Using Serial Line
IP Protocols

HP 9000 Networking

Using Serial Line IP Protocols

HEWLETT®
PACKARD

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

Copyright Notices. ©copyright 1983-95 Hewlett-Packard Company, all rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

©copyright 1979, 1980, 1983, 1985-93 Regents of the University of California

This software is based in part on the Fourth Berkeley Software Distribution

under license from the Regents of the University of California.

Trademark Notices UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

MS-DOS and Microsoft are U.S. registered trademarks of Microsoft Corporation.

OSF/Motif is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries.

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: February 1991

Second Edition: July 1992 (HP-UX Release 9.0)

Thirdt Edition: January 1995 (HP-UX Release 10.0)

Related Information

The Serial Line Internet Protocol (SLIP) is described in RFC1055.

The Compressed header Serial Line Internet Protocol (CSLIP) is described in RFC1144.

HP's Point-to-Point Link (PPL) provides a command program, *ppl*, which is a SLIP daemon. *ppl* is described in detail in the man page *ppl(1)*. Its configuration files are described in *ppl.remotes(4)*, *ppl.users(4)*, *ppl.ipool(4)*, *ppl.ptmp(4)*, and *pplstat(1)*.

PPL configuration requires the user to specify a serial port. This is described in the man page *termio(7)*.

The use of UUCP *Devices, Dialers,* and *Systems* files in the PPL configuration is optional. Refer to the man page *uucp(1)* for details when necessary.

Refer to the following manuals for additional information:

System Administration Tasks

Installing and Administering LAN/9000 Software

DTC Device File Access Utilities and Telnet Port Identification

Preface

This document describes how to use serial line IP protocols (SLIP and CSLIP) with HP 9000 computers. The protocols can be used with the Point-to-Point Link facility (PPL) of HP's LAN/9000 product. SLIP is also supported over modem-to-DTC-LAN connections in binary mode up to 9600 baud for HP 9000 Series 700 and Series 800 hosts.

This document assumes the reader is familiar with HP 9000 system administration and basics of serial communication.

Contents

Contents

Contents

Contents

1

Serial Line IP Basics

This chapter describes the basics of serial line IP and HP's Point-to-Point
Link networking facility.

Why Use Serial Line IP?

Terrain, distance, and property rights often limit LAN cabling. Where coax is restricted, network connections can be made by serial lines.

Serial lines are relatively inexpensive and easy to install. However, you need special software to run TCP/IP applications over a serial link. Specifically, you need software that implements protocols such as Serial Line Internet Protocol (SLIP).

HP's Point-to-Point Link

HP 9000 computers implement SLIP and other serial line IP protocols with a facility called Point-to-Point Link (PPL). The PPL facility is supplied with HP's LAN/9000 product. PPL supports both the SLIP and CSLIP protocols.

PPL includes a command program (named *ppl*), a supporting utility, and 3 sample configuration files.

You can use HP's PPL to:

- Dial-in to HP 9000 serial IP lines.

- Dial-out from HP 9000 serial IP lines.

- Direct connect to HP 9000 serial IP lines.

Dialing In to an HP 9000

Users at remote supported terminals or PCs can establish dial-in IP connections with or without login.

For a login connection, the user establishes a modem link to an HP 9000 and logs in as usual. After login, the user invokes *ppl* at the HP-UX shell prompt. *ppl* starts the desired serial IP protocol, allowing the user full network access to the HP 9000.

For a connection without login, the user simply dials in to a preset HP 9000 SLIP line. The specified serial protocol is already running on that line.

Dialing Out from an HP 9000

HP 9000 users can establish dial-out IP connections with or without login. This may be to any remote host that runs a supported serial IP protocol.

For a login connection, the user simply invokes *ppl* at the HP-UX shell prompt. *ppl* establishes a modem link to the specified host and logs in. *ppl* runs a command on the remote machine to initiate the specified serial IP protocol and establish the connection.

For a connection without login, the user's action is the same. After the user invokes *ppl*, a preset SLIP line is assigned to that user. The user has full network access to the remote host.

Direct Connections

The system administrator may set up a direct (hardwired) IP connection between an HP 9000 and a remote host. In effect, this is simply a long distance extension of the LAN.

Which Protocol to Use

- Using **SLIP**

  SLIP, Serial Line Internet Protocol, is currently a de facto standard, commonly used for point-to-point serial connections running TCP/IP. If you must establish a link with a peer host that is unable to use the protocol CSLIP, use SLIP.

- Using **CSLIP** or **CSLIPA**

  CSLIP, Compressed header Serial Line Internet Protocol, refers to a SLIP implementation that incorporates Van Jacobson's TCP packet header compression algorithm. CSLIP reduces the number of protocol overhead bytes required for typical packets in a TCP data connection to improve interactive responsiveness.

  CSLIP is most useful for interactive connections (for example, telnet or X-windows). No substantial difference in throughput exists, however, for large data transfer (for example, ftp or rcp) whether SLIP or CSLIP is used.

  HP's PPL has two ways to configure CSLIP: CSLIP or CSLIPA. HP recommends using CSLIP for interactive connections whenever possible.

  If you connect to a system with unknown capabilities (possibly SLIP or CSLIP), you should not use CSLIP unless you are sure the system supports this particular protocol. Otherwise, no communication is possible. Use CSLIPA to switch automatically to the correct protocol. CSLIPA will transmit compressed headers if the other system sends them. This prevents the need to choose the correct protocol or find out what capabilities the other system has.

- Using **ASLIPC** or **ASLIPS**

  ASLIP, Abbreviated Serial Line IP Protocol, is slightly faster than SLIP, because it contains less overhead in its header. ASLIP Clients and ASLIP Servers (ASLIPC and ASLIPS) differ in the way addresses are copied and synthesized across the serial line as shown in the following diagram.

```
CLIENT SIDE                                           SERVER SIDE
outgoing.ip_src - discarded     remote addr - incoming.ip_src
outgoing.ip_dst     ------------------------> incoming.ip_dst

incoming.ip_src  <------------------------ outgoing.ip_src
incoming.ip_dst <---- local addr  discarded -outgoing.ip_dst
```

  CLIENT SIDE: the client will discard the IP source address prior to sending the packet. The server will add the remote address back into the IP header when it receives the packet.

SERVER SIDE: The server discards the outbound packet destination IP address before transmitting the packet. The client side will add the local address back into the IP header upon receiving the packet.

This only works when the client **IS NOT** a gateway, since the server will not send abbreviated packets if packet destination does not match the server's configured remote address and the client will not send abbreviated packets if source address does not match configured local address. If you use ASLIP for gateways, all gateway packets are sent with complete header information; no abbreviation will take effect. So, using ASLIP with gateways is not recommended.

Since CSLIP is available, HP recommends using CSLIP instead of ASLIP whenever possible. CSLIP is an industry standard.

- Using **PPP**

HP's PPL does not support the Point-to-Point Protocol (PPP).

2

Setting Up Serial IP Lines

Before you get started, you should already have installed any necessary
hardware (for example, cables or modems). Before you can use serial IP
lines on HP 9000 computers, determine whether you need to install the
SLIP/PPL software, then configure PPL with the information in this chapter.

Installing SLIP/PPL Software

You do not need to install the SLIP/PPL software if:

- the LAN/9000 networking product was pre-installed on your system (instant ignition),

- you have already used the HP-UX *swinstall* program to install the Core Networking Bundle. The SLIP-RUN fileset is part of this software bundle.

To determine whether SLIP/PPL software has been installed on your system and is included in your HP-UX kernel,

1 Type the command:

```
ls /var/adm/sw/products/Networking
```

If SLIP-RUN appears in the list of filesets, then the SLIP/PPL software is already installed on your system.

2 Type the command:

```
grep ni /stand/system
```

If **ni** appears in the list of dfile keywords, then SLIP/PPL is already included in your HP-UX kernel.

If the SLIP/PPL software is not already installed or is not included in your HP-UX kernel, install it by running the HP-UX *swinstall* program with the command:

```
/usr/sbin/swinstall
```

Highlight the SLIP-RUN fileset from the Software Selection Window and choose "Install" from the menu. When you click OK, *swinstall* will install the SLIP-RUN fileset and regenerate the kernel.

NOTE: The SLIP-RUN fileset is not included with the Series 700 HP-UX Desktop product.

