User's Guide

PortServer II®

Intelligent Network Communications and Terminal Server

Table of Contents

Important Information

Federal Communications Commission (FCC) Statements

Radio Frequency Interference (RFI) (FCC 15.105)

The PortServer II has been tested and found to comply with the limits for Class A digital devices pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential environment. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try and correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Labeling Requirements (FCC 15.19)

This device complies with part 15 of FCC rules. Operation is subject to the following two conditions:

1. this device may not cause harmful interference, and
2. this device must accept any interference received, including interference that may cause undesired operation.

Modifications (FCC 15.21)

Changes or modifications to this equipment not expressly approved by Digi International may void the user's authority to operate this equipment.

Cables (FCC 15.27)

This equipment is certified for Class A operation when used with shielded cables.

Industry Canada Compliance Statements

This Class A digital apparatus meets the requirements of the Canadian Interference Causing Equipment Regulations (ICES-003 of Industry Canada, Class A).

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Certification

The Digi International PortServer II Intelligent Network Communications and Terminal Server meets the following standards:

- FCC Part 15, Class A
- ICES-003, Class A
- EN 55022, Class A
- VCCI, Class I
- EN50082-2 Heavy Industry
- UL-1950
- CSA C22.2 No.950
- EN60950

About this User's Guide

This *PortServer II User's Guide* is designed to be read by an installer or network administrator who needs to install PortServer II, configure it for normal operation, and connect it to a network. You should use it with the *PortServer II Command Reference Guide*, which contains information on the operational and configuration commands.

Where do I look for more information?

This User's Guide is divided into eighteen chapters as follows:

Chapter 1 Introduction contains a brief introduction to PortServer II and how you can use it to connect serial devices to a network. Read this chapter for general background information or if you are designing a network that will include PortServer II.

Chapter 2 Operation explains the front panel and connectors. Make sure you understand this chapter before you proceed with installation or configuration.

Chapter 3 Installation. Refer to this chapter for general site requirements and procedures you should follow for connecting up PortServer II.

Chapter 4 Basic Operation describes how to set the basic functions of PortServer II, including its connection to the Ethernet. It also provides a general description of the other parameters that you will need to configure for users, including a description of how to use the configuration commands. It also describes how to provide a menu for each user.

Chapter 5 Configuring Terminals describes how to add connections for "dumb" terminals to PortServer II.

Chapter 6 Configuring Security includes information on configuring manual and automatic logins to PortServer II for all types of users.

Chapter 7 Configuring Multiple Sessions and Multiple Screens describes how to configure Portserver II for terminals that run multiple sessions and multiple screens.

Chapter 8 Configuring WAN Connections describes how to configure connections to WANs (Wide Area Networks), including intra-company networks and the Internet.

Chapter 9 Configuring Modem Connections. This chapter describes how to configure or modify PortServer II to operate over modem links.

Chapter 10 Configuring TCP/IP Routing describes how to configure each of the different types of TCP/IP routing that PortServer II supports.

Chapter 11 Configuring RealPort Connections describes how to configure or modify connections to terminals, printers, and other devices if you have the RealPort protocol running on your host server.

Chapter 12 Configuring SNMP describes what to do if your system supports SNMP (Standard Network Management Protocol) and you want PortServer II to respond to SNMP requests.

Chapter 13 Configuring Printer Connections describes how to configure or modify PortServer II for connection to a printer.

Chapter 14 Configuring Frame Relay describes how to use PortServer II with a Frame Relay service.

Chapter 15 Configuring RADIUS describes to configure PortServer II to use the RADIUS protocol for dial-up connections. You must have RADIUS available on your server.

Chapter 16 Remote Configuration describes how to configure PortServer II from a remote host. You can update both the PortServer II software and your own configuration file in this way.

Chapter 17 Troubleshooting. Refer to this chapter for help if PortServer II is unable to connect to your network, or if it displays an error message.

Chapter 18 Digi Support Services describes various ways that you can contacting Digi International if you have a problem with PortServer II.

The **Glossary** contains a list of technical terms, and their explanations.

