

DNS

Domain Name Service (DNS) translates user-friendly host names (such as versalynx.com) into IP addresses. The One Port supports a simple DNS resolver that sends host and domain name queries to a domain name server for resolution. The resolver has the following characteristics:

- The maximum label length is 32 characters. (A label is defined to be a name that starts with a letter, ends with a letter or digit, and contains only letters, digits, or hyphens. Labels within a name are separated by dots.)

- The maximum default domain name length is 64 characters (see the **dname** command). The domain name consists of one or more labels.

- The maximum query name (the combination of the default domain name and the host name) length is 96 characters.

- A name is first searched in the host database. If it is not found, a DNS query is then sent to the domain name server (if the sever has been specified—see the **dserver** command). The default domain name is appended to names containing one label; multiple label names are queried as given. A trailing dot (the root domain) is always removed from the name.

- The One Port will wait up to five seconds for a response from the domain name server.

- Resolved queries are not stored in the host database (see the **hosts** command).

- The user commands utilizing the DNS resolver are: **nslookup**, **ping**, **rlogin**, and **telnet**.

DHCP

After The One Port is turned on, it may attempt to acquire its IP address using the Dynamic Host Configuration Protocol (see the **snet** and **ifconfig** commands).

The One Port's implementation of the Dynamic Host Configuration Protocol (DHCP) is detailed below:

- The One Port requests the following options in the `DHCPDISCOVER` packet:

  1. The subnet mask (see **ifconfig**)
  2. The default route (see **route**)
  3. The domain name server (see **dserver**)
  4. The host name to be assigned to The One Port (see **hosts**)
  5. The domain name (see **dname**)
  6. The route table (static routes; see **route**)
  7. An infinite lease time
  8. A maximum DHCP message size of 576 bytes

  The DHCP server does not need to provide any of the requested options (but must conform to the maximum message size).

  The One Port treats any lease time given by the server as an infinite lease. In addition, The One Port never issues a `DHCPRELEASE` packet. The One Port does, however, issue a `DHCPDISCOVER` packet whenever it is powered on (unless it has been assigned an IP address with the **ifconfig** command) and, therefore, can be given a new IP address at that time.

  If more than one option of the same type is given, only the first value of the duplicate option is accepted. Additional options (that is, those not listed above) offered by the server are ignored.

The One Port ignores domain name and host name option values that exceed The One Port's maximum length values for these items.

- DHCPDISCOVER packets are retransmitted with the following delay sequence until a response is received: 4, 8, 16, 32, and 64 second intervals and every 64 seconds thereafter.

- The One Port accepts the first valid DHCPOFFER packet (in the case of multiple servers responding to the DHCP-DISCOVER packet).

- A DHCPINFORM packet is not sent to the server (nor broadcast) when The One Port is assigned an IP address with the **ifconfig** command.

- All DHCP packets are silently discarded after The One Port is configured with an IP address.

SNMP

The One Port supports an agent implementation of the Simple Network Management Protocol (SNMP).

MIB-II Objects

The agent process accepts and responds to incoming SNMP requests from a Network Management Station (NMS) for all Management Information Base II (MIB-II) objects except for those of the Exterior Gateway Protocol (EGP). Each object in the EGP group has an access level set to "not-accessible." All other managed MIB-II objects have an access level of either "read-only" or "read-write" and are listed in the table found in Appendix C, "SNMP MIB-II Objects." An object's effective access mode can, however, be altered by the access mode of the community to which the object belongs (see "Community Profile" below).

Community Names

The One Port's community name table holds a maximum of four entries and has two default entries: "public" and "private." The access modes are read-write and read-only, respectively.

The table entries are displayed or modified with the **snmpcom** command.

A community's access mode applies to all The One Port's MIB-II objects (i.e., all objects are in the same community view).

Community Profile

The effective access mode of a MIB object is the most restrictive mode of either the community to which the object belongs or the access mode of the MIB object itself. This is referred to as the community profile. In relation to SNMP, the community profile defines which SNMP operators (get, get-next, set and trap) can be used to access a MIB object.

Trap Generation

The SNMP agent sends trap Protocol Data Units (PDUs) to network management stations listed in The One Port's manager trap table. The agent enters a manager's IP address and community name into the table after receiving a set-request PDU from the manager requesting trap generation (see the `snmpEnableAuthenTraps` MIB-II object in Appendix C). It removes a manager from the table after receiving a set-request PDU requesting that trap generation be disabled. In addition, network management stations can be deleted from the trap table with the **snmptrap** command.

The SNMP agent sends trap PDUs for the following events:

- `coldStart (0)`—any One Port reset condition
- `linkDown (2)`
- `linkUp (3)`
- `authenticationFailure (4)`

The manager trap table holds a maximum of four entries.

3

Network Protocol Specifications 4

The Internet reference model with the protocols and the user commands supported by The One Port is shown in Table 3.

Layer	**Protocols and User Commands**			
Application	**arp**	**ping**	DHCP, DNS, RIP, SNMP	LPD **rlogin**, **telnet**
Transport			UDP	TCP
Internet	ARP, RARP	ICMP	IP	
Network Interface	Ethernet, PPP, SLIP Device Drivers			
Hardware	RS-232, Twisted-Pair Ethernet			

Table 3. *The Internet Reference Model*

Table 4 lists important information regarding the implementation of these protocols on The One Port.

The specification values that can be modified with the **snet** command are indicated with an asterisk (*). The default values are given along with the protocol item's minimum and maximum values (in square brackets).

Protocol Specification	Description
ARP	
Maximum number of table entries	10
Maximum number of retries on pending entries	4; at one second intervals
Default table entry time-to-live (TTL)*****	10 minutes [1 second to 1 hour]
DHCP	See Chapter 3
DNS	See Chapter 3
Ethernet Device Driver	
Maximum number of transmission retries on collisions	15 using the 802.3 exponential backoff algorithm
Broadcast Address	FF:FF:FF:FF:FF:FF
ICMP	
Packet types generated	Echo request
Packet types processed	Echo reply, redirect message
IP	
Maximum datagram length	1,500 octets
Default TTL*****	128 gateway hops [1 to 999]
Default fragment reassembly TTL*****	1 minute [1 second to 1 hour]
Receive broadcast address	All ones or all zeros
Send broadcast address	Not used
Destination options	Not supported
PPP	See Chapter 3

Table 4. *Protocol Specifications*

Protocol Specification	Description
RARP	
Request messages	Sent every second during the first minute after power on and every minute thereafter until The One Port has an IP address (see the **snet** and **ifconfig** commands)
RIP	
Type	Passive—processes response messages only
Protocol port	520
Route TTL	3 minutes
Maximum number of routes	10
Maximum number of routes processed per packet	10
RLOGIN	See Chapter 3
SLIP	See Chapter 3
SNMP	
Protocol ports	161, 162
Versalynx private enterprise OID	895
MIB-II objects	See Chapter 3 and Appendix B
TCP	
Maximum number of connections	1
Maximum retransmission attempts	12; intersegment delay starts at 1/2 second and increases by 2*n* each interval to a maximum of 20 seconds. The connection is closed after all retransmission attempts fail

Table 4. *Protocol Specifications (cont.)*

Protocol Specification	Description
TCP (cont.)	
Persist time (window at zero)	∞ (segments are sent as long as required); intersegment delay starts at 1/2 second and increases by 2*n* each interval to a maximum of one minute
Default segment lifetime (MSL)*****	2 seconds [1 second to 1 hour]
Time delay after close	2 times the MSL Note: To handle possible delayed segments from a remote host, the TCP module restarts the close delay timer for every non-SYN segment (RESET is sent in response to a SYN segment). If the host continues to send non-SYN messages, the TCP connection continues to use The One Port's resources and prevents other uses of the TCP module
Default urgent data processing mode*****	BSD style
UDP	
Read time-out	0.2 seconds
Well-known protocol ports	161 (SNMP), 162 (SNMP-trap), and 520 (RIP)
Valid protocol port range	2,050–65,535

Table 4. *Protocol Specifications (cont.)*

RS-232 Terminal Cable

The RS-232 terminal cable has an RJ-45 connector on one end and a D-Subminiature connector on the other.