Configuring PPL - Quick Reference

DIAL-IN	DIAL-OUT with UUCP name	DIAL-OUT without UUCP name	DIRECT CONNECT
Step 1: Create device file with dial-in minor number for serial port. Refer to *termio(7)* man page.	**Step 1:** Create device file with dial-out minor number for serial port. Refer to *termio(7)* man page.	**Step 1:** Create device file with dial-out minor number for serial port. Refer to *termio(7)* man page.	**Step 1:** Create device file with direct connect minor number. Refer to *termio(7)* man page.
Step 2: Create a device file for network interface with a major number of 56 and a minor number of 0.	**Step 2:** Create a device file for network interface with a major number of 56 and a minor number of 0.	**Step 2:** Create a device file for network interface with a major number of 56 and a minor number of 0.	**Step 2:** Create a device file for network interface with a major number of 56 and a minor number of 0.
Step 3: Configure your modem.	**Step 3:** Configure your modem.	**Step 3:** Configure your modem.	**Step 3:** Configure your modem.
Step 4: Modify the file /etc/ppl/ppl.remotes. No need to modify UUCP files.	**Step 4:** Modify the files /etc/uucp/Devices, /etc/uucp/Dialers, and /etc/uucp/Systems to enter line speed, phone #, system name, and device file name.	**Step 4:** Modify the file /etc/uucp/Devices and add entries to the file /etc/uucp/Dialers if you are not using a Hayes-compatible modem..	**Step 4:** Modify or add an entry in the file /etc/uucp/Devices to reflect direct connect.
Step 5 (optional): Modify the file /etc/ppl/ppl.ipool.	**Step 5:** Modify the file /etc/ppl/ppl.remotes to enter UUCP name and optional log in info..	**Step 5:** Modify the file /etc/ppl/ppl.remotes to enter line speed, addresses, phone #, system name, device file name, and optional log in info.	**Step 5:** Modify the file /etc/ppl/ppl.remotes to enter line speed, addresses, phone #, system name, and device file name.
Step 6 (optional): Modify the file /etc/ppl/ppl.users.	**Step 6 (optional):** Modify the file /etc/ppl/ppl.users.	**Step 6 (optional):** Modify the file /etc/ppl/ppl.users.	**Step 6 (optional):** Modify the file /etc/ppl/ppl.users.

Creating Device Files for PPL

You need to create device files in the **/dev** directory for serial ports and the network interface used by PPL. This section provides brief instructions for creating device files. For more information, refer to *System Administration Tasks* or the *termio(7)* man page.

Creating Device Files for Serial Ports

Each serial port used for PPL may require one to four device files. Depending on the hardware attached, you may need files for dial-in, dial-out, or direct connection.

Use SAM, the System Administration Manager, to create device files. You can also use the *mknod(1M)* command to create device files, for example:

```
mknod /dev/ttyd10 c 1 0x000101 #dial-in
```

Creating the device files with appropriate minor numbers is important. The *termio(7)* man page describes how to derive the minor number of the device file for dial-in, dial-out, and direct connections. Major/minor numbers may be different from system to system.

The following is an example of an entry in **/dev** for the device **tty0b1**, a dial-in modem with hardware flow control enabled.

```
major number = 1
minor number = 0x000012
```

where the following template is resolved:

```
0xIIPPHM
```

II   Two hexadecimal digits (8 bits) to indicate the instance of the serial interface. The instance of the serial interface is determined by running the program *ioscan -f* and looking in the "I" column. On a Series 700, port A will generally = 0 and port B = 1.

PP   Two hexadecimal digits (8 bits) to indicate the port number of this device on the serial interface. On a Series 700 serial port, this will always be 0.

H	One hexadecimal digit (4 bits) which controls diagnostic access and hardware flow control (HP J2094A only). Bit 0 enables RTS/CTS hardware flow control.
M	One hexadecimal digit (4 bits) to determine port access type. Values for each bit are:

bit 3 = TI/ALP

bit 2 = 0 means simple protocol (U.S.A.)
bit 2 = 1 means CCITT protocol (Europe)

bits 0 and 1 = 00 direct, 01 dial-out modem, 10 dial-in modem, 11 unused.

See the *termio(7)* man page for more details about configuring a serial device.

Creating a Device File for the Network Interface

Check to see if you have a **/dev/ni** device file with a major number of 56 and minor number of 0. If you do not, create this device file using the following command:

```
mknod /dev/ni c 56 0 #network interface
```

The *ppl* program uses this device to pass packets between the serial line and IP layer.

Configuring Your Modem

Make sure you are using a modem that your MUX card supports. MultiTech modems and Hayes Smartmodems are generally supported. The following are general recommendations for configuring your modem. Please refer to your owner's manual for a detailed explanation of the modem settings.

General Recommendations for Modem Settings

- The modem setting, Xon/Xoff, should be turned off, because SLIP and CSLIP do not allow Xon/Xoff. If this setting is turned on, the transferred data may be corrupted.

- In general, the supported baud rate is limited by the system hardware. For example, some serial interface cards support a maximum speed of 19200 baud, whereas Series 700 built-in serial ports support a maximum speed of 57600 baud. PPL supports up to 57600 baud in the software if the hardware can support this speed.

- If both your serial interface card and your modem support hardware flow control (RTS/CTS), HP recommends using it. The minor number of the device file has a bit to indicate hardware flow control. All systems with HP-PBs (HP Precision Bus) support hardware flow control. Refer to the man page *termio(7)* for specific models and references.

- SLIP over DTCs requires Hayes-compatible modems that are supported with HP 9000 Series 700/800 systems and DTCs, for example, the US MultiTech modem. The maximum dial-in speed supported for SLIP over DTC connections is 9600 baud. You should also configure the modem with flow control turned off.

- If you are using a MultiTech V.32B modem, the following register settings are helpful:

```
>atL5
B1 E1 M1 Q0 R0 V1 X0 &E1 &E4 &E6 &E8 &E10 &E13 &E15
$MB9600 $SB9600 $BA0 &W1
```

The following list explains each of these register settings.

B1	selects Wait-for-Dial-Tone Dialing
E1	means do echo Command Mode characters
M1	means Monitor speaker On until Carrier detected
Q0	means Result Codes sent
R0	means modem will not reverse modes

V1	means Result Codes sent as words (verbose mode)
X0	selects Basic Result Codes
&E1	means V.42 Auto-Reliable Mode
&E4	means CTS modem-initiated flow control
&E6	means Xon/Xoff not passed through
&E8	means Enq/Ack pacing off
&E10	means Normal Mode flow control off
&E13	means Pacing on
&E15	means data compression enabled
$MB9600	selects 9600 bps on-line
$SB9600	selects 9600 bps at serial port
$BA0	means Baud Adjust is off, speed conversion is on
&W1	causes modem to store its current parameters into non-volatile RAM, and modem will load these for future sessions instead of reading factory ROM and DIP switch defaults

If you have some other type of modem that is not described here, you should configure it with similar functions (for example, RTS/CTS flow control set and Xon/Xoff not set), using your modem's command set and register semantics.

HP recommends verifying that your modem connection is working correctly before configuring a SLIP connection. This will detect any hardware, software, or modem configuration problems that are not related to the protocol.

To verify that the modem connection is working, use the following command to connect to a remote system.

```
cu -l <device file> -s<speed>
```

You should be able to connect to a remote system using the *cu* command before configuring PPL. The remote system should display a login prompt and users can login to the remote system. This will verify that the modem line is functional.

Using UUCP Configuration Files

PPL provides the flexibility to use UUCP files for dial-out connections. For dial-in and direct connections, you do not need to edit the UUCP files. If you use the dial-out feature, you need to add entries to these UUCP configuration files: `/etc/uucp/Devices` and `/etc/uucp/Dialers`. If you use the dial-out feature and you specify a UUCP system name in the *ppl.remotes* configuration file, you also need to add entries in the `/etc/uucp/Systems` UUCP configuration file.

NOTE:

PPL uses the UUCP configuration files only, not UUCP executables. Only the *Devices*, *Dialers,* and *Systems* files need to be installed. The format of the files must match HoneyDanBer (HDB) UUCP, which is supplied with all supported HP-UX releases.