Which chapters should I read?

If you are not familiar with PortServer II, you should begin by reading Chapters 1 and 2 which provide a general description of PortServer II and its operation.

If you want to install Portserver II, read Chapter 3.

To configure PortServer II for the first time, read Chapter 4, which provides a “quick start” sequence for configuring the main functions.If you wish to change the initial configuration, or add more devices or users, refer to the relevant information in Chapters 5 through 15.

If you have several PortServer II units to install, or have many similar users, refer to Chapter 17.

Document Conventions

Throughout this User's Guide, we use certain formats and presentations to indicate information of special significance:

Note: A Note gives background or supplementary information. It may also give a hint or reminder that makes a task quicker or easier.

Important: An Important statement contains a step or action that, if overlooked, may cause a problem or incorrect operation.

Caution!

A Caution gives information that is crucial to the correct operation of the equipment. Failure to heed a caution may result in damage to PortServer II and/or the network.

Warning!

A Warning gives vital information. Failure to heed a warning could result in injury to yourself or others, or serious legal liability.

Italic text is used to emphasize a statement, or to give a cross-reference to another chapter of this document or to another document.

`Courier text` indicates a key or sequence of keys on the terminal that you should press.

Other PortServer II Documentation

For further information on PortServer II, refer to the following documents:

- PortServer II Intelligent Network Communications and Terminal Server Command Reference Guide 92000246
- PortServer II Release Notes
- RealPort Device Driver Software Manual for AIX Release 4.1.x 92000235A
- RealPort Device Driver Software Manual for AIX (earlier) 92000196A
- RealPort Device Driver Software Manual for SCO OpenServer System V Release 3.2 92000159A
- RealPort Device Driver Software Manual for Solaris (SPARC) 2.3, 2.4, and Solaris (x86) 2.4 92000184A
- RealPort Device Driver Software Manual for Novell Netware AIO 92000172A

Chapter 1

Introduction

When should I read this Chapter?

Read this chapter if you are unfamiliar with PortServer II, and want an overview of its functions and possible applications.

In this Chapter

This chapter introduces you to the PortServer II and describes its features. It includes the following topics:

Features and Functions

Thank you for purchasing one of Digi International's PortServer II family of communications and terminal servers. PortServer II allows you to connect up 16 serial devices to a single TCP/IP Ethernet network port. Up to 48 additional devices (64 total) can be added by the connection of external expansion modules. You can attach a variety of serial devices, including PCs, Internet links, asynchronous terminals, printers, ISDN terminal adapters, and modems. You can also attach parallel devices such as printers, if you use an expansion module with a parallel port. Because PortServer II is independent of the hardware connected to it, users can access any type of networked server running the TCP/IP protocol suite.

A system configuration diagram with many of PortServer II features in use is shown on the next page:

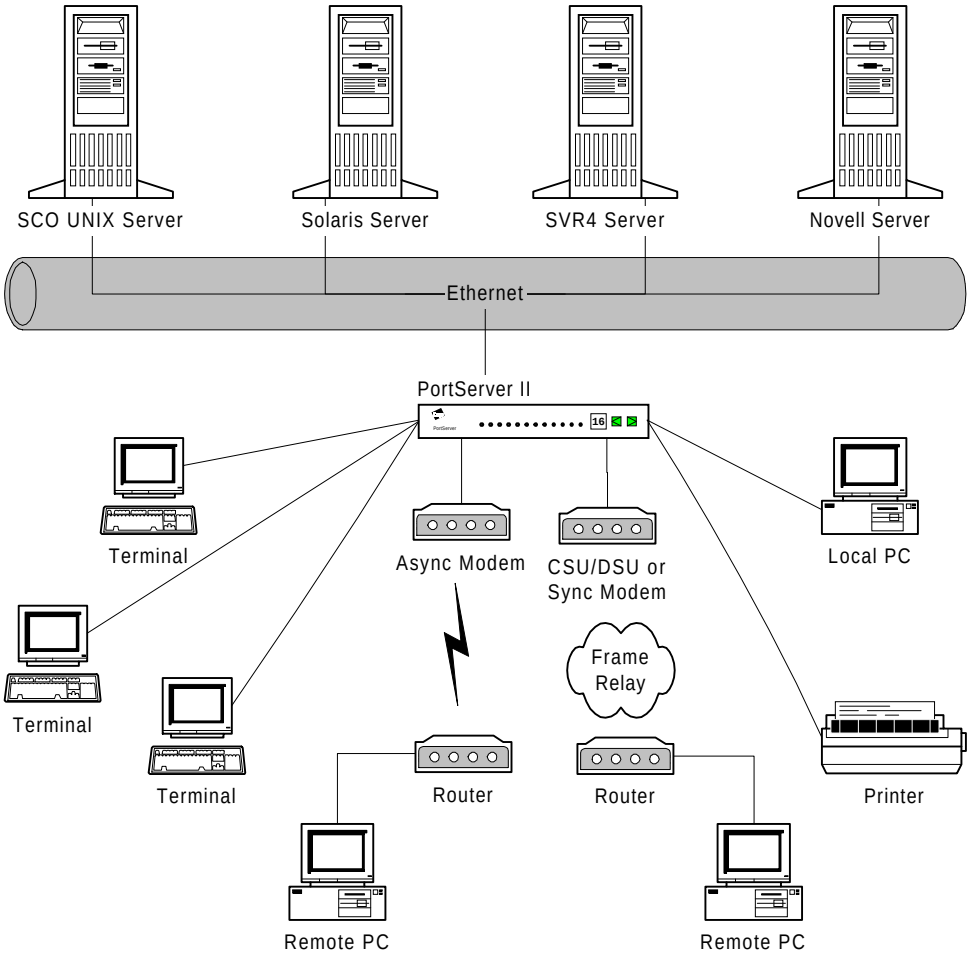

Figure 1 PortServer II Features and Functions

The following peripherals are shown connected to the PortServer II:

- **Terminals**. Any terminal that includes a standard RS-232 serial interface can be connected to a serial port. PortServer II communicates with the host computer through the Ethernet port using TCP/IP. Where multi-user systems are already installed, multiple host sessions are available, up to a limit of nine sessions per port.

PortServer II allows terminals to connect to a server using Telnet, Rlogin, or TTY.

Note: Depending on the terminal type(s), the server may need to be loaded with Digi International's RealPort™ software to allow the server and its applications to make full use of PortServer II's capabilities. See *Chapter 11, Configuring RealPort Connections* for more information.

- **Local PCs**. Any PC can be connected to a PortServer II serial port. The user has full access to all applications on the server. Typically, you run a terminal emulator or SLIP/PPP software package on the PC.
- **Printers**. Any serial printer that supports the RS-232 interface can be connected to a PortServer II serial port. Parallel printers may be connected via the PORTS/8emp expansion module.
- **Asynchronous Modem.** PortServer II can provide connections to remote terminals (including terminal emulations), remote PCs, or Internet service providers. PortServer II supports PPP (Point-to-Point Protocol), SLIP (Serial Line Interface Protocol), and CSLIP (Compressed Serial Line Interface Protocol) connections up to 115.2 Kbps.

More detailed descriptions of the features supported for each of these different types of connection are given later in this chapter.

At the LAN interface, you only require a single Ethernet connection, regardless of how many serial devices are connected to the PortServer II. Ethernet 10BaseT (Twisted Pair) and 10Base2 (BNC coaxial Thinnet) connectors are provided.

Basic Operation

Configuration of serial ports

Each PortServer II serial port can be configured for a specific type of device and/or service, for example, an asynchronous modem with PPP service. Once a port has been configured, it cannot be used with another type of device or service unless you reconfigure it. To maintain system security, configuration can only be carried out at the `root` login level, and other users can only temporarily change the configuration of the serial port to which they are attached.

Configuration information is held in Tables within the PortServer II memory. For example, there is a table of all configured users, a table of connections, and so on.

Terminals and users

Each serial port that has a terminal connected to it must first be configured with the emulation used by the terminal (for example, VT100), and the baud rate and control parameters.