You can use cables with four-, six- or eight-pin RJ-45 connectors to plug into The One Port's modular outlet labeled "Terminal." A four-pin (wire) connector supports only the transmission and reception of data. A six-pin connector supports data input and output and hardware flow control. An eight-pin connector supports full signaling (including modem control). All connector configurations handle software flow control. The **stty** command enables and disables both hardware and software flow control.

The pin-outs of the three RJ-45 connector configurations are shown in Table 5 and the RJ-45 connector pin positions are shown in Figure 3.

RJ-45 Pin #	**Description**	**8-pin**	**6-pin**	**4-pin**
1	Modem control output	X		
2	H/W flow control input	X	X	
3	Signal ground	X	X	X
4	Data output	X	X	X
5	Data input	X	X	X
6	Carrierdetect	X	X	X
7	H/W flow control output	X	X	
8	Modem control input	X		

Table 5. *RJ-45 Connector Pin-Out*

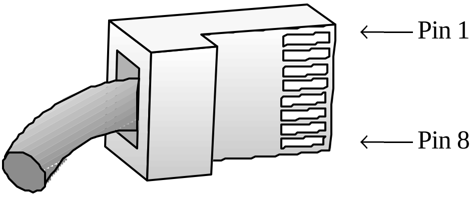

Figure 3. *RJ-45 Pin Positions*

The RJ-45 to D-Subminiature connector wiring depends on the type of device connecting to The One Port—Data Terminal Equipment (DTE) or Data Communication Equipment (DCE). The signal direction determines the device type. In most cases, The One Port connects to a DTE device (such as a PC or a terminal). The One Port to DTE device cable wiring is shown in Tables 6 and 7. Table 8 gives the cable wiring necessary to connect The One Port to a DCE device.

RJ-45 Pin #	**Signal Name**	**Description**	**DB25 Pin #**	**DB9 Pin #**	**The One Port**	**DTE PC**
1	DSR	Data Set Ready	6	6	Output	Input
2	RTS	Request to Send	4	7	Input	Output
3	GND	Signal Ground	7	5	—	—
4	RXD	Receive Data	3	2	Output	Input
5	TXD	Transmit Data	2	3	Input	Output
6	CD	Carrier Detect	8	1	Input	—
7	CTS	Clear to Send	5	8	Output	Input
8	DTR	Data Term Ready	20	4	Input	Output

Table 6. *The One Port-to-DTE Cabling (with RTS/CTS hardware flow control)*

RJ-45 Pin #	Signal Name	Description	DB25 Pin #	DB9 Pin #	The One Port	DTE Terminal
1	CTS	Clear to Send	5	8	Output	Input
2	DTR	Data Term Ready	20	4	Input	Output
3	GND	Signal Ground	7	5	—	—
4	RXD	Receive Data	3	2	Output	Input
5	TXD	Transmit Data	2	3	Input	Output
6	CD	Carrier Detect	8	1	Input	—
7	DSR	Data Set Ready	6	6	Output	Input
8	RTS	Request to Send	4	7	Input	Output

Table 7. *The One Port-to-DTE Cabling*
(with DTR/DSR hardware flow control)

RJ-45 Pin #	Signal Name	Description	DB25 Pin #	DB9 Pin #	The One Port	DCE Modem
1	DTR	Data Term Ready	20	4	Output	Input
2	CTS	Clear to Send	5	8	Input	Output
3	GND	Signal Ground	7	5	—	—
4	TXD	Transmit Data	2	3	Output	Input
5	RXD	Receive Data	3	2	Input	Output
6	CD	Carrier Detect	8	1	Input	—
7	RTS	Request to Send	4	7	Output	Input
8	DSR	Data Set Ready	6	6	Input	Output

Table 8. *The One Port-to-DCE Cabling*
(with RTS/CTS hardware flow control)

Unshielded Twisted-Pair Ethernet Cable

The modular outlet labeled “TPE” connects The One Port to a network (usually via a concentrator or a hub). The unshielded twisted-pair cable wiring is shown in Table 9 below.

RJ-45 Pin #	**Signal Name**	**Description**	**RJ-45 Pin #**
1	TD+	Transmit data	1
2	TD-	Transmit data	2
3	RD+	Receive data	3
4	N/C	No connect	4
5	N/C	No connect	5
6	RD-	Receive data	6
7	N/C	No connect	7
8	N/C	No connect	8

Table 9. *Straight UTP Cable Wiring*

To connect The One Port directly to a single Ethernet device that does not swap signals (such as a computer), use a “swapped” or crossed twisted-pair cable. The wiring is shown in Table 10 below.

RJ-45 Pin #	**Signal Name**	**Description**	**RJ-45 Pin #**
1	TD+	Transmit data	3
2	TD-	Transmit data	6
3	RD+	Receive data	1
4	N/C	No connect	4
5	N/C	No connect	5
6	RD-	Receive data	2
7	N/C	No connect	7
8	N/C	No connect	8

Table 10. *Swapped UTP Cable Wiring*

⚠ The RS-232 and Ethernet cables should have non-conductive protective boots (or sleeves) over the RJ-45 connectors to remove electrostatic discharge susceptibility.

Troubleshooting 6

This chapter lists several potential problems and their possible causes.

1. **The power status indicator does not light.**

   - The external power supply is not attached or is inoperable.
   - The power supply is not plugged into a powered outlet.
   - The power switch is not in the ON position.
   - The One Port is not functional. Return it to Versalynx for repair (see the “Limited Warranty” statement at the beginning of this manual for warranty and return information).

2. **The One Port is turned on but nothing is displayed on the terminal.**

   - The terminal is not turned on.
   - The RS-232 cable is not the correct type (DCE or DTE), is connected to the wrong outlet on The One Port or on the terminal, or is faulty.
   - The terminal modes are set incorrectly (see the **stty** command). If you see control characters or other miscellaneous characters on your terminal screen, the terminal baud rate or character size may be set incorrectly or the terminal cable may not be attached.
   - Output is disabled (see the **silent** command).

3. **Ping cannot communicate to another network device.**

   - The One Port does not have a viable IP address.
   - The twisted-pair cable is not the correct type (straight or crossed). See the “Unshielded Twisted-Pair Ethernet Cable” section of Chapter 5.

- The **ping** IP address is incorrect. Verify that the host database entries are correct if you are using **ping** with a host name. Use **arp** to validate that The One Port has been able to associate the IP address with an Ethernet address. If the IP address of the host you are trying to reach does not have a valid Ethernet address in the ARP table, then The One Port's network ARP request did not receive a reply.
- The route table entries (**route**) or the network masks (**ifconfig**) are not compatible between The One Port and the destination device. Verify each device is "reachable" from the other and that they are in the same subnet (if applicable).
- The One Port is not connected to the network. Verify that the link integrity LED is on. Check the twisted-pair cable and the network connections.
- The network is not operational or the link states of The One Port and the corresponding hub device or host computer are set at opposite values.

4. Rlogin returns "cannot connect to *host*".

- Verify that the items in number three above do not apply.
- Verify that the TCP and rlogin urgent settings are correct (see the **snet** command). Set the TCP "flshurg" value to "off" and retry the connection.

Appendix A—User Commands

This appendix provides an alphabetical listing of the user commands. The following typographic conventions are used:

[ ] Items in square brackets are optional.

( ) Parentheses group command elements and are used to clarify command syntax.

... Ellipses indicate that an argument can be repeated multiple times.

bold Bold type items must be typed as shown. They are commands, option indicators, or keywords.

italic Italic type indicates a user-defined command line argument.

All commands display the usage (help) message when given a question mark (**?**) as the only argument.

NAME

arp - display or modify the Internet-to-Ethernet table

SYNOPSIS

arp [*host*]
arp [**-f**] [(**add** *host eaddr* [**perm**]) | (**delete** *host*)]

DESCRIPTION

arp displays all ARP table entries when invoked without arguments. If *host* is given, **arp** displays the table entry for *host*.

The **-f**, **add** and **delete** options modify the ARP table as described below.