The UUCP *Devices* file must contain an entry for each serial port available for dial-out. For example, if `cul0b2` is the serial line and 9600 is the speed, then the *Devices* file looks like this:

```
ACU cul0b2 - 9600 hayes
```

hayes refers to an entry in the UUCP *Dialers* file that specifies the command sequences that are sent to the modem to start up a connection. The UUCP *Dialers* file should contain dialing configurations for each type of modem used in the *Devices* file.

The *Systems* file may contain optional entries for each remote host that may be called. The UUCP *Systems* file is scanned for entries matching the UUCP system name in the *ppl.remotes* file. If you specify a UUCP system name, a corresponding entry for that system name must also be included in the *Devices* file.

If the appropriate entries are not already in the UUCP configuration files, add them to help configure PPL. See the next sections for details about each of the UUCP files.

The UUCP Devices File

In general, entries in the UUCP *Devices* file have the following format:

```
device_type  device_file  calling_unit  speedialer_token_pairs
```

Entries in your *Devices* file may look like the following:

```
# Note: cua0p2, cul0p2 are attached to a hayes modem
# tty0p3 is a direct connection to hpisox1
# Note: blank lines and lines that begin with a
# <space>, <tab>, or # are ignored
hpfcxt culp0 0 9600 direct
hpisox1 tty0p3 - 9600 direct
ACU cul10 - 2400 hayes
Direct tty0p3 - 9600 direct
Direct cul09 0 2400 direct
happy cul0p2 - 9600 hayes
```

For modem connections, examine the existing *Devices* file and, if necessary, add lines to suit the type of modem, corresponding device files, and baud rates. For example, on a Series 800 system using a Hayes Smartmodem set to 9600 baud and connected to port 2 of MUX0, you would add the following line to *Devices*:

```
ACU cul0p2 - 9600 hayes
```

The following is a description of the fields in the *Devices* file entry.

Name of Field	Description
device_type	Describes the type of communication link. You can use one of the following options here:

- Use Direct for direct connections or modem connections on a leased line to a remote system for cu connections only.

- Use ACU for modem connections. ACU stands for Automatic Calling Unit.

- For direct connections to specific remote machines and when using the *Systems* file by specifying a UUCP name in *ppl.remotes*, use the remote machine's system name (such as **hpisox1**). This *Devices* entry is linked with a corresponding entry for the remote machine in the *Systems* file by matching the system name in this field with the system name in the device field of the *Systems* file.

- For a network connection, you can use the network switches

micom or develcon, which have corresponding entries in the *Dialers* file. If you want to use other network switches, make sure they have a corresponding entry in *Dialers*.

device_file   Device file name for the data communications line that is used. Contains the name of the device file in the `/dev` directory that is used by the associated communications line. For example, use `tty04` for `/dev/tty04`.

calling_unit   Device file name for the I/O interface associated with a separate Automatic Calling Unit (ACU). If the device_type field is ACU and the dialer is an 801 model (dialer is a separate device from the modem), this field contains the name of the device file associated with the 801 dialer. If you are using a non-801 dialer (such as a Hayes Smartmodem with a built-in dialer), then enter a zero (0) or a dash (-) in this field as a placeholder.

speed   Baud rate of the device, perhaps with dialer specifier. This field must match the speed field of the corresponding entry in the file *Systems* file.

dialer_token_pairs This field points to the first field of a corresponding entry in the *Dialers* file. This tells what dialing instructions (if any) are needed.

The UUCP Dialers File

The following information describes the format of individual entries. The various fields in each entry provide instructions to the UUCP subsystem for setting up an automatically dialed modem connection and initializing communication between systems. The *Dialers* file is used in conjunction with the *Devices* file.

The general syntax for a line in the *Dialers* file is:

```
modem_type   translation   expect-send [expect-send]
```

The following is a description of the fields in the *Dialers* file entry.

Name of Field   Description

modem_type   Names the type of dialer. This field must match the fifth field (or any odd-numbered field past the fifth field) of the

Devices file. When the type in this field is DIRECT, no additional fields are required in the *Dialers* entry.

translation Translation string, in which the first of each pair of characters is converted (translated) to the second character in the pair. This is usually used to translate the = and - into whatever the modem requires for *wait-for-dial-tone* and *pause* commands.

expect-send Specifies the expected "conversation" between local and remote systems (a backslash begins each escape sequence).

Here are some sample entries similar to what you may typically encounter in a *Dialers* file:

```
penril =W-P "" \d > s\p9\c )-W\p\r\ds\c-) y\c : \E\TP > 9\c OK
ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
vadic =K-K "" I IDLE \d\d\005\p HELLO D NUMBER? \T\r DIALING \d\d LINE
develcon "" "" \pr\ps\c,est:\007 \E\D\e \007
micom "" "" \s\c NAME? \D\r\c GO
direct
```

Here is an example of a *Dialers* entry for a Hayes compatible modem.

```
hayes =,-, "" \dATZ\d\r OK\r \dAT\r\c OK\r \dATS7=45\r\c OK\r \EATDT\T\r\c CONNECT
```

The following is a list of the escape sequences used in the *Dialers* file.

ESC Sequence	Function
""	Do not wait for any special responses (expect a null string).
\\	Send a backslash character (\).
BREAK	Cause a BREAK.
EOT	Send an End-Of-Transmission signal (two Control-Ds (CTRL-D).
\D	Use phone number or abbreviation without Dialcodes translation.
\E	Enable echo checking. System waits for each transmitted character to be received back (echoed) from the remote system before proceeding.
\K	Insert a BREAK.
\N	Send a null character.
\T	Use Dialcodes file to translate phone number or abbreviation.
\b	Send a backspace character.
\c	Suppress new-line character at the end of the send string.
\d	Delay for about 2 seconds.

ddd	Send the octal number equivalent of a character.
\e	Disable echo checking.
\n	Send a new-line character.
\p	Pause for 1/4 to 1/2 seconds.
\r	Send a carriage return.
\s	Send a space character.
\t	Send a tab character.

The UUCP Systems File

The general syntax for a line in the *Systems* file is:

```
sysname  time;retry  device,protocol  speed  phone chat_script
```

A typical entry in the *Systems* file looks like the following:

```
ace Any;15 ACU,g 9600 303-999-1212 "" \r\d\r\d\r ogin:-BREAK-ogin: ace ssword: mars
```

The following is a description of the fields in the *Systems* file entry.

Name of Field	Description
sysname	This string identifies a remote system. It is a tag name from *uname -n*, **/etc/clusterconf** (if it is an HP Clustered Environment), or from the alias set up in the Permissions file.
	A local system needs an entry for every remote system. This name must begin with a letter, be no more than seven (7) alphanumeric characters long, and be different from all the other node names listed in the *Systems* file.
time	Specifies days and times your system may call the remote system.
	• Use *Any* for any day of the week, *Wk* for any weekday, and *Su*, *Mo*, *Tu*, *We*, *Th*, *Fr*, or *Sa* to specify a day. To specify a range of time when your system can communicate with the remote system, use a 24-hour clock with the format start-stop. For example, to tell your system that it can call or be called between 6 pm and 1 am on weekdays and any time on weekends, enter: *Wk1800-0100,Sa,Su*.
	• Use *Never* with no additional fields to make your system a passive system (that is, your system does not call the remote

system, it only receives calls from the remote system).

This field is separated from the next field by a semicolon (;) with no spaces.

retry
When an attempt to connect to a remote system has failed, this specifies the number of minutes which must elapse before your system can make another attempt. On failing, a status file (lock file) is created that prevents retrying until the specified minutes pass. The retry time specifies earliest time for attempting connection, not earliest time another try will be made.

device
Specifies type of connection device.

- Use ACU for a modem connection. ACU stands for Automatic Calling Unit.

- Use system name (such as **ace**) for a direct connection.

This field is separated from the next field by a comma (,) with no spaces.

protocol
Specifies the data communication protocol that your system and the remote system use to transfer data after establishing a connection.

- Use g (default) with dial-up phone lines and less reliable connections (sends data in 64-byte packets with acknowledgments after packet).

- Use f with direct connections (minimal error checking, checksum after each file).

- Use x with X.25 (sends file followed by checksum and retransmits file on an error).

speed
Specifies baud rate for data transmission between systems.

- Direct connections typically use 9600 baud.

- Modem connections depend on the modem setting, but usually are one of these: 300, 1200, 2400, 4800, 9600, or 19200 baud.

phone
Specifies phone number of remote system. Two other

symbols can be used with the phone number:

- An equal sign (=) means to wait for a secondary dial tone.