Once the serial port has been configured, you can set up one or more users for the terminal. For each user, you should enter a login name and password. It is also possible to customize the login information. For example, you might give each user a different selection menu on successful login, or they may automatically connect to a particular service or application.

PortServer II allows up to nine separate sessions on each of its serial ports. Short-cut keys are provided to allow a user to temporarily escape from a session to access the PortServer commands.

Printers

Each serial port that has a printer connected to it must be configured with data and control information appropriate for the printer type. PortServer II is compatible with most protocols that can be used to access printers.

Modems and dial-up links

A single modem connection can support any or all of the following:

- Outgoing calls for hosts on the network.
- Outgoing WAN connections initiated by PortServer II.
- Incoming WAN connections.
- Incoming terminal-style connections.

For each modem or other dial-up device that you connect to PortServer II, you must configure its baud rate and control parameters.

Communication protocols supported by PortServer II include PPP, SLIP, and CSLIP. Each of these protocols allows PortServer II to establish a link over standard telephone lines when traffic warrants a call or if a trigger event occurs (for example, a particular time or day).

A remote user or system may dial into PortServer II using a terminal emulator such as ProComm Plus. When traffic has finished, the connection may be hung up to reduce telephone costs. Alternatively, the connection may remain continuously established. **Be aware that additional connect time costs could be incurred if the latter method is used**.

PPP

PPP (Point to Point Protocol) encapsulates network level IP protocol data on transmissions between point-to-point links. PortServer II supports auto-detection of CHAP (Challenge Handshake Authentication Protocol) and PAP (Password Authentication Protocol) on serial ports that are configured for PPP. CHAP authentication is attempted first, then PAP authentication, if CHAP fails.

If the user's software accepts the PAP authentication request, PortServer II requires a PAP ID and password to establish the connection.

If the user's software accepts CHAP authentication, PortServer II sends a CHAP packet that includes an ID and a random number. The user's software must respond with the same ID and the user name in encrypted form. When PortServer II receives the response, it verifies the encrypted data. The connection is only established when both have verified the encrypted data. This method of authentication ensures that a transmitted password cannot be stolen by another user to gain unauthorized access. To ensure security is maintained, CHAP challenges the connection at random intervals.

SLIP and CSLIP

SLIP (Serial Line Interface Protocol) or CSLIP (Compressed Serial Line Interface Protocol) can be used with older systems that do not support PPP.

Filters and scripts

Filters can be used with PPP, SLIP, and CSLIP connections to bring up or maintain a connection, to pass or block packets, or to log packets. For example, you can write a filter that will block all incoming packets, except those to a certain IP address, and so build a "firewall". Other filters block selected packets, for example, to block out broadcast messages. Filters are typically used to control access to specific hosts, networks or services, and thus increase security on your system. If PortServer II detects packets that are blocked by a filter, it drops the packets.

Scripts are used to establish outgoing connections by dialing modems and logging in to remote systems. They can also be used to initialize and test modems. A typical script initializes the local modem, then a second script sends a text string to log in to the remote location. It then waits for a predefined reply string to be received from the remote site before it establishes the connection.

RADIUS

PortServer II supports the RADIUS (Remote Authentication Dial In Service) standard for authentication of dial-up users. A RADIUS server receives user connection requests, authenticates the user against a password file or database, and returns sufficient configuration information to allow the service to be initiated. If RADIUS is used on a connection, the RADIUS server is responsible for all authentication, and PortServer II only routes messages and responses.

Frame Relay

Frame Relay is a switched digital service available from many providers that permits several *virtual connections* to share a single *physical connection*. Each virtual connection is referred to as a PVC (Permanent Virtual Connection) and corresponds to a link between two points on the network; part of that link (the physical connection) may be shared by other users. The maximum speed of the link is determined by the bandwidth of the physical connection.

Note: Frame Relay also allows the use of SVCs (Switched Virtual Circuits). SVCs are not supported by PortServer II.