OPTIONS

-f Flush the ARP table. If this option is used with the **add** option, **arp** first removes all table entries before adding the newly defined entry.

add Create an ARP table entry that maps *host* to Ethernet address *eaddr*. *host* is a host name or an IP address in dotted decimal notation. *eaddr* is represented with six hexadecimal bytes separated by colons (e.g., `00:20:D0:00:3F:57`). **perm** makes the entry permanent.

delete Remove *host* from the ARP table.

SEE ALSO

SNMP MIB-II `atTable` object in Appendix C
SNMP MIB-II `ipNetToMediaTable` object in Appendix C

NAME

dname - display or set the domain name

SYNOPSIS

dname [*name*]

DESCRIPTION

dname displays the default domain name when invoked without the *name* argument; otherwise, the domain name is set to *name*.

The domain name is appended onto host names (if the host name is a single label) and used for name resolution in the **nslookup**, **ping**, **rlogin**, and **telnet** commands.

Setting *name* to a null string (“”) deletes the default domain name.

NOTES

The maximum domain name length is 64 characters. The maximum length of a label within the domain name is 32 characters. The domain name consists of one or more labels.

SEE ALSO

dserver, **nslookup**
“DNS” section of Chapter 3

NAME

dserver - display or set the domain name server

SYNOPSIS

dserver [**delete |** *ipaddress*]

DESCRIPTION

dserver displays or sets the domain name server. *ipaddress* is the IP address (in dotted decimal notation) of the host to which DNS name queries should be sent.

OPTIONS

delete Delete the currently assigned domain name server.

SEE ALSO

dname, **nslookup**
“DNS” section of Chapter 3

NAME

echo - print arguments to the screen

SYNOPSIS

echo [*argument* …]

DESCRIPTION

echo displays the arguments on the screen. This command is used in conjunction with the **initcmd** command to inform the user of the initial operational characteristics after The One Port has been turned on.

SEE ALSO

initcmd

NAME

help - display the command list

SYNOPSIS

help [**-v**]
ls [**-v**]
? [**-v**]

DESCRIPTION

help displays the available commands.

The syntax of an individual command is displayed when the command is given a question mark (**?**) as the only argument.

OPTIONS

-v List the command names and a one-line description of each (verbose).

NAME

hosts - associate a host name with an IP address

SYNOPSIS

hosts [**-f**] [(**add** *ipaddress host*) | (**delete** *host*)]
hosts [*oneport_name*]

DESCRIPTION

hosts associates a host name with an Internet Protocol address. It displays the host database entries when invoked without arguments. The second usage of **hosts** sets the host name of The One Port.

The **-f**, **add** and **delete** options modify the host database as described below.

OPTIONS

-f Flush the host database. If this option is used with the **add** option, **hosts** first removes all host entries from the host database before adding the new host name and address.

add Add the *host* and associated *ipaddress* to the host database. If a host has more than one name (a "nickname"), you can execute **hosts** for each nickname to add the nickname into the host database. *ipaddress* is given in dotted decimal notation.

delete Delete the *host* entry from the host database.

NOTES

The host database holds a maximum of 11 entries. Two entries, however, are reserved for The One Port and the local host.

Each host name can be a maximum of 32 characters in length.

The system prompt is set to the *oneport_name* (the default is `oneport`).

oneport_name can also be set by an NMS using the SNMP protocol since it is equivalent to the MIB-II `sysName` object.

SEE ALSO

ifconfig
SNMP MIB-II `ipAddrTable` object in Appendix C
SNMP MIB-II `sysName` object in Appendix C

NAME

ifconfig - display or set network interface information

SYNOPSIS

ifconfig [**ec0** | **lo0**]
ifconfig [*ipaddress* [*mask*]]

DESCRIPTION

The first usage of **ifconfig** displays the current state and configuration information of both network interfaces if no arguments are specified; otherwise, it displays the information of the network interface given on the command line.

The state information includes:

- the interface state—UP (enabled)
- the interface type (e.g., LOOPBACK)
- the allocation of system resources (RUNNING)
- the trailer usage (NOTRAILERS)

The configuration information includes:

- the Internet (inet) address of The One Port in dotted decimal notation
- the network mask (netmask) where ones in the address bits indicate the network portion of the address and zeros indicate the host portion of the address
- the Ethernet address (ether) of The One Port

The second usage of **ifconfig** sets The One Port's IP address and, optionally, the network mask. *ipaddress* contains the network number and a unique host number (for The One Port). If *ipaddress* is the only argument specified, **ifconfig** sets the network mask according to the network address class. If both *ipaddress* and *mask* are specified on the command line, *mask* determines the subnet field of the host portion of the address. *mask* contains ones in the bit positions corresponding to the standard network portion, and ones and zeros in the host portion of the address. The bits that are set to one in the host portion of the address specify the subnet field. The subnet field should be contiguous with the network portion.

ipaddress is given in dotted decimal notation. *mask* is specified in dotted decimal notation or as a hexadecimal number preceded with 0x.

OPTIONS

ec0 Display the Ethernet interface status and configuration.

lo0 Display the loopback interface status and configuration.

EXAMPLE

```
oneport% ifconfig
lo0: flags=49<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1 netmask FF000000
ec0: flags=61<UP,NOTRAILERS,RUNNING>
        inet 192.10.160.3 netmask FFFFFF00
        ether 00:20:D0:00:3F:57
```

NOTES

The **lo0** interface state is always `UP`. The **ec0** interface state is set to `UP` after a valid IP address has been assigned to The One Port. The state cannot be set to `DOWN`.

Since The One Port does not send broadcast packets, outgoing broadcast addresses are not supported.

The One Port will not attempt to acquire its IP address with either DHCP or RARP after being assigned an IP address with this command.

SEE ALSO

hosts, **snet**
SNMP MIB-II `ifTable` object in Appendix C

NAME

initcmd - display or set the system initialization command(s)

SYNOPSIS

initcmd [**“**[**-**]*cmd* [; [**-**]*cmd*] …**”** [**power**] [**dtr**]]

DESCRIPTION

initcmd displays the system initialization command string and execution conditions when invoked without arguments; otherwise, it sets the initialization command string.

Immediately after The One Port is turned on (or after the modem control input line is asserted), it executes the commands specified in the initialization command string. *cmd* is any valid user command. Commands are separated with a semi-colon (;). The One Port does not display the command or the system prompt if the command is preceded with a hyphen (-) character. The initialization command string cannot exceed 255 characters in length.

OPTIONS

power Run the commands after power on (default).

dtr Run the commands after assertion of the modem control input line.

EXAMPLE

To enable The One Port to automatically log on to a remote server after it is turned on, you should assign an IP address to The One Port with the **ifconfig** command, set up the host database with the **hosts** command and then set the system initialization command string with the **initcmd** command:

```
oneport% initcmd "-echo 'Logging on to sun4';-rlogin
sun4"
```

To remove initialization commands from the command string, type:

```
oneport% initcmd ""
```

SEE ALSO

echo, **lock**, **wait**

NAME

link - display or set the link integrity test state

SYNOPSIS

link [**on** | **off**]

DESCRIPTION

Each network device on a twisted-pair Ethernet network should send a link test pulse at regular intervals. Since not all devices conform to the IEEE 802.3 10BASE-T standard, **link** allows you to set the link integrity test state of The One Port.

Without arguments, **link** reports the link integrity test state and the state of the network connection (up or down). When the link state is **off**, **link** always reports that the network connection is up since the reception of link test pulses is not expected or required. When the test state is **on**, **link** reports that the network link state is up if The One Port is receiving link test pulses; otherwise, **link** reports that the link state is down.

OPTIONS

on Enable the link integrity test state. The One Port generates link test pulses and requires the connecting device to send test pulses in order to have a valid network connection.

off Disable the link integrity test state. The One Port does not generate link test pulses.

EXAMPLE

```
oneport% link
link integrity test is on
link is up
```

SEE ALSO

“TPE Link Test” section of Chapter 3

NAME

lock - lock the terminal

SYNOPSIS

lock

DESCRIPTION

lock clears the screen and prompts for a password to unlock the terminal. A carriage return terminates the password. The **passwd** command specifies the lock password.

If The One Port is reset with the terminal lock active, the commands given in the initialization string (see the **initcmd** command) are executed after the terminal is unlocked.