- A dash (-) will generate a 5-second pause.

For a direct connection, enter a dash (-) in this field instead of a phone number.

chat_script   Specifies the expected "conversation" (login prompt and corresponding response) between the two systems.

The chat_script field contains the information necessary for logging in to the remote system. This information is given as a series of fields and subfields in the format:

```
[expect send]
```

where **expect** is the string expected to be read and **send** is the string to be sent when the expect string is received. These expect/send fields are separated by spaces. The expect fields only need to contain the last portion of the expected string (to cover the possibility of differences in uppercase and lowercase characters).

For example, **ogin** matches either Login or login because both end in "ogin", which is the expected string. The expect field can also be made up of subfields of the format:

```
expect1[-send-expect2]...
```

where the **send** is sent if the **expect1** is not successfully read and the **expect2** is the next expected string. These subfields are separated by hyphens (-).

For example, **ogin-@-ogin** expects login. If a login is received, the program goes on to the next field; if it does not get the login it sends an **@** (default KILL) character to clear any bad text in the remote input buffer followed by a new-line character, then expects login again.

Many experienced UUCP users always send one or more **@** (default KILL) characters as the first text to the remote system (three in succession should be more than adequate for any normal situation). This clears the input buffer on the remote system of any unwanted text, ensuring a correct interpretation of the login string by the remote system. This technique is especially useful on direct connections where meaningless noise can be interpreted as text

from the line by the receiving system when the transmitting device (whether a computer or local terminal) is shut down overnight, loses power, or otherwise places abnormal signal voltages on the line (such as when someone unplugs the RS-232-C connector). To use this technique, construct an expect/send string that resembles the following:

```
" " @ gin:-@-ogin: sys5 ssword: abcxyz
```

or

```
" " @@@ gin:-@-ogin: sys5 ssword: abcxyz
```

The @ sends the line-kill character before logging in. **abcxyz** represents the password, and should be replaced with a valid password string for the remote system.

When *Systems* was originally installed, the access protection mode may not have been set to 400. Since this file could contain proprietary information that may be detrimental to system security if it was accessed by unauthorized users, be sure to change its mode to 400 so that only the owner (uucp) can read the contents (**chmod 400 /usr/lib/uucp/Systems**). Be sure that uucp remains the owner because uucp must be able to read *Systems* when establishing connections for data transfers.

The following is a list of the escape sequences used in the *Systems* file.

ESC Sequence	Function
" " " "	Do not wait for any special responses (expect a null string).
@	Send a KILL character (clears unwanted text from remote system's input buffer).
\\	Send a backslash character (\).
BREAK	Cause a BREAK.
EOT	Send an End-Of-Transmission signal (two Control-Ds (CTRL-D).
\K	Insert a BREAK.
\N	Send a null character.
\b	Send a backspace character.
\c	Suppress new-line character at the end of the send string.
\d	Delay for about 2 seconds.
\ddd	Send the octal number equivalent of a character.
\n	Send a new-line character.
\p	Pause for 1/4 to 1/2 seconds.
\r	Send a carriage return.

\s	Send a space character.
\t	Send a tab character.

A correctly structured chat_script field looks like the following:

```
login:-\d\d\d@\c-login:XYZ  ssword: Plymouth
```

Here are a few examples of different kinds of entries in the *Systems* file.

For a modem connection, use a line like the following:

```
test1 Any;15 ACU,g 9600 303-999-1212 "" \r\d\r\d\r ogin:-BREAK-ogin: test1 ssword: one
```

For a direct connection, use a line like the following:

```
test2 Any;15 hptest2,f 9600 - "" \r\d\r\d\r ogin:-BREAK-ogin: test2 ssword: two
```

To require a remote system to call you, use a line like the following:

```
hpdsfg Never
```

Building PPL Configuration Files

PPL allows you to configure SLIP/CSLIP connections in many different forms. You can use PPL to set up a dial-in, dial-out, or direct connection on HP-UX, and you can choose the protocol, SLIP or CSLIP, for each connection.

PPL configuration involves creating and editing these files:

- *ppl.remotes*
- *ppl.users*
- *ppl.ipool*

For most applications, you need only configure the *ppl.remotes* file. Adding the correct information to this file will allow you to set up point-to-point serial connections to multiple remote nodes. The *ppl.users* file allows you to configure a mapping between a user name and a remote system. The *ppl.ipool* file is used only with modem pools, where the host system dynamically assigns IP addresses to incoming calls.

Each PPL configuration file is a simple ASCII file that the super-user creates (owns). The files are located in the **/etc/ppl** directory. A template is provided for *ppl.remotes*.

NOTE: Of the three configuration files, only *ppl.remotes* is required for PPL operation. *ppl.users* and *ppl.ipool* are optional files. For dial-out connections, entries are required in the UUCP *Devices* and *Dialers* files.

When to Use Dial-In or Dial-Out with Login

This configuration allows different users from different hosts to login to a dial-in system, each specifying a different user login and host address.

If the called (dial-in) host has a getty running on the serial port to which the modem is attached, the calling host can establish a modem connection first, then log in and run *ppl* using the "log in info" and "command name" sequence in the *ppl.remotes* file. The *ppl* process should be running on both local and remote hosts after the connection is made.

When to Use Dial-In or Dial-Out without Login

This configuration is useful when the remote system has a dedicated SLIP line connected to a modem. The preset SLIP line may be started on the remote host at boot time or anytime thereafter. SLIP is easier to configure, because login information on the command line is not required. This configuration, however, does not provide security on the remote host, since no login is required.

If the called (dial-in) host already has a preset SLIP line, (*ppl* is running continuously on the called host), then the calling (dial-out) host does not need to specify the "log in info" and "command name" in the *ppl.remotes* file.

The preset SLIP line is like a LAN card that a system administrator has configured to be constantly up and running. In this case, *ppl* establishes the SLIP connection immediately when a dial-in request is made. The IP addresses on each system must be pre-configured prior to initiating the dial-in. The called host views each caller as having the same IP address. For configuration details, refer to the *ppl.remotes* templates: dial-out without login and dial-in to preset SLIP line.

Editing the ppl.remotes File

The *ppl.remotes* file is a required file for all PPL connections. It is the central database used by the *ppl* program. *ppl.remotes* contains an entry for each remote host to which *ppl* can establish a connection. The entry contains parameters that allow *ppl* to set up the connection.

Each entry consists of a multi-line form. Depending on the type of connection you are permitting — direct, dial-in, dial-out, or both dial-in and dial-out — you enter different data. However, the format of the form is the same for all four types. Included in this section are examples of *ppl.remotes* files for:

- first-time PPL configuration,
- direct connection,
- dial-in with login,
- dial-in to preset SLIP line,
- dial-in over DTCs,
- dial-out with login,
- dial-out without login.

A template for *ppl.remotes* exists in the /etc/ppl directory. It provides a blank form you may copy for each *ppl.remotes* file entry. The filled-in forms may be concatenated in any order.

Following is the format of a *ppl.remotes* entry form:

```
#   your comments here
        # remote hostname or Internet address
        # local hostname or Internet address
        # Internet mask
        # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
        # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
        # UUCP system name
        # line parity [EVEN] [ODD] [NONE]
        # line speed
        # serial line
        # phone number
        # modem control available
        # log in info
        # command name
```

The first line is reserved for comments describing the entry. If necessary, use additional lines for comments. Begin each comment line with a "# ".

The other lines are for entering data. They begin with a data field, followed by a comment that describes the data on that line. Depending on the type of connection, some data fields may not be used. However, all lines must be present for the form to be read by *ppl*.

NOTE:

Do not delete or edit the text of the built-in comment lines, because they are used by *ppl* to parse the *ppl.remotes* file.

Following is a description of each data field:

remote hostname or Internet address (required). This identifies the remote host. It may specify a hostname or alias, provided the name or alias can be resolved to an Internet address (by /etc/hosts, the NIS, or domain name server). It may also be the Internet address in dot notation. This field is required for all connections.

local hostname or Internet address (optional). This identifies the local host. It may specify a hostname or alias, provided the name or alias can be resolved to an Internet address (by /etc/hosts, the NIS, or domain name server). It may also be the Internet address in dot notation. This field is optional for all connections. If the field is left blank, *ppl* assigns a local Internet address from *ppl.ipool*.