Each PVC is identified by a unique number called a DLCI (Data Link Channel Identifier), which is used in Frame Relay "cloud" packets for routing.

Frame Relay allows network devices to exchange status information using the LMI (Local Management Interface). DLCI lists may also be "learned" from the LMI.

SNMP

SNMP (Simple Network Management Protocol) allows a server or PC (the SNMP Manager) to gather data and event records from devices connected to the network. The SNMP Manager may also set thresholds and alarms on network devices. PortServer II includes an SNMP agent that allows it to interact with the SNMP Manager. PortServer II maintains several MIBs (Management Information Bases) which are databases of the functions and events that it tracks for the SNMP Manager.

RealPort protocol

RealPort™ is a protocol developed by Digi International that permits the ports on the PortServer II to be controlled by a host server. RealPort provides "real" TTY access, and allows the terminal users seamless access to server-resident applications and data. The server can change port parameters such as baud rate and flow control. It is also possible for more than one server to control ports on the PortServer II; for example, one server may use the odd numbered ports, while another server may use the even numbered ports. RealPort can also provide access for printers and modems. It is also possible to share the same ports with different servers running RealPort drivers.

Note: The appropriate Digi device driver software must be installed on each server to implement RealPort.

Authentication and security

In addition to the authentication features provided for dial-up connections by PAP/CHAP and RADIUS described previously, PortServer II provides two levels of access security, based on passwords. Supervisory access is available using the root login and permits the configurations and functions of the PortServer II to be changed. Normal user logins provide transparent access to applications, and do not allow access to PortServer II configuration data. If maintaining security is not critical, users may be configured to permit logging in without a password or for automatic login.

Ethernet connection and dynamic IP addressing

PortServer II requires an IP address to allow it to communicate on the Ethernet. This can be assigned manually or, if a RARP (Reverse Address Resolution Protocol) server is connected to the network, PortServer II attempts to acquire its IP address automatically.

Each device that connects to a PortServer II can have its own IP address and, if it does, PortServer II will route TCP/IP packets to and from the device transparently. This feature permits PortServer II to operate as a router between all IP addressable hosts that are connected to its serial ports or to the Ethernet. PortServer II supports RIP (Routing Information Protocol), allowing it to inform other routers of routes, and to learn new routes from other routers.

You can configure PortServer II to use a pool of IP addresses for the devices that connects to it. This approach avoids the need to configure an address for each individual device or user. When a device or user requests a connection, PortServer II assigns the next available IP address from the pool. Consequently, a device may have a different IP address for each session.

Statistics and logging

PortServer II maintains a log of user activities, including login requests, time of login, and services used. The content of the log is definable by the system administrator. This log may be viewed by anyone who accesses PortServer II as `root`.

Statistics relating to PortServer II operation are also available from an SNMP Manager.

Remote configuration

To allow PortServer II to be easily updated with new features, you can download new software from a server using TFTP or Bootp commands.

You can also configure PortServer II's system-specific parameters from a remote terminal or host. PortServer II's internal configuration can be retrieved by TFTP commands, updated with a text editor, then reloaded into PortServer II.

Expanding PortServer II with additional ports

The standard PortServer II provides 16 serial ports. If you require additional ports at any time, you can connect external modules from the Digi International PORTS range to PortServer II's EBI (External Bus Interface) connector, as described below.

Note: Data speed through the serial ports under sustained load may be less than that specified if you connect external modules.

Description of PortServer II Hardware

The Digi PortServer II intelligent terminal server allows you to connect up to 64 RS-232 asynchronous serial devices (such as terminals and printers) to an Ethernet network. Both Twisted Pair (10BaseT) and Thinnet (10Base2) cabling connectors are provided on the side of the PortServer II box. If your network uses Thicknet cabling, use a transceiver from Digi International.

The PortServer II hardware features a 20 MHz 32-bit IDT 3051 RISC microprocessor and a 82596 32-bit network interface controller. It includes 2 Mbytes of flash memory, 2 Mbytes of RAM, and four Cirrus quad UARTs with Direct Memory Access (DMA). Self-tests on power-up help ensure reliability. Front-panel LEDs and controls can be used for diagnostic testing and performance checks, monitoring either RS-232 or Ethernet activity.