EXAMPLE

```
oneport% lock
   ...
enter password to unlock terminal
```

SEE ALSO

initcmd, **passwd**

NAME

lockmem - display or set the system parameters lock state

SYNOPSIS

lockmem [**on** | **off**]

DESCRIPTION

Without arguments, **lockmem** reports the lock state of the system parameters. When the lock state is **off** (unlocked), the system parameters are reset to the factory default values after The One Port is turned off and on five consecutive times without keyboard input. Otherwise, the system parameters are not affected by power cycles (i.e., the settings are retained).

OPTIONS

on The system parameters are not affected by power cycles.

off The system parameters are reset to the factory default values after five power cycles.

SEE ALSO

system
"Non-Volatile Memory" section of Chapter 3

NAME

nslookup - query the domain name server

SYNOPSIS

nslookup *host*

DESCRIPTION

nslookup queries the domain name server for the IP address of *host* and displays the result.

The default domain name is appended to *host* if *host* is a single label or name (i.e., it has no dot separators).

NOTES

The host database is not queried. All requests are sent to the domain name server.

The queried name (the result of the host name appended with the default domain name) cannot exceed 96 characters in length.

SEE ALSO

dname, **dserver**
“DNS” section of Chapter 3

NAME

passwd - set the terminal lock password

SYNOPSIS

passwd

DESCRIPTION

passwd sets the password used to unlock the terminal (see the **lock** command).

The terminal lock feature prevents other users from using your terminal and logging on to remote hosts (with your user name and host database information) while the terminal is unattended.

The lock password is removed by:

- repeatedly resetting The One Port as described in the Non-Volatile Memory” section of Chapter 3.
- entering carriage returns in response to the two **passwd** prompts

The lock password can be up to 32 characters in length and is case sensitive. Longer passwords provide better security.

EXAMPLE

```
oneport% passwd
setting the terminal lock password
new password:
retype new password:
password changed
```

SEE ALSO

lock, **usrname**

NAME

ping - send ICMP ECHO_REQUEST datagrams

SYNOPSIS

ping *host* [*timeout*]
ping -s *host* [*packetsize*] [*count*]

DESCRIPTION

ping sends an ICMP ECHO_REQUEST datagram to the specified host (or network device) and expects an ECHO_RESPONSE datagram in return.

The first form of **ping** waits up to *timeout* seconds for a response from *host*. If the host responds, **ping** displays "*host* is alive;" otherwise, **ping** displays "no answer from *host*." The default *timeout* value is 20 seconds. *host* is given in dotted decimal notation or by name.

The second form of **ping** sends one ICMP ECHO_REQUEST datagram every second and displays a status line for each ECHO_RESPONSE datagram received. The status line includes the number of bytes received, the host name or IP address, the ICMP sequence number and the round-trip time. No status is displayed if the host does not respond. At the completion of the command, **ping** displays a summary of the packets sent, the packets received, the percentage packet loss and the packet round-trip time statistics.

The default datagram *packetsize* is 64 bytes (56 bytes of data plus eight bytes of ICMP packet header). **ping** sends ten packets unless the *count* argument is present on the command line.

OPTIONS

-s Send one ICMP ECHO_REQUEST datagram every second.

EXAMPLE

```
oneport% ping -s sun4 56 3
PING sun4:  56 data bytes
64 bytes from sun4: icmp_seq=0 time=0 ms
64 bytes from sun4: icmp_seq=1 time=0 ms
64 bytes from sun4: icmp_seq=2 time=0 ms

----sun4 PING Statistics----
3 packets transmitted, 3 packets received, 0%
packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

NOTES

Round-trip times under 50 milliseconds in duration are displayed as zero milliseconds.

NAME

ppp - point-to-point protocol daemon

SYNOPSIS

ppp [**ipaddress**] [**-acfc**] [**-magic**] [**-map** *n*] [**-mru** *n*] [**-pfc**]

DESCRIPTION

ppp encapsulates IP datagrams for transmission to a serial device connected to The One Port's "Terminal" port.

ppp uses the baud rate, hardware flow control, and modem change detection settings as configured with the **stty** command.

OPTIONS

ipaddr

Negotiate the client's (such as a PC) IP address to be **ipaddr**. When this parameter is provided, **ppp** will not accept an IP address from the client. If **ipaddr** is not provided, the client's IP address is obtained from the client.

-acfc

Disable address/control field compression (both send and receive).

-magic

Disable magic number negotiation (both send and receive).

-map *n*

Negotiate the asynchronous character map to *n*. **ppp** requests the client escape the characters corresponding the map bits. *n* is a 32-bit hexadecimal number preceded with 0x. Bit zero (0x00000001) represents the ASCII NUL character (0x00) and bit 31 (0x80000000) represents the ASCII US character (0x1F). The default is to negotiate no escaping of characters.

-mru *n*

Negotiate the Maximum Receive Unit (MRU) value to *n*. The minimum value is 128 bytes; the maximum is 1,500 bytes. The default is 1,500 though a value of 296 (40 bytes for the TCP/IP headers and 256 bytes for data) is suggested for slower connections.

-pfc Disable protocol field compression (both send and receive).

EXAMPLE

```
oneport% ppp 192.10.160.220 -mru 296
```

SEE ALSO

slip
"PPP" section of Chapter 3

NAME

rlogin - log on to a remote host

SYNOPSIS

rlogin [**-8**] [**-e** *char*] [**-l** *username*] *host*

DESCRIPTION

rlogin establishes a login session with a remote host. The remote host permits you to log on without a password if The One Port is considered a “trusted host” and if the host has a login name identical to *username* (if the **-l** option is used) or to the system user name.

OPTIONS

-8 Pass eight-bit data across the network. The default is to use seven-bit data.

-e *char* Set the escape character used in the escape sequence to disconnect **rlogin** from the remote host. The default character is the tilde (i.e., the escape sequence is “~.”).

-l *username*
Set the user name. This overrides the system user name.

SEE ALSO

telnet, **usrname**
“Rlogin” section of Chapter 3

NAME

route - display or modify the routing table

SYNOPSIS

route [**-n**]
route [**-f**] **add** [**host** | **net**] *destination gateway metric*
route [**-f**] **delete** [**host** | **net**] *destination gateway*

DESCRIPTION

route displays or modifies the routing table entries. Normally, the table entries are created or modified through the **ifconfig** command or the RIP protocol. The entries may also be changed by an NMS using the SNMP protocol. Only routes entered by this command are saved in non-volatile memory.

destination and *gateway* are specified in dotted decimal notation or by name (except the default route, 0.0.0.0, can also be specified as a destination with the key word **default**). The maximum value for *metric* is 16 (which represents an infinite metric).

OPTIONS

-f Flush the route table except for the local host and The One Port entries. If this option is used with the **add** option, **route** first removes the route entries before adding the newly defined route.

-n Display names in dotted decimal notation.

add Create a route to *destination*. If *destination* has a local part of zero and **host** or **net** is not specified, then **route** treats *destination* as a network; otherwise, it considers *destination* a host. *metric* is the number of gateway hops to *destination*. If *destination* is on a directly attached network, *metric* must be zero.

delete Remove the route to *destination*.

host *destination* is a host.

net *destination* is a network.

EXAMPLE

To add a default route using host 192.10.160.200 as the gateway, enter the following command:

```
oneport%    route    add    net    default
192.10.160.200 1
```

SEE ALSO

hosts, **ifconfig**
SNMP MIB-II `ipRouteTable` object in Appendix C

NAME

silent - display or modify system message output

SYNOPSIS

silent [[-]**banner**] [[-]**command**] [[-]**prompt**] [[-]**sane**]

DESCRIPTION

silent enables or disables system message output. A parameter preceded with the hyphen (**-**) character disables the type of output specified.

OPTIONS

[-]banner	Welcome banner.
[-]command	Command output.
[-]prompt	Command line prompt.
[-]sane	Welcome banner, command output, command line prompt and character echo (`stty -echo`).

EXAMPLE

To disable all terminal output, enter the following two commands:

```
oneport% stty -echo
oneport% silent -banner -command -prompt
```

or:

```
oneport% silent -sane
```

SEE ALSO

stty

NAME

slip - serial line Internet Protocol daemon

SYNOPSIS

slip ipaddr

DESCRIPTION

slip encapsulates IP datagrams for transmission to a serial device connected to The One Port's "Terminal" port.

slip uses the baud rate, hardware flow control, and modem change detection settings as configured with the **stty** command.

OPTIONS

ipaddr	The client's (such as a PC) IP address.

EXAMPLE

slip displays the server (The One Port) and client addresses for possible chat script usage:

```
oneport% slip 192.10.160.220
server address is 192.10.160.3
your address is 192.10.160.220
```

SEE ALSO

ppp
"LAN-to-LAN" section of Chapter 3
"SLIP" section of Chapter 3

NAME

snet - display or modify network protocol parameters

SYNOPSIS

snet [(**ARP**|**IP**|**TCP**|**rlogin**) [*option value* ...]]

DESCRIPTION

snet displays or sets various network protocol parameters. With no arguments or with a single protocol argument, **snet** reports the current values of all the protocols or of the specified protocol, respectively.

OPTIONS

Each protocol and its available options are listed below. If two options of the same protocol and type are given on the command line, the last one is used (set).

ARP	**ttl**	Set the time-to-live duration for new ARP table entries. The valid range is from one to 3,600 seconds.
IP	**addr**	Set the automatic IP address acquisition method. The valid settings are: **none**, **dhcp**, **rarp**, or **both** (DHCP and RARP). If the IP address has been set with **ifconfig**, The One Port will not attempt to acquire an address through DHCP or RARP.
	rttl	Set the datagram fragment reassembly time-to-live duration. The valid range is from one to 3,600 seconds. See also the MIB-II `ipReasmTimeout` object in Appendix C.
	ttl	Set the datagram time-to-live duration. The valid range is from one to 999 gateway hops. See also the MIB-II `ipDefaultTTL` object in Appendix C.

TCP **icmd** Accept an incoming TELNET connection to the command line interface. The valid settings are **on** or **off**.

icon Accept an incoming raw TCP socket connection. The valid settings are **on** or **off**.

idle Set the maximum time before a raw TELNET or socket connection is automatically shutdown due to no packet activity. The valid range is from zero to 3,600 seconds. Zero means no timeout.

msl Set the maximum segment lifetime. The valid range is from one to 3,600 seconds.

port Set the TCP port number on which an incoming connection is accepted.

recon Automatically reestablish a lost or broken TELNET session. The valid settings are **on** or **off**.

urgdata Set the urgent data processing mode. When set to **one**, the received packet should have at most one octet containing the urgent data and the urgent pointer should reference the octet one position beyond the urgent data (BSD style). When set to **many**, TCP processes all data up to and including the data pointed to by the urgent pointer as urgent data (non-BSD style).

rlogin **flshurg** Set the TCP flushing mode. When set to **on**, TCP starts flushing all data as soon as a packet indicates that there is urgent data in the current packet or in a subsequent packet. This mode provides improved rlogin response time to the interrupt character when a program is sending large amounts of data to the screen. The valid settings are **on** and **off**.

urgcnt Set the number of urgent data octets that must be received before starting **flshurg** (if **on**). **urgcnt** should be set to at least three octets to allow the remote host rlogin server to send several initialization commands. The valid range is from one to ten.

EXAMPLE

The command

```
snet tcp urgdata many
```

sets the TCP urgent data processing mode to non-BSD style.

NOTES

The default settings are:

```
ARP:     ttl      =  600  seconds
IP:      addr     = both
         rttl     =   60  seconds
         ttl      =   16  gateway hops
TCP:     icmd     =   on
         icon     =   on
         idle     =    0  seconds
         msl      =    2  seconds
         port     = 3001
         recon    =  off
         urgdata  =  one  octet per packet
rlogin:  flshurg  =  off
         urgcnt   =    3  octets
```

SEE ALSO

ifconfig
“Host-Initiated Connections” section of Chapter 3
“Rlogin” section of Chapter 3
Chapter 4—“Network Protocol Specifications”
Appendix C—SNMP MIB-II Objects

NAME

snmpcom - display or modify the SNMP community name table

SYNOPSIS

snmpcom [**-f**] [(**add** *community* (**rw**|**ro**)) | (**delete** *community*)]

DESCRIPTION

snmpcom displays the SNMP community names and access modes when invoked without arguments; otherwise, the community name table is modified by the options described below.

community can be up to 32 characters in length. The access mode is either read-write (**rw**) or read-only (**ro**). The community table holds a maximum of four entries.

The default community table has two entries:

Community Name	Access Mode
“public”	read-write
“private”	read-only

OPTIONS

-f Flush the community name table. If this option is used with the **add** option, **snmpcom** removes all table entries before adding the new community name and access mode.

add Add the *community* name and access mode to the community name table.

delete Delete the *community* entry from the community name table.

SEE ALSO

“SNMP” section of Chapter 3
Appendix C—SNMP MIB-II Objects

NAME

snmpcon - display or set the MIB-II `sysContact` object value

SYNOPSIS

snmpcon [*contact*]

DESCRIPTION

snmpcon displays the MIB-II `sysContact` object value when invoked without the *contact* argument; otherwise the MIB-II object is set to *contact*.

contact can be a maximum of 32 characters in length.

SEE ALSO

“SNMP” section of Chapter 3
SNMP MIB-II `sysContact` object in Appendix C

NAME

snmploc - display or set the MIB-II `sysLocation` object value

SYNOPSIS

snmploc [*location*]

DESCRIPTION

snmploc displays the MIB-II `sysLocation` object value when invoked without the *location* argument; otherwise the MIB-II object is set to *location*.

location can be a maximum of 32 characters in length.

SEE ALSO

"SNMP" section of Chapter 3
SNMP MIB-II `sysLocation` object in Appendix C

NAME

snmptrap - display or modify the SNMP manager trap table

SYNOPSIS

snmptrap [**-f**] [**delete** *host*]

DESCRIPTION

snmptrap, when invoked without arguments, displays the network managers (hosts) and their community names that are regis tered to receive SNMP traps; otherwise, the manager trap table is modified as described by the options below.

A network manager is placed in the table (registered) when it sends a set-request PDU to enable trap PDU generation. It is removed from the table when The One Port receives a set-request PDU to disable trap PDU generation. The table holds a maximum of four host entries.

OPTIONS

-f Flush the manager trap table.

delete Delete *host* from the manager trap table. *host* is specified in dotted decimal notation or by name.

NOTES

The SNMP agent sends trap PDUs to the network managers listed in the manager trap table for the following events:

- `coldStart (0)`—any One Port reset condition
- `linkDown (2)`
- `linkUp (3)`
- `authenticationFailure (4)`

SEE ALSO

"SNMP" section of Chapter 3
SNMP MIB-II `snmpEnableAuthenTraps` object in Appendix C

NAME

stty - display or set the terminal modes

SYNOPSIS

stty [*option …*]

DESCRIPTION

stty sets the terminal modes. Without arguments, **stty** reports the current terminal settings. (Note: Memory must be unlocked before the modes can be changed—see the **lockmem** command.)

OPTIONS

The options listed below are independent of each other. If two options of the same type are given on the command line, the last one is used (set). An option is disabled when preceded with the hyphen (-) character.

Control characters:

intr erase kill eof start stop
Set the special characters for the interrupt signal, erase character, erase line, end-of-file, resume output and suspend output functions, respectively.

Input modes:

[-]brkint
Signal interrupt on break.

[-]icrnl
Map ASCII CR to NL on input.

[-]ignbrk
Ignore break conditions.

[-]ignpar
Ignore characters with parity errors.

[-]istrip
Strip input characters to 7-bits.

[-]iuclc
Map upper-case characters to lower-case characters on input.

[-]ixany
Allow any input character to resume output.

[-]ixoff
Enable start/stop (software) input flow control.

[-]ixon
Enable start/stop (software) output flow control.

Local modes:

[-]echo
Enable the echoing of input characters to the terminal.

[-]echoctl
Echo control characters as `^char` and the delete character as `^?`.

[-]echoe
Echo the erase character as BS-SP-BS.

[-]echok
Echo the NL character after reception of a **kill** character.

[-]echoke
Erase the entire input line after reception of the **kill** character using BS-SP-BS.

[-]echonl
Echo the NL character.

[-]icanon
Enable canonical input processing of **erase** and **kill** characters, line delineation by **eof** and NL characters, and **echoctl**, **echoe**, **echok**, **echoke**, **echonl** echo modes.

[-]isig
Enable **intr** signal.

Output modes:

[-]olcuc
Map lower-case characters to upper-case characters on output.

[-]onlcr
Map NL to ASCII CR-NL on output.

[-]tabs

Expand tabs to spaces. Tab stops are set at eight character intervals.

Terminal settings:

50 75 110 134 150 200 300 600 1200 1800 2400 4800 9600 19200 38400 57600 115200

Set the terminal baud rate to the given speed.

[-]clocal

Ignore modem control changes. When this mode is disabled (**-clocal**), modem line changes will terminate commands such as **telnet** and **ppp**. In addition, characters are not sent out the “Terminal” port unless the modem control input line is active.

cols

Set the number of columns supported by the terminal.

[-]cremote

The modem control output line is asserted when a raw TELNET or socket connection exists. In this mode, the line is normally wired to the connecting device’s DCD signal.

cs5 cs6 cs7 cs8

Set the character size.

even odd none space mark

Set the type of parity generation and detection.

rows

Set the number of rows supported by the terminal.

[-]rtscts

Enable hardware flow control.

sane

Set all modes to the default values except for the baud rate, **term**, **rows** and **cols** settings.

sb1 sb1.5 sb2

Set the number of stop bits per character. Note: **sb1.5** is only valid with **cs5**; **sb2** is valid will all character sizes except **cs5**.

term

Set the terminal type.

EXAMPLE

The command