The Internet address for each SLIP interface must be unique in the system. That is, SLIP must have a different Internet address from that of the LAN interface on the same system. Also, the SLIP and LAN interfaces must be in different subnets.

Internet mask (optional). This specifies a mask which may be used to set up an IP subnet. If left blank, the mask is derived from the local Internet address.

protocol. [SLIP] [ASLIPC] [ASLIPS] [PPP] [CSLIPA] [CSLIP] **(required).** This specifies the protocol to be used for the connection: Serial Line Internet Protocol (SLIP); Abbreviated Serial Line Internet Protocol Client (ASLIPC); Abbreviated Serial Line Internet Protocol Server (ASLIPS); Point-to-Point Protocol (PPP); Automatic Switching to CSLIP (CLSIPA); Compressed header SLIP (CSLIP). This field is required for all connections For more information about these protocols, refer to the section, "Which Protocol to Use," in chapter 1. HP recommends using CSLIP whenever possible, otherwise use SLIP.

NOTE:

Currently, only the SLIP, CSLIP, and CSLIPA options are recommended. PPP must not be specified, since PPL does not support the PPP protocol.

type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT] **(required).** This specifies the connection type. It is required for all connections. For a direct connection (hardwired), enter DIRECT; for dial-in, DIALIN; for dial-out, DIALOUT. To accommodate either dial-in or dial-out, enter DIALIN & DIALOUT.

UUCP system name (optional). This field is **optional** for dial-out. It is intended for experienced UUCP users. If you specify a UUCP system name, the *Devices, Dialers,* and *Systems* files are used for placing a dial-out call. The *Systems* file is scanned for entries matching the UUCP system name. If you specify a UUCP system name, then you should leave the line speed, serial line, and phone number fields blank in *ppl.remotes*. A corresponding entry for that UUCP system name must also be included in the *Devices* file.

You should leave the UUCP system name blank for:

- dial-in,
- direct connect,
- dial-out with speed, serial line, and phone number specified in *ppl.remotes*.

  For dial-out, you must still include entries in the *Devices* and *Dialers* files.

line parity [EVEN] [ODD] [NONE] (optional). This field is used for dial-out and is optional. It specifies the line parity to be used after the modem connection has been established.

line speed (required for dial-out without UUCP files). This field is required if you are using dial-out, but not using a UUCP system name, or if you are using a direct connection. It specifies the line speed to be used before and after the modem connection is established. If you are using a UUCP system name, *ppl* gets the line speed from the UUCP *Systems* file.

In general, the supported baud rate is limited by the system hardware. For example, some MUX cards support a maximum speed of 19200 baud. Series 700 built-in serial ports support a maximum speed of 57600 baud. PPL supports up to 57600 baud in the software if the hardware can support this speed. Please refer to your hardware documentation.

serial line (required with UUCP files). This field is required in all cases, unless you specify a UUCP system name for dial-out. Enter the name of a specific `/dev` node (device file) to be used for the connection (for example, `tty0p7` or `tty0p4`). For dial-in, the device file specified should have an entry in `/etc/inittab` to be respawned and a getty already running on this device file. If you are using a UUCP system name for dial-out, *ppl* gets the serial line from the UUCP *Devices* file.

phone number (required for dial-out without UUCP files). This field is required for dial-out if you are not using a UUCP system name. Enter the phone number of the remote host (for example, `5554423022` or `555-442-3022`). The entry may consist of numeric characters and "-" or "=" symbols. A "-" specifies a 5-second pause. An "=" specifies wait for a secondary dial tone. If you are using a UUCP system name, *ppl* gets the number from the UUCP *Systems* file.

modem control available (required for dial-out and direct). This field is required for all dial-out and direct connections. It controls the use of modem signals during and after the dialing phase of *ppl*. If you specified DIRECT in the options field above, enter `NO` here. If you specified DIALOUT or DIALIN & DIALOUT, enter `YES` here. Please refer to the data communication devices support documentation for information about the modems that termio supports. In general, MultiTech and Hayes Smartmodem are supported.

log in info (required for dial-out with login). This field is required for dial-out connections with login. It supplies the dialogue needed to login to the called host. The syntax is identical to the login entry in the UUCP *Systems* file. It includes a series of expect/send strings, separated by spaces.

command name. This field is required for dial-out connections with login. This should be the system name or IP address of the local host. The command name supplies the dialogue needed to invoke the serial line IP protocol on the called host. (Note that, for an IP link to be established, compatible protocols must be running on each side of the line.) The field uses the same syntax as the *log in info* field above.

First-Time PPL Configuration

If you are configuring and running PPL for the first time:

1 Review the list of configuration suggestions.

2 Study the sample *ppl.remotes* files.

3 Follow the procedure for starting PPL and verifying the connection.

The following suggestions will help you configure PPL for the first time.

- Run the *cu* command to verify the hardware, modem, and cable are working correctly.

- Use SLIP protocols on both endpoints.

- Do not specify a UUCP system name in *ppl.remotes* unless you are familiar with UUCP files. Instead, specify device file, speed, and phone number in *ppl.remotes* on the calling host.

- Assign unique IP addresses for SLIP links to avoid routing ambiguities. You should give the endpoints of the SLIP link addresses on a different IP network (subnet) number than the network (or subnet) connecting the systems over Ethernet, FDDI, etc.

  For example, if the Ethernet interfaces are at 15.13.136.12 and 15.13.136.24, the SLIP interfaces should NOT be on the 15.13.136 subnet. Instead, you should pick unassigned IP addresses with a different network or subnet. A good choice for one end of the SLIP link is 193.10.11.1 and the other 193.10.11.2. This way, normal services will continue to flow across the Ethernet as usual.

- Do not specify the "log in info" and "command name," unless you are already

familiar with the syntax of UUCP expect/send strings.

- You do not need to use *ppl.users* or *ppl.ipool* for this simple example.

- Use verbose mode *ppl -v* if you have super-user capability. *ppl* records its actions in the log file, **/var/ppl/log**.

Here is an example of the *ppl.remotes* file that does not use a UUCP system name for a dial-out connection to a preset SLIP line.

```
193.10.11.2    # remote hostname or Internet address
193.10.11.1    # local hostname or Internet address
               # Internet mask
SLIP           # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIALOUT        # type [DIRECT][DIALIN][DIALOUT][DIALIN & DIALOUT]
               # UUCP system name
NONE           # line parity [EVEN] [ODD] [NONE]
9600           # line speed
/dev/cul0b2    # serial line
5551212        # phone number
YES            # modem control available [YES] [NO]
               # log in info
               # command name
```

Here is the *ppl.remotes* file of the corresponding incoming call.

```
193.10.11.1    # remote hostname or Internet address
193.10.11.2    # local hostname or Internet address
               # Internet mask
SLIP           # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIRECT         # type [DIRECT][DIALIN][DIALOUT][DIALIN & DIALOUT]
               # UUCP system name
NONE           # line parity [EVEN] [ODD] [NONE]
               # line speed
/dev/ttyd0b1   # serial line
               # phone number
YES            # modem control available [YES] [NO]
               # log in info
               # command name
```

NOTE: The type is DIRECT not DIALIN, because this is considered a preset SLIP line. **/dev/ttyd0b1** should have a dial-in minor number instead of direct connect, since it is connected to a modem.

Follow these steps to start PPL:

1   Edit the *ppl.remotes* file on both the local and remote hosts.

2   Make sure that no getty is running on **ttyd0b1** on the incoming host.

3   Execute the following command on the incoming (called) host:

```
ppl -v 0o 193.10.11.1 &
```

ppl should complete successfully and display the message, "Initialization complete, running protocol."

Stop here if *ppl* fails. Check the log file for error messages.

4 Execute the following command on the calling host:

```
ppl -v -o 193.10.11.2
```

ppl should complete successfully and display the message, "Initialization complete, running protocol."

Stop here if *ppl* fails. Check the log file for error messages.

5 Verify the SLIP connection by executing the following command on the calling host:

```
ping 193.10.11.2
```

If ping fails, refer to chapter 4, "Troubleshooting PPL."

Sample ppl.remotes for Direct Connections

Here is an example of a *ppl.remotes* file on both the local and remote hosts for direct connections. Generally, such a connection is established at boot time, using a start-up script. This example assumes the necessary cable connections have been made between ports on each machine. The sample also assumes the system, minnie, is configured with the appropriate direct connect device file.