PortServer II features Digi International's External Bus Interface (EBI) connector, which can be used to "daisy-chain" up to three external Digi PORTS modules (PORTS/16em, PORTS/8em and PORTS/8emp) to add extra ports.

After the power-up self-tests and loading of the operating software (either from firmware or downloaded from the network), PortServer II sends login messages to all terminals connected to it. (Ports set up as printers or modems do not receive login messages; the factory default sets all ports as terminals.)

When they receive the login prompt, users can log into the PortServer II. Depending upon their privilege level (as defined by the system administrator), they can issue commands to the PortServer II to change parameters or connect to one of the network systems. The PortServer II software allows the system administrator to set up password-protected accounts with various privilege levels to restrict users' access to systems on the network.

Typical Applications

Terminal server configuration - local devices and RealPort

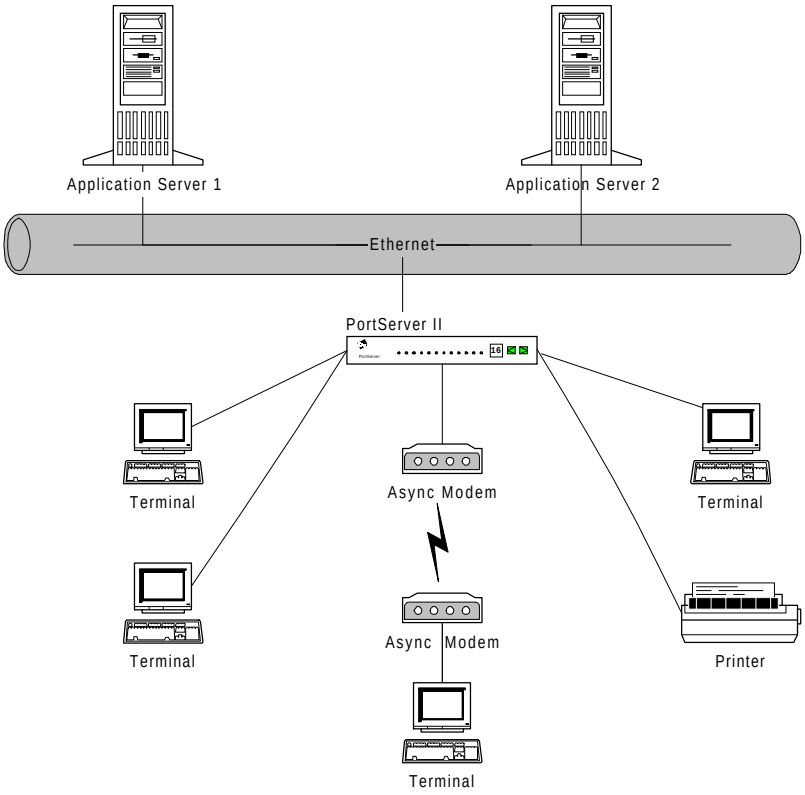

Figure 2 PortServer II - Local Devices and RealPort

In this configuration, several "dumb" terminals are connected to the Ethernet by a PortServer II. This example shows that it is possible to connect different devices to the same PortServer II, including modems and printers. Features to note include:

- Each device can communicate with several hosts or application servers.
- The application servers may be running Digi International's RealPort software (see *Chapter 11, Configuring RealPort Connections*).
- The application servers may be running different operating systems.
- PortServer II and application servers are connected by Ethernet (see *Chapter 4*).
- Terminals, modem, and printer are connected to serial ports on PortServer II.