```
        oneport% stty term vt100 -istrip erase ^?
```

sets the terminal type to vt100, disables the stripping of input characters and sets the erase character to DEL (delete).

NOTES

The default settings are:

```
term settings:  wyse50  24x80  9600  cs8  sb1  none
                clocal
                -cremote -rtscts
input modes:    ignbrk  -brkint  ignpar  istrip  icrnl
                -iuclc ixon -ixany ixoff
output modes:   -olcuc onlcr -tabs
local modes:    isig icanon echo echoe echok -echonl
                echoctl echoke
control chars:  intr (^C) erase (^H) kill (^U) eof
                (^D) start (^Q) stop (^S)
```

NAME

system - display or reset the system configuration parameters

SYNOPSIS

system [**-r**]

DESCRIPTION

system displays the following system information when invoked without the **-r** option:

- the copyright notice
- the code release level
- The One Port's Ethernet address
- The One Port's host name and IP address
- the default domain name
- the domain name server
- the system user name
- the system initialization command string

OPTIONS

-r Reset the system initialization key. The system parameters are reset to their factory default settings after The One Port is turned off and on. Note: any configuration changes made by commands executed after you run "**system -r**" are lost when The One Port is turned off and on.

SEE ALSO

hosts, **ifconfig**, **initcmd**, **usrname**
"Non-Volatile Memory" section of Chapter 3

NAME

telnet - remote host user interface using the TELNET protocol

SYNOPSIS

telnet [*host* [*port*]]

DESCRIPTION

telnet communicates with another host using the TELNET protocol. If **telnet** is invoked without arguments, **telnet** enters command mode and displays the “`telnet> `” prompt; otherwise, it attempts to connect directly to *host*. *host* is specified in dotted decimal notation or by name. The default *port* number is 23.

NOTES

telnet does not perform the normal initial option negotiation when *port* is present on the command line unless the port number is preceded with a hyphen (-) character.

The *port* number can also be preceded with a “d” or an “r” character to run **telnet** in raw mode (see the “Serial-Line Extender” section of Chapter 3 for details).

SEE ALSO

rlogin
“Serial-Line Extender” section of Chapter 3
“TELNET” section of Chapter 3

NAME

usrname - display or set the system user name

SYNOPSIS

usrname [*username*]

DESCRIPTION

usrname displays the system user name when invoked without the *username* argument; otherwise, the user name is set to *username*.

rlogin uses the system user name unless the name is specified on the command line (see the **-l** option of **rlogin**).

The default system user name is “guest.” *usrname* can be a maximum of 32 characters in length.

SEE ALSO

rlogin

NAME

wait - wait for a time period or until the IP address is acquired

SYNOPSIS

wait (*cnt* | **IP**)

DESCRIPTION

wait waits for *cnt* seconds or, with the **IP** parameter, until The One Port acquires an IP address (through DHCP or RARP). This command is normally used in conjunction with the **initcmd**.

SEE ALSO

echo, **initcmd**

Appendix B—RCON Syntax

NAME

rcon - establish an outgoing connection with The One Port

SYNOPSIS

rcon [**-b** *baud*] [**-h**] [**-p** *port*] [**-r**] [**-s**] [**-v**] *oneport node*

DESCRIPTION

rcon establishes an outgoing (remote) connection with The One Port and transfers data between The One Port and an host application using the system's pseudo-terminal (pty) driver. The slave side of the pty is linked to the user defined node, *node*. Data written to *node* is transferred to The One Port and data received from The One Port is written by **rcon** to *node* making it available (readable) by the application.

The One Port's terminal modes can be set as described in the OPTIONS section below. *oneport* is given in dotted decimal notation or by name. The slave pty special device file is linked to *node*; therefore, *node* should not already exist (or be used) in the system.

OPTIONS

-b *baud* Set the baud rate on The One Port. Hardware flow control should be enabled for speeds above 9600 baud. The default baud rate is 9600.

-h Enable hardware flow control.

-p *port* The incoming TCP port number on The One Port. This must match the port number already configured on The One Port. The default port number is 3001.

-r Enable raw mode. No character processing is performed.

-s Enable software flow control.

-v Display copyright and version information.

EXAMPLE

The command