You do not need to use a UUCP system name. No getty should be running on the serial port at either end. The "log in info" and "command name" are not required.

```
#   Direct connect
#
#       Sample entry in ppl.remotes file on machine minnie.
#       Sets up direct SLIP connection to mickey. To users
#       at both machines, minnie and mickey appear to be
#       on same LAN.
#
mickey # remote hostname or Internet address
minnie # local hostname or Internet address
       # Internet mask
SLIP   # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIRECT # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
       # UUCP system name
ODD    # line parity [EVEN] [ODD] [NONE]
9600   # line speed
tty04  # serial line
       # phone number
NO     # modem control available
       # log in info
       # command name
```

ppl may be executed at boot time by including the following in a start-up script on minnie:

```
ppl -o mickey
```

Sample ppl.remotes for Dial-In with Login

Here is an example of a *ppl.remotes* file on both the local and remote hosts for dial-in connections with login. Keep in mind that the local host is the HP 9000 you are configuring. The remote host is the machine that is dialing in.

This example assumes the system, biggy, is configured with the appropriate dial-in device file and that a modem is online. It also assumes biggy has a getty running on the serial port to which the modem is attached.

```
#   Dial-in with login.
#
#       Sample entry in ppl.remotes file on machine biggy.
#       Enables machine smally to dial in and have a SLIP
#       connection to biggy.  The user at smally must have
#       a login on biggy.  Connection works as follows:
#       Machine smally establishes a modem connection.  User
#       logs in and runs ppl.  ppl converts the serial line
#       to SLIP.
#
smally # remote hostname or Internet address
biggy  # local hostname or Internet address
       # Internet mask
SLIP   # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIALIN # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
       # UUCP system name
NONE   # line parity [EVEN] [ODD] [NONE]
       # line speed
       # serial line
       # phone number
NO     # modem control available
       # log in info
       # command name
```

Sample ppl.remotes for Dial-In to Preset SLIP Line

Here is an example of a *ppl.remotes* file on both the local and remote hosts for dial-in connections to preset SLIP lines. This example assumes the system, pizza, is configured with the appropriate dial-in device file and that a modem is online. It also assumes pizza does not have a getty running on the serial port to which the modem is attached.

NOTE:

In this preset case, the type is DIRECT not DIALIN, but the minor number of the serial line is DIALIN not DIRECT. Always configure the preset SLIP line before attempting an outgoing connection.

```
#   Dial-in to preset SLIP line.
#
#       Sample entry in ppl.remotes file on machine pizza.
#       Enables machines of type pasta to dial in and have
#       a SLIP connection to pizza (pizza views each caller
#       as the same IP address).  Connection works as fol-
#       lows:  ppl runs continuously on pizza.  When a dial-
#       in request is heard at ttyd01, ppl establishes the
#       SLIP connection immediately. #
pasta  # remote hostname or Internet address
pizza  # local hostname or Internet address
       # Internet mask
SLIP   # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIRECT # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
       # UUCP system name
NONE   # line parity [EVEN] [ODD] [NONE]
2400   # line speed
ttyd01 # serial line
       # phone number
YES    # modem control available
       # log in info
       # command name
```

ppl may be started on pizza at boot time or anytime thereafter. It may be executed at boot time by including the following in a start-up script:

```
ppl -o pasta &
```

Sample ppl.remotes for Dial-In over DTCs

The only supported configuration of SLIP over DTCs is dial-in. The following steps are necessary for configuring dial-in using a DTC and SLIP.

1 Create and edit *ppl.remotes* on both the local and remote hosts.

2 Configure the DTC using HP OpenView DTC Manager.

   Set binary mode and the telnet port identification feature on the DTC port using the HP OpenView DTC Manager, version 14.0 or later. The DTC configuration should contain the following entries:

```
DTC PARAMETERS                      VALUE USED
Attached Device                     US modem
Modem behavior                      Standard DCE
Escape from data transfer char CTRL K (^K)
Max number of sessions              5
Default destination                 IP address of remote host
Enable port                         Yes
Enable automatic connection         Yes
```

```
Binary mode                    Yes
Switching allowed              No
```

3 Configure the nailed telnet port in the **/etc/ddfa/dp** configuration file, using the DTC Device File Access (DDFA) utility on the called host.

4 Run *dpp* on the **dp** file to create the nailed DTC port on the called host. For more information, refer to the DDFA reference manual.

Here is an example of the *ppl.remotes* file for the calling host.

```
192.1.1.5       # remote hostname or Internet address
192.1.1.1       # local hostname or Internet address
                # Internet mask
SLIP            # protocol [SLIP] [ASLIPC] [ASLIPS] [PPP] [CSLIPA] [CSLIP]
DIALOUT         # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
                # UUCP system name
NONE            # line parity [EVEN] [ODD] [NONE]
9600            # line speed
/dev/cul0p1·    # serial line
1234567         # phone number
YES             # modem control available [YES] [NO]
ogin: slipdtc ssword: slip22 slip>          # log in info
"" /usr/bin/ppl\s-i\s192.1.1.1 running      # command name
```

Here is the *ppl.remotes* file for the corresponding called host.

```
192.1.1.1       # remote hostname or Internet address
192.1.1.5       # local hostname or Internet address
                # Internet mask
SLIP            # protocol [SLIP] [ASLIPC] [ASLIPS] [PPP] [CSLIPA] [CSLIP]
DIALIN          # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
                # UUCP system name
NONE            # line parity [EVEN] [ODD] [NONE]
9600            # line speed
/dev/dtcb1p1    # serial line
                # phone number
YES             # modem control available [YES] [NO]
                # log in info
                # command name
```

/dev/dtcb1p1 is the device filename of a nailed DTC port as defined by the DDFA utility.

Sample ppl.remotes for Dial-Out with Login

Here is an example of a *ppl.remotes* file for dial-out connections with login. Again, the local host is the HP 9000 you are configuring. The remote host is the machine to which HP 9000 users are dialing out. Keep in mind that, in both cases, *ppl* does the login.

This example does not use a UUCP system name. It assumes the system, misty, has the necessary UUCP configuration files entries for dial-out to the system, foggy. It also assumes misty is configured with the appropriate dial-out device file and that a modem is online.

```
#   Dial-out with login (no UUCP system name).
#
#       Sample entry in ppl.remotes file on machine misty.
#       Enables machine misty to dial out and have a SLIP
#       connection to foggy.  The IP addresses of misty and
#       foggy are known to each machine beforehand.  Con-
#       nection works as follows:  User at misty runs ppl,
#       which dials out to foggy.  ppl logs in and runs SLIP
#       on foggy using login and command sequence listed in
#       this entry.
#
foggy    # remote hostname or Internet address
misty    # local hostname or Internet address
         # Internet mask
SLIP     # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIALOUT  # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
         # UUCP system name
NONE     # line parity [EVEN] [ODD] [NONE]
9600     # line speed
/dev/ttyd0b2 # serial line
5551212  # phone number
YES      # modem control available
ogin: guest  ssword: guest guest_tristar # log in info
"" ppl\s-i\smisty running                # command name
```

The login and password on foggy are both *guest*. The "log in info" field is using the UUCP expect/send strings. The command that executes SLIP on foggy is *ppl -i misty*. To clearly understand how the expect/send dialog occurs, execute the *ppl* command with the *-v* option. The command displays "running" after starting up. The syntax for the login, password and command sequences in *ppl.remotes* must follow UUCP syntax.

Sample ppl.remotes for Dial-Out without Login

Here is an example of a *ppl.remotes* file for dial-out connections without login. This example assumes the system, slinky, has the necessary UUCP configuration file entries for dial-out to the system, stuffy. Here, stuffy is known to the UUCP *Systems* file as hiho. This example also assumes slinky is configured with the appropriate dial-out device file and that a modem is online.

```
#   Dial-out without login (UUCP system name).
#
#       Sample entry in ppl.remotes file on machine slinky.
#       Enables machine slinky to dial out and have a SLIP
#       connection to stuffy.  stuffy has a dedicated SLIP
#       line connected to a modem that answers to a phone
#       number known to slinky.  The IP addresses of slinky
#       and stuffy are known to each machine beforehand.
#       Connection works as follows:  User at slinky runs
#       ppl, which dials out to the preset SLIP line.  The
#       connection is established immediately.
#
stuffy # remote hostname or Internet address
slinky # local hostname or Internet address
       # Internet mask
SLIP   # protocol [SLIP][ASLIPC][ASLIPS][PPP][CSLIPA][CSLIP]
DIALOUT# type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
hiho   # UUCP system name
NONE   # line parity [EVEN] [ODD] [NONE] 9600
       # line speed
       # serial line
       # phone number
YES    # modem control available
       # log in info
       # command name
```

Here is the entry in the UUCP *Systems* file for this example:

```
hiho Any;1 ACU 9600 5551212 ogin: notused ssword: xyz
```

This hiho entry in the *Systems* file indicates that a Hayes Smartmodem with 9600 baud rate should be used and the number to call is 5551212.