Terminal server configuration - remote devices and RealPort

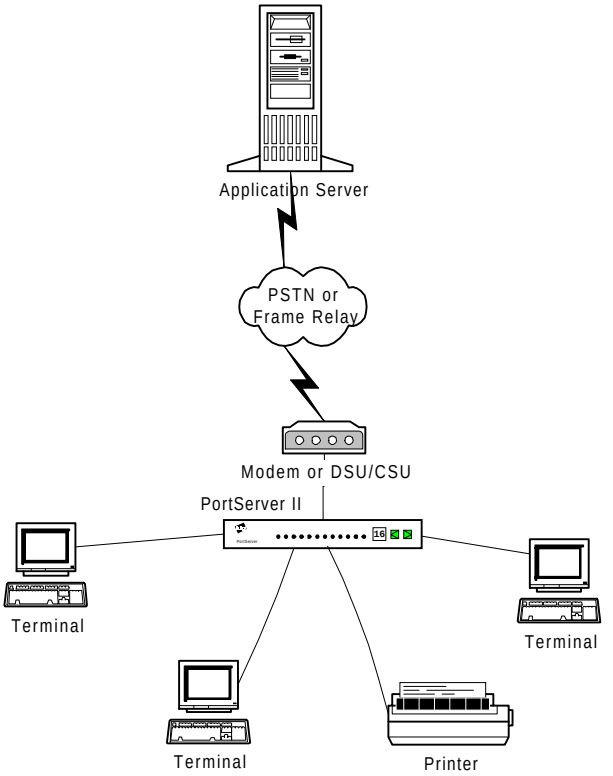

Figure 3 PortServer II - Remote Devices and RealPort

In this configuration, several “dumb” terminals are connected to a corporate application server by means of a PortServer II and the PSTN (Public Service Telephone Network) or Frame Relay. This example shows that it is also possible to connect a variety of different devices to the PortServer II in a “dial-up” configuration, including printers, modems, and other peripherals that might be found in a typical branch or satellite office. Features to note include:

- The corporate application server may be running RealPort software (see *Chapter 11, Configuring RealPort Connections*).
- PortServer II’s Ethernet port is configured with an IP address (see *Chapter 4*).
- PortServer II and the application server could be connected by a “dial-up” line (not shown, see *Chapter 9, Configuring Modem Connections*).
- Terminals, modem, and printer are connected to serial ports on PortServer II.

Terminal server configuration - multiple remote devices at several locations

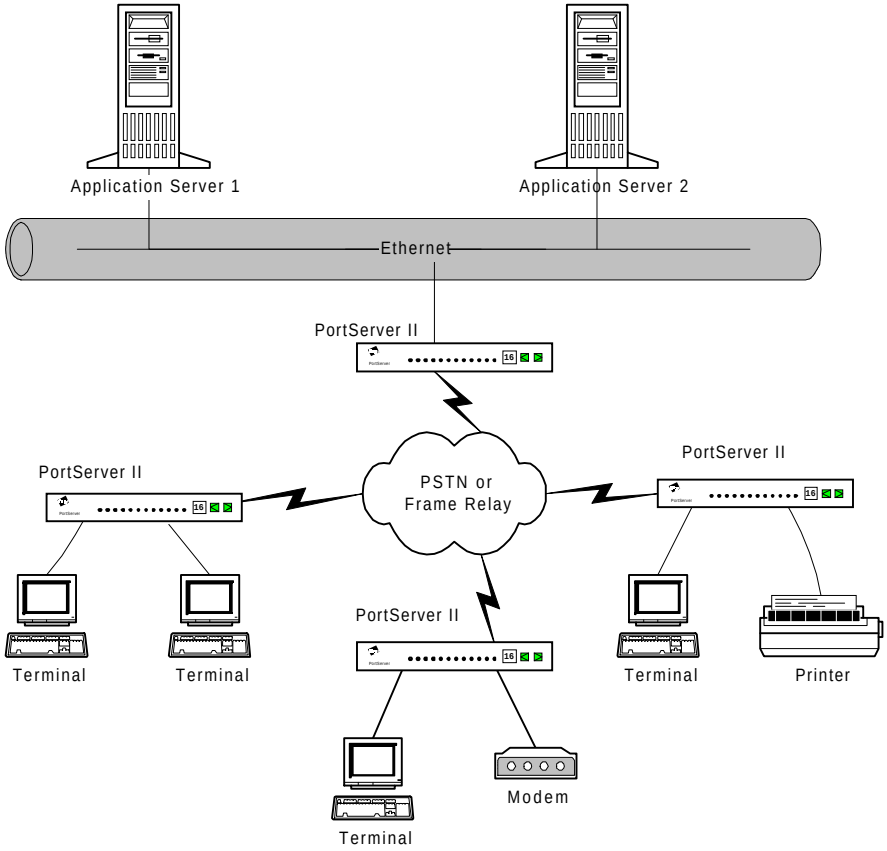

Figure 4 PortServer II with multiple remote devices at several locations

In this configuration, several branch or satellite offices are connected to a corporate application server by means of PortServer IIs and the PSTN (Public Service Telephone Network) or Frame Relay. Each location includes printers, modems, and other peripherals. Features to note include:

- The corporate application server is connected to the main corporate LAN.
- The corporate LAN is connected to the PSTN by a PortServer II (see *Chapter 9*).
- The corporate application server may be running RealPort software (see *Chapter 11*).

- Each remote location is connected to the PSTN by a PortServer II (see *Chapter 9*).
- Each remote PortServer II can be configured to call the corporate server, or the corporate server can call each remote location. Alternatively, the configuration could allow either location to initiate a connection.

  Note: If the remote offices are equipped with "smart" terminals or PCs, a PortServer II at a remote office can call another remote office.

- The Ethernet ports on the remote PortServer IIs are not used in this configuration.

  Note: The Ethernet port may require an IP address assigned to it to provide correct functionality. The address may be re-used for the WAN connections (this is sometimes called "unnumbered" IP link operation).

- The remote peripherals are all connected directly to serial ports on the PortServer IIs.

TIP — Leased Lines

If you are using PortServer II with standard leased lines and synchronous CSUs, Frame Relay often gives the best results. If you have asynchronous CSUs, PPP may be the best option. Note that the DCD signal is ignored for asynchronous connections.

Communications server configuration - remote dial-in users at several locations

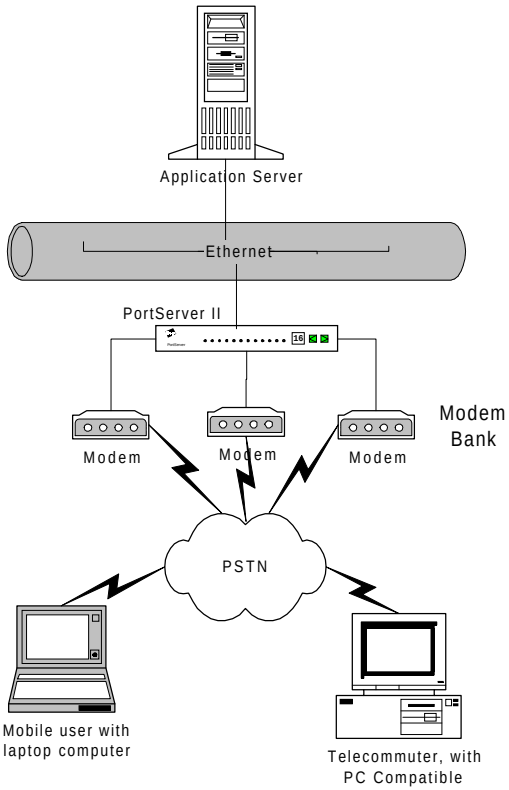

Figure 5 PortServer II with remote dial-in users

In this configuration, several sales personnel and telecommuters, are connected to an application server by means of a PortServer II and the PSTN. Features to note include:

- Each dial-in user has access to the corporate server.
- The corporate server is connected to PortServer II by Ethernet (see *Chapter 4*).
- RealPort software is not required on the corporate server.
- The remote users connect on dial-in TCP/IP connections, using PPP, SLIP, or CSLIP. For example, they may use Windows 95's HyperTerminal program to make PPP connections (see *Chapter 9*).
- PortServer II can make a primary authentication of users, or the server may be a RADIUS server for ease of configuration (see *Chapter 15, Configuring RADIUS*).

Communications server configuration - dial-out access to the Internet