```
# ./rcon -h -r oneport /dev/oneport
```

starts the **rcon** daemon (program) running in raw mode with hardware flow control enabled. The `oneport` parameter is the name of The One Port as found in the `/etc/hosts` file (from which the **rcon** program derives The One Port's IP address). `/dev/oneport` is the name of the special device file **rcon** creates which other programs use to access The One Port's RS-232 port.

rcon is usually invoked from a boot or initialization script.

MAKE INSTRUCTIONS

Use the default **make** rules, type "`cc -o rcon rcon.c`" or on some systems, type "`cc -o rcon rcon.c -lsocket`". You may also need to enable a preprocessor directive specific to your machine or operating system (such as -DSCO). See `rcon.c` for the list of directives.

Appendix C—SNMP MIB-II Objects

The One Port's Simple Network Management Protocol (SNMP) agent accepts and processes all MIB-II objects defined in RFC 1213 except for the Exterior Gateway Protocol (EGP) objects. The access level of each EGP object is "not-accessible."

An NMS can monitor the objects listed in the table below. Objects with "read-write" access are marked with an asterisk (*); objects that can be modified with the **snet** command are indicated with a dagger (†).

MIB-II Object	Description
System Group	
`sysDescr`	Device description—"The One Port (X.Y)"; where X.Y is the firmware revision level
`sysObjectID`	1.3.6.1.4.1.895.1.1 (iso.org.dod.internet. private.enterprises.versalynx.tservers. oneport)
`sysUpTime`	Time since restart in 1/100's of a second
`sysContact*`	Name of the contact person; size is [0..31] (see **snmpcon**)
`sysName*`	Device name (the same as the host name of The One Port—see **hosts**); size is [0..31]
`sysLocation*`	Device location; size is [0..31] (see **snmploc**)
`sysServices`	Device services—72 (transport and application layer services provided)

MIB-II Object	Description
Interfaces Group	
`ifNumber`	Number of network interfaces
`ifTable`	*The Interfaces Table*
`ifIndex`	Interface number
`ifDescr`	Interface description
`ifType`	Interface type: 6 = Ethernet CSMA/CD 24 = software loopback
`ifMtu`	Maximum transmission unit size
`ifSpeed`	Transmission speed in bits-per-second
`ifPhyAddress`	Media physical address
`ifAdminStatus*`	Desired interface state: 1 = up 2 = down 3 = testing Note: The object can be written but the new value is ignored. The One Port's state cannot be changed.
`ifOperStatus`	Operational state: 1 = up 2 = down 3 = testing
`ifLastChange`	Time since operational state was changed in 1/100's of a second
`ifInOctets`	Number of bytes received
`ifInUcastPkts`	Number of unicast packets accepted
`ifInNUcastPkts`	Number of non-unicast packets accepted
`ifInDiscards`	Number of packets discarded
`ifInErrors`	Number of malformed packets received
`ifInUnknownProtos`	Number of packets with unknown protocols
`ifOutOctets`	Number of bytes sent
`ifOutUcastPkts`	Number of unicast packets sent
`ifOutNUcastPkts`	Number of non-unicast packets sent

MIB-II Object	Description
Interfaces Group (cont.)	*The Interfaces Table* (cont.)
ifOutDiscards	Number of outbound packets discarded
ifOutErrors	Number of outbound packets discarded due to errors
ifOutQlen	Length of output packet queue
ifSpecific	Interface specific MIB—(0.0)
Address Translation Group	
atTable	*The Address Translation Table*
atIfIndex*	Interface number
atPhysAddress*	Media physical address Note: The table entry is deleted when this object has a zero length in a set-request PDU.
atNetAddress*	IP network address
IP Group	
ipForwarding*	IP forwarding: 1 = gateway 2 = host Note: The object can be written but the new value is ignored. The One Port is always a host.
ipDefaultTTL*†	Default time-to-live in gateway hops
ipInReceives	Number of input datagrams
ipInHdrErrors	Number of datagrams discarded due to header errors
ipInAddrErrors	Number of datagrams discarded due to address errors
ipForwDatagrams	Number of datagrams forwarded
ipInUnknownProtos	Number of datagrams discarded due to unknown protocols

MIB-II Object	Description
IP Group (cont.)	
ipInDiscards	Number of datagrams discarded due to other reasons
ipInDelivers	Number of datagrams delivered
ipOutRequests	Number of datagrams requested
ipOutDiscards	Number of datagrams discarded for other reasons
ipOutNoRoutes	Number of datagrams discarded for no route
ipReasmTimeout†	IP reassembly timeout in seconds
ipReasmReqds	Number of IP fragments received
ipReasmOKs	Number of datagrams reassembled
ipReasmFails	Number of reassembly failures
ipFragOKs	Number of datagrams fragmented
ipFragFails	Number of datagrams for which fragmentation failed
ipFragCreates	Number of fragments created
ipAddrTable	*The IP Address Table*
ipAdEntAddr	IP network address
ipAdEntIfIndex	Interface number
ipAdEntNetMask	Subnet-mask
ipAdEntBcastAddr	Not used—always zero
ipAdEntReasmMaxSiz e	Largest IP datagram that can be reassembled
ipRouteTable	*The IP Routing Table*
ipRouteDest*	Destination IP address
ipRouteIfIndex*	Interface number
ipRouteMetric1*	Primary routing metric. Note: The maximum value is 16 (which represents an infinite metric). This object must be zero if ipRouteType is direct (3).

MIB-II Object	Description
IP Group (cont.)	*IP Routing Table* (cont.)
`ipRouteMetric2*`	Alternate routing metric Note: The object can be written but the value is never used.
`ipRouteMetric3*`	Alternate routing metric Note: The object can be written but the value is never used.
`ipRouteMetric4*`	Alternate routing metric Note: The object can be written but the value is never used.
`ipRouteNextHop*`	Next hop IP address
`ipRouteType*`	Route type: 1 = other 2 = invalid (delete) 3 = direct 4 = remote
`ipRouteProto`	Routing protocol acquired from: 1 = other 2 = local (manual) 3 = netmgmt 4 = ICMP
`ipRouteAge*`	Route entry age since last update in seconds 999 = no timeout
`ipRouteMask*`	Route subnet-mask
`ipRouteMetric5*`	Alternate routing metric Note: The object can be written but the value is never used.
`ipRouteInfo`	Route specific MIB—(0.0)

MIB-II Object	Description
IP Group (cont.)	
`ipNetToMediaTable`	*The IP Address Translation Table*
`ipNetToMediaIfIndex`*	Interface number
`ipNetToMediaPhysAddress`*	Media physical address
`ipNetToMediaNetAddress`*	IP network address
`ipNetToMediaType`*	Type of mapping: 1 = other 2 = invalid (delete) 3 = dynamic 4 = static
`ipRoutingDiscards`	Number of valid route entries discarded
ICMP Group	
`icmpInMsgs`	Number of messages received
`icmpInErrors`	Number of messages received with errors
`icmpInDestUnreachs`	Number of destination unreachable messages received
`icmpInTimeExcds`	Number of time exceededs received
`icmpInParmProbs`	Number of parameter problems received
`icmpInSrcQuenchs`	Number of source quenches received
`icmpInRedirects`	Number of redirects received
`icmpInEchos`	Number of echo requests received
`icmpInEchoReps`	Number of echo replies received
`icmpInTimestamps`	Number of timestamp requests received
`icmpInTimestampReps`	Number of timestamp replies received
`icmpInAddrMasks`	Number of address mask requests received
`icmpInAddrMaskReps`	Number of address mask replies received
`icmpOutMsgs`	Number of messages requested to be sent
`icmpOutErrors`	Number of messages not sent due to errors
`icmpOutDestUnreachs`	Number of destination unreachables sent
`icmpOutTimeExcds`	Number of time exceededs sent
`icmpOutParmProbs`	Number of parameter problems sent

MIB-II Object	Description
ICMP Group (cont.)	
`icmpOutSrcQuenchs`	Number of source quenches sent
`icmpOutRedirects`	Number of redirects sent
`icmpOutEchos`	Number of echo requests sent
`icmpOutEchoReps`	Number of echo replies sent
`icmpOutTimestamps`	Number of timestamp requests sent
`icmpOutTimestampReps`	Number of timestamp replies sent
`icmpOutAddrMasks`	Number of address mask requests sent
`icmpOutAddrMaskReps`	Number of address mask replies sent
TCP Group	
`tcpRtoAlgorithm`	TCP retransmission algorithm: 4 = Van Jacobsen’s algorithm
`tcpRtoMin`	Minimum retransmission timeout in milliseconds
`tcpRtoMax`	Maximum retransmission timeout in milliseconds
`tcpMaxConn`	Maximum number of connections
`tcpActiveOpens`	Number of active open connections
`tcpPassiveOpens`	Number of passive open connections
`tcpAttemptFails`	Number of failed connection attempts
`tcpEstabResets`	Number of reset connections
`tcpCurrEstab`	Number of currently open connections
`tcpInSegs`	Number of segments received
`tcpOutSegs`	Number of segments sent
`tcpRetransSegs`	Number of segments retransmitted

MIB-II Object	Description
TCP Group (cont.)	
`tcpConnTable`	*The TCP Connection Table*
`tcpConnState*`	TCP connection state: 1 = CLOSED 2 = LISTEN 3 = SYN_SENT 4 = SYN_RECEIVED 5 = ESTABLISHED 6 = FIN_WAIT_1 7 = FIN_WAIT_2 8 = CLOSE_WAIT 9 = LAST_ACK 10 = CLOSING 11 = TIME_WAIT 12 = DELETE_TCB Note: The only values that can be written are “CLOSED” and “DELETE_TCB.” Both settings immediately terminate an open connection.
`tcpConnLocalAddress`	Local IP address
`tcpConnLocalPort`	Local TCP port number
`tcpConnRemAddress`	Remote IP address
`tcpConnRemPort`	Remote TCP port number
`tcpInErrs`	Number of segments received in error
`tcpOutRsts`	Number of segments sent with the RST (reset) flag set
UDP Group	
`udpInDatagrams`	Number of datagrams received
`udpNoPorts`	Number of datagrams discarded due to unknown ports
`udpInErrors`	Number of datagrams discarded due to other errors
`udpOutDatagrams`	Number of datagrams sent
`udpTable`	*The UDP Listener Table*
`udpLocalAddress`	Local IP address
`udpLocalPort`	Local UDP port number

MIB-II Object	Description
SNMP Group	
`snmpInPkts`	Number of packets received
`snmpOutPkts`	Number of packets sent
`snmpInBadVersions`	Number of packets with the version number not equal to zero
`snmpInBadCommunityNames`	Number of packets with an unknown community name
`snmpInBadCommunityUses`	Number of packets with an illegal operation for the provided community
`snmpInASNParseErrs`	Number of packets with encoding or syntax parsing errors
`snmpInBadTypes`	Number of PDUs with an unknown type Note: This object is no longer used.
`snmpInTooBigs`	Number of PDUs received with a `tooBig` error
`snmpInNoSuchNames`	Number of PDUs received with a `noSuchName` error
`snmpInBadValues`	Number of PDUs received with a `badValue` error
`snmpInReadOnlys`	Number of PDUs received with a `readOnly` error
`snmpInGenErrs`	Number of PDUs received with a `genErr` error
`snmpInTotalReqVars`	Number of MIB objects successfully retrieved from get- and get-next-request PDUs
`snmpInTotalSetVars`	Number of MIB objects modified
`snmpInGetRequests`	Number of accepted and processed get-request PDUs
`snmpInGetNexts`	Number of accepted and processed get-next-request PDUs

MIB-II Object	Description
SNMP Group (cont.)	
snmpInSetRequests	Number of accepted and processed set-request PDUs
snmpInGetResponses	Number of accepted and processed get-response PDUs
snmpInTraps	Number of accepted and processed trap PDUs
snmpOutTooBigs	Number of PDUs sent with the error status set to tooBig
snmpOutNoSuchNames	Number of PDUs sent with the error status set to noSuchName
snmpOutBadValues	Number of PDUs sent with the error status set to badValue
snmpOutReadOnlys	Number of PDUs sent with the error status set to readOnly Note: This object is no longer used.
snmpOutGenErrs	Number of PDUs sent with the error status set to genErr
snmpOutGetRequests	Number of get-request PDUs sent
snmpOutGetNexts	Number of get-next-request PDUs sent
snmpOutSetRequests	Number of set-request PDUs sent
snmpOutGetResponses	Number of get-response PDUs sent
snmpOutTraps	Number of trap PDUs sent
snmpEnableAuthenTraps*	trap PDU generation state: 1 = enabled 2 = disabled Note: Traps, when enabled, are generated for coldStart (any One Port reset condition), linkUp, linkDown, and authenticationFailure.

Appendix D—RFCs

The Request for Comments (RFCs) applicable to The One Port are listed in the table below:

RFC	Description
768	User Datagram Protocol (UDP)
791	Internet Protocol (IP)
792	Internet Control Message Protocol (ICMP)
793	Transmission Control Protocol (TCP)
826	Address Resolution Protocol (ARP)
854	TELNET Protocol
855	TELNET Option Specification
856	TELNET Binary Transmission
857	TELNET Echo Option
858	TELNET Suppress Go Ahead Option
859	TELNET Status Option
860	TELNET Timing Mark Option
861	TELNET Extended Options
903	Reverse Address Resolution Protocol (RARP)
1034	Domain Names - Concepts and Facilities
1035	Domain Names - Implementation and Specification
1055	Serial Line IP (SLIP)
1058	Routing Information Protocol (RIP)
1079	TELNET Terminal Speed Option
1091	TELNET Terminal Type Option
1155	Structure of Management Information (SMI)
1157	Simple Network Management Protocol (SNMP)
	Continued on next page

RFC	Description
1179	Line Printer Daemon Protocol (LPD)
1213	Management Information Base (MIB-II)
1282	BSD rlogin
1332	PPP Internet Protocol Control Protocol (IPCP)
1372	TELNET Remote Flow Control Option
1661	The Point-to-Point Protocol (PPP)
1662	PPP in HDLC-like Framing
2131	Dynamic Host Configuration Protocol (DHCP)
2132	DHCP Options and BOOTP Vendor Extensions

Glossary

The definitions provided below are applicable to The One Port and to TCP/IP networks. Other, and undoubtedly broader, definitions exist.

address mask

A 32-bit mask used to select a section of an Internet address for subnet addressing. The mask delineates the host and network portions of an IP address.

ARP

Address Resolution Protocol. The Internet protocol used to bind an IP address to a hardware (Ethernet) address.

ASCII

American Standard Code for Information Interchange. The ASCII character set contains 96 printable characters and 32 non-printable control characters. Each character is represented by a seven-bit quantity. An eighth bit can be used for parity checking.

bridge

A device that connects two or more networks at the physical (hardware) layer and transfers packets between them. Bridges transfer packets while repeaters forward electrical signals.

datagram

The basic IP protocol unit containing the source address, the destination address and the data to be transferred.

DCE

Data Communication Equipment. Devices, such as modems, which transmit data on RS-232 signal RXD and receive data on signal TXD.

DHCP

Dynamic Host Configuration Protocol. An Internet protocol devices use to acquire configuration information (including their IP address) from a server.

dotted decimal notation

The representation of a 32-bit IP address in four eight-bit decimal numbers separated by periods (e.g., 192.10.160.3).

DNS

Domain Name Service. The set of distributed servers and resolvers (along with the protocol and the host databases) used to translate text names into IP addresses.

DTE

Data Terminal Equipment. Devices, such as terminals, which transmit data on RS-232 signal TXD and receive data on signal RXD. The RS-232 signal names are defined relative to a DTE device (i.e., TXD equals transmitted data).

EGP

Exterior Gateway Protocol. The protocol used to advertise the IP addresses of one system to a gateway of another.

Ethernet

The network hardware that utilizes Carrier Sense Multiple Access with Collision Detection (CSMA/CD or 802.3) technology to exchange data between systems.

Ethernet address

The unique 48-bit number assigned to a network device.

gateway

An IP router or network device that connects two or more networks and transfers packets between them.

host

A computer connected to a network that provides network services.

hub

A device that connects the nodes of a multipoint (star) network.

ICMP

Internet Control Message Protocol. A protocol that handles error and control messages. ICMP is actually part of IP itself.

internet

A collection of gateway interconnected networks that function as one large network.

Internet

The Internet is the largest internet in the world. It uses the TCP/IP protocol suite for data communication and consists of many universities, government and commercial sites.

IP

Internet Protocol. The protocol of the Internet protocol suite that provides a connectionless, best-effort packet delivery service.

LPD

Line Printer Daemon Protocol. The protocol used to control printing.

MIB

Management Information Base. A set of database objects that are accessible by SNMP.

nickname

An alternate host name used in place of the official host name.

PDU

Protocol Data Unit. The unit of data transferred between protocols usually consisting of protocol-specific infomation and user data.

ping

Packet InterNet Groper. A program used to test the reachability of a network device. Ping uses ICMP echo request and reply messages.

port

An abstraction used to distinguish between multiple destinations (programs) on a host.

PPP

Point-to-Point Protocol. A protocol used to encapsulate network layer protocol datagrams over synchronous or asynchronous links. Unlike SLIP, PPP provides mechanisms for peers to negotiate various link options.

RARP

Reverse Address Resolution Protocol. An Internet protocol devices use to acquire their IP address (by broadcasting their Ethernet address) from a server.

RFC

Request For Comments. A name given to the numbered collection of papers covering proposed and accepted IP standards and other related material.

RIP

Routing Information Protocol. The protocol used to exchange routing information.

router

A device responsible for selecting the network path over which datagrams are sent.

SLIP

Serial Line Internet Protocol. A protocol used encapsulate IP datagrams over a serial line.

SNMP

Simple Network Management Protocol. A protocol used to monitor network devices.

TCP

Transmission Control Protocol. A protocol that provides a reliable, connected, full-duplex packet delivery service using IP for datagram delivery.

TPE

Twisted-Pair Ethernet. The twisted-pair cable used for packet transmission on an Ethernet network. TPE is defined by the IEEE 802.3 10BASE-T specification.

UDP

User Datagram Protocol. A protocol used by application programs to send datagrams to other application programs. UDP uses IP for datagram delivery and provides ports to identify the destination program.

Glossary

Index

Notes

Notes

Notes

Notes