Here is the entry in the UUCP *Devices* file for this example:

```
hiho cul0b2 - 9600 hayes
```

This entry in the *Devices* file indicates that **cul0b2** is the corresponding device file to use.

Creating the ppl.users File

The *ppl.users* file may be used for dial-in or dial-out. It maps each user name to a single remote host. This allows a dial-in user to invoke *ppl* without identifying the remote host that is calling. It allows a dial-out user to invoke *ppl* without specifying the remote host that is being called.

The format of a *ppl.users* file entry is:

```
<username>  <SLIP hostname>
```

where *username* is the login of the user, *SLIP hostname* is the identifier of the remote host. The *SLIP hostname* may be either an Internet address in dot notation, or host alias, provided the alias is known to the local host. The alias must be listed in `/etc/hosts` or the NIS database, or it must be resolved using the domain name server.

The SLIP hostname or SLIP IP address should be different from the Ethernet (FDDI, etc.) IP hostname address on the same system.

When *ppl* is invoked, it finds the hostname associated with the user's login. It uses this name to access connection parameters stored in *ppl.remotes*.

Following are sample *ppl.users* file entries:

```
robd    columbia.nasa.gov
davey   elk
rons    reindeer
susie   123.33.42.1
```

Creating the ppl.ipool File

The *ppl.ipool* file may be used for dial-in. It specifies a pool of local Internet addresses to be used for modem connections.

This file is only necessary if you are using a modem pool. Generally, a modem pool is set up when the UUCP subsystem is configured at boot time. This allows the host to assign modem ports to dial-in users as needed.

When *ppl* receives a dial-in request, it identifies the remote host that is calling, then checks if a specific port and a local Internet address are allocated for use by that host. Similarly, *ppl* selects a local Internet address from *ppl.ipool*.

Format of a *ppl.ipool* file entry is:

```
<Internet_address>
```

where *Internet_address* is any valid Internet address assigned to the local host. It must be in dot notation.

Following are sample *ppl.ipool* file entries:

```
123.22.33.11
123.22.33.112
```

3

Using PPL

This chapter explains how to use PPL features and how to monitor PPL events. This chapter covers:

- Using the Dial-In feature.
- Using the Dial-Out feature.
- Using a Direct Connection.
- Recording connection information with a Billing File.
- Checking connection status.

Using the Dial-in Feature

Any remote host or PC with appropriate dial-out capability may dial-in to an
HP 9000 serial IP line. HP systems, however, require:

- The remote host to have an entry in the *ppl.remotes* file of the HP 9000.

- The remote host to use the serial IP protocol specified in that *ppl.remotes* entry.

- For dial-in with login, the remote user must have a login on the HP 9000.

Using the Dial-In Feature with Login to Establish a Connection

The dial-in host with login must have a getty running on the device file used
for the SLIP connection and an entry in the **/etc/inittab** to respawn the
getty. For example, **/dev/ttyd0b1** is the device file specified in the
ppl.remotes file and the speed is 9600 baud.

- **/etc/inittab** should have an entry that looks like the following:

  ```
  slip:2:respawn:/etc/getty -h ttyd0b1 9600M
  ```

- A getty should be running on **/dev/ttyd0b1** before a SLIP connection is
  made. Do the following to check this:

  ```
  >ps -ef | grep getty
  /etc/getty -h ttyd0b1 9600M
  ```

- **/etc/gettydefs** should have the following definition:

  ```
  9600M      # B9600 OPOST ONCLR CS8
             # B9600 CS8 ISTRIP TAB3 HUPCL ICRNL IXON IXANY
             # OPOST ONCLR ISIG ICANON ECHO ECHOK ISTRIP INPCK
  #login:#9600M
  ```

  CS8 (8 bits) is required for SLIP/CSLIP connections.

The Remote Host is an HP System

If both the calling and called hosts are HP systems, then establishing a SLIP
connection is transparent to the users. The dial-in feature requires the calling
host to specify the login information and command name in the *ppl.remotes*
of the calling host. Users only need to execute the *ppl* command on the
calling host and establishing the SLIP connection is automatic.

The following steps describe the sequence of events that an HP dial-in host expects while establishing a SLIP connection with login. Steps 1 through 3 are performed automatically by *ppl* on the calling host.

1   Initiate a modem connection to the HP 9000 dial-in system.

2   Login to the HP 9000 dial-in host using the login information specified in the *ppl.remotes* file on the calling host.

3   Start *ppl* on the HP 9000 dial-in system at the HP-UX shell prompt using the command line information specified in the *ppl.remotes* file on the calling host.

    The *ppl* program should be running in the background on both the calling and called hosts after the SLIP connection is established successfully.

4   Verify the SLIP connection. For example, *ping* to the called host, hopper, over the SLIP connection by entering:

```
ping hopper
```

The Remote Host is a Non-HP System

If the calling host is not an HP system, then the user on the calling host must invoke a SLIP program which is equivalent to *ppl* on the calling host. Depending on the equivalent SLIP program used, steps 1 through 3 described above may or may not be performed automatically by the program.

Using the Dial-In Feature without Login to Establish a Connection

The dial-in feature without login is equivalent to dialing into an HP 9000 preset line. To set up a preset SLIP line on the dial-in host, use the following procedure.

1   Make sure the type DIRECT (not DIALIN) in *ppl.remotes*.

2   Check the minor number of the device file used. It should indicate dial-in, not direct.

3   Make sure no getty is running on the device file that is used for the SLIP connection.

4   The HP 9000 dial-in host must invoke *ppl* to set up the preset SLIP line before a remote host tries to establish a SLIP connection. Use the *-o* option for preset line configuration, instead of the *-i* option as follows:

```
ppl -o <remote SLIP IP address> &
```

5 The calling host dials in the number of the preset HP 9000 SLIP line. This number is generally provided by the HP 9000 system administrator.

6 If the calling host is an HP 9000 system, start *ppl* to initiate the modem connection. If the calling host is a non-HP 9000 system, invoke a SLIP program equivalent to *ppl*.

The SLIP connection is available immediately after the modem connection is established. *ppl* should be running in the background on both the local and remote hosts after a SLIP connection is made.

Terminating a Dial-In Connection

The IP connection is lost when the line is hung up from either end. The *ppl* program senses loss of carrier and terminates gracefully.

The HP 9000 user can also terminate the IP connection with one of the lock files that *ppl* creates in the **/var/ppl** directory. The lock files are of the form *ppl.laddr* and *ppl.raddr*, where *laddr* and *raddr* are Internet addresses (in hexadecimal notation) of the IP connection.

To kill *ppl* using the lock files, type the following at the shell prompt:

```
sh lock_file_name
```

where *lock_file_name* is the name of either the remote or local lock file. The lock files have a shell script that causes *ppl* to exit gracefully.

Following is a sample lockfile:

```
kill -15  2213 # rhost_name=slip4 creator=joelw
# tty=/dev/tty0 Linet=slip1 Rinet=15.255.136.6 protocol=SLIP
```

Using the Dial-out Feature

The PPL dial-out feature is available to any HP 9000 user. For dial-out only, PPL utilizes the UUCP configuration files. If you specify a UUCP system name, the *Devices*, *Dialers* and *Systems* files are consulted. If you do not specify a UUCP system name, only the *Devices* and *Dialers* files are used.

Using the Dial-Out Feature to Establish a Connection

Use the following procedure for dial-out connections with or without login. From the HP 9000 user's perspective, the two types of connections are the same. To establish a dial-out SLIP connection, follow these steps.

1 Start *ppl* at the HP-UX shell prompt and specify the called host by name or Internet address.

   For example, if jumbo is the remote SLIP system and chip is the local user's login name invoking *ppl*, enter:

   ```
   ppl -o <jumbo> # jumbo's ip_address defined in /etc/hosts
   ```

   If jumbo is not specified in the *ppl* command line, then *ppl* checks the *ppl.users* file for an entry with the user's login chip. If an entry is found, *ppl* used the associated SLIP hostname to search *ppl.remotes* for connection parameters.

2 After *ppl* starts, a display appears like the following:

   ```
   local ppl: starting for chip at Fri Jan 15 12:16:19 1990
   ```

   ppl establishes a modem connection. If a login and a command sequence are provided in *ppl.remotes*, *ppl* logs in and starts the desired protocol on the remote host. For connections without login, *ppl* must already be running on the remote host.

   After the IP connection is established, the following is displayed:

   ```
   initialization complete, protocol running
   ```

3 A SLIP connection has been established and the HP-UX shell prompt returns. Use the *ping* command to verify the SLIP connection to a remote host.

Dialing Out with Login

The login information and command line in the *ppl.remotes* file must be specified. The login information supplies the dialog needed to login on the remote system. It includes the login name, password, and the expected shell prompt. The syntax is identical to the UUCP *Systems* file.

The command name supplies the dialog needed to run a complementary SLIP protocol command on the remote system. It includes a series of expect/send strings separated by spaces. For remote HP 9000 systems, this command is typically used to invoke the *ppl* program on the remote system. The expect string is "running" to indicate PPL has been initialized successfully on the remote system.

If the remote host is a non-HP system, then the command name should specify the equivalent SLIP program on the remote host and the expect string may be one of the output strings from the equivalent SLIP program.

Dialing Out to an Unknown Remote Address

An HP 9000 user may want to dial out to a remote SLIP connection, but the remote IP address is unknown. This may be the case if the remote machine has a modem pool for dial-in. Similar to the way an HP 9000 uses *ppl.ipool*, the remote machine may have a scheme to dynamically assign local addresses.

In this situation, PPL dial-out does not work initially, because a preconfigured remote address cannot exist in *ppl.remotes*. For PPL dial-out to work, use this procedure.

1   Connect to the modem from which you will dial out, for example:

```
cu -s2400 -l tty03 dir
```

2   Send a command string to the modem to dial out to the remote host, for example:

```
atdt5551212
```

3   Assuming there is a getty listening at that port, login to the remote system.

4   Use the appropriate command sequence to execute SLIP on the remote system. When the program starts, note the local IP address used (this is the remote address from the perspective of the HP 9000).

5   At the HP 9000, open another window and edit *ppl.remotes* to include the remote IP address.

6   Back in the first window, start *ppl* by entering:

```
~&ppl -o <remote SLIP IP address>; sleep 9999
```

7   In the second window, verify the SLIP connection by using the command:

```
ping <remote SLIP IP address>
```

Terminating a Dial-Out Connection

Once started, the *ppl* program runs until the carrier signal is lost or the program is stopped. The user may stop the program by using lock files. You can use the same procedure that is described in the section, "Terminating a Dial-In Connection," in this chapter.

Using a Direct Connection

The HP 9000 system administrator may set up a direct PPL connection. This can be initialized at boot time or anytime thereafter. Direct connections do not require modems on either end. They are hardwired and, therefore, the cables used may be different from system to system.

To set up a direct connection at boot time, the super-user may include the following line in a start-up script:

```
ppl -o <remote SLIP IP address>
```

NOTE:

The two machines should be connected via serial ports. There should not be a getty running on the serial port of either machine.

Recording Connection Information with a Billing File

You can set up a file to record connection information such as user, start time, and end time. Such a file is useful for billing purposes. To use this feature, create an empty ASCII file (zero length) named *bill* in the **/var/ppl** directory.

After termination of each invocation of *ppl*, information for that connection is entered in the file. Records within the file are ordered according to end time.

The following is a sample billing file (containing two records):

; 11/19/89 15:07:29 ; 11/19/89 14:37:27 ; goofy ; 70.0.0.4 ; joelw ; ttyd0 ; SLIP;
; 11/19/89 15:18:43 ; 11/19/89 09:33:10 ; goofy ; joelw_pc ; joelw ; ttyd3 ; SLIP;

Each field is separated by a semicolon (;). In order, the fields are:

- End time.

- Start time.

- Local SLIP host name (or SLIP Internet address).

- Remote SLIP host name (or SLIP Internet address).

- Login of user who invoked *ppl*.

- tty port used.

- Protocol used.

NOTE: The billing file is not pruned automatically. Be sure to check and/or prune the file at intervals appropriate to PPL usage.

To disable this feature, simply remove the billing file. If the file does not exist, PPL ignores this feature.

Checking Connection Status

You can configure PPL to maintain a status file of active connections. The file is binary, but can be displayed with *pplstat*.

Creating a Status File

The status file can be set up at boot time or anytime thereafter. To do so, create a zero-length ASCII file named *ptmp* in the **/var/ppl** directory.

The *ptmp* file is maintained by invocations of *ppl*. Each time a new serial port is used, an entry is added to the file for that port. Any subsequent *ppl* use of that serial port results in an update to the file.

To create and initialize the file at boot time, add the following to a start-up script:

```
PTMP = /var/ppl/ptmp
if [ -f $PTMP ]
then
        rm $PTMP
        touch $PTMP
        chmod 644 $PTMP
fi
```

To create the file anytime after boot time, enter the following:

```
cd /var/ppl
touch ptmp
chmod 644 ptmp
```

To disable this feature, simply remove the **/var/ppl/ptmp** file. If the file does not exist, PPL ignores this feature.

Displaying a Status File

The *ptmp* file is binary. To display the file, use *pplstat* by entering the following command:

```
pplstat
```

A display appears like the following:

```
log_name      joelw
rhost_name    slip4
```

```
system_name    slip4
Linetaddr      slip1
Rinetaddr      43.0.0.2
tty            /dev/tty0
ni             /dev/nio
status         SLIP
killfile (r)   /var/ppl/ppl.2b000002
killfile (l)   /var/ppl/ppl.2b000001
start_time     Wed Jan 10 13:24:17: 1990
```

NOTE: *pplstat* only displays information for connections that are not idle.

4

Troubleshooting PPL

Follow these steps to troubleshoot PPL.

1   Make sure the modem is working correctly by executing the command:

```
cu -l <device name>
```

If you cannot call up another system, the modem settings may be incorrect.

2   Use the correct modem cables for each platform.

Modem cables are different from hardwired cables, so Series 700 systems may require different cabling than Series 800 systems.

3   If the command *cu* is working correctly, but *ppl* is not, the possible problems are:

a   Routing problems.

Check the Internet addresses of the PPL and LAN interfaces. They must have different Internet addresses and must be in different subnets. If the IP addresses are the same for both LAN and SLIP, this will cause routing ambiguity.

b   Protocol mismatch on local and remote systems.

c   Modem settings.

Make sure you are using CS8 and make sure that Xon/Xoff is turned off in the **/etc/gettydefs** file.

d   Device file major and minor numbers.

To verify that major and minor numbers of the device file used for each platform are correct, execute the command:

```
ls -l /dev/*
```

The PPL device file should be **/dev/ni**, the major number, 56, and the minor number, 0. If you cannot locate this device file, create it using the following command:

```
mknod /dev/ni c 56 0 #network interface
```

4   Use the -T tracefile option to turn on PPL tracing.

ppl performs user space level tracing for SLIP connections that it creates. All data that are going from IP to the tty line and back are traced. This option is different from the system tracing utility *nettl*, which performs tracing for the entire system.

5   Use the -v option to turn on verbose mode.

This options requires super-user capability. Verbose debugging information is displayed on the terminal and logged in the **/var/ppl/log** file. Secure information is included.

The log file is not pruned automatically. Be sure to check and/or prune the file at intervals appropriate to *ppl* usage, because the log file grows without limitations.

You can also use the LAN/9000 diagnostics *ping* and *netstat* to help troubleshoot SLIP connections, because they work over a PPL link the same as over the LAN. This is also true for network logging and tracing tools. No special accommodation must be made to use network diagnostics, logging and tracing over IP serial lines.

Index

Index

HEWLETT®
PACKARD

**Manufacturing
Part No.
98194-90051**

Manual Part No.
98194-90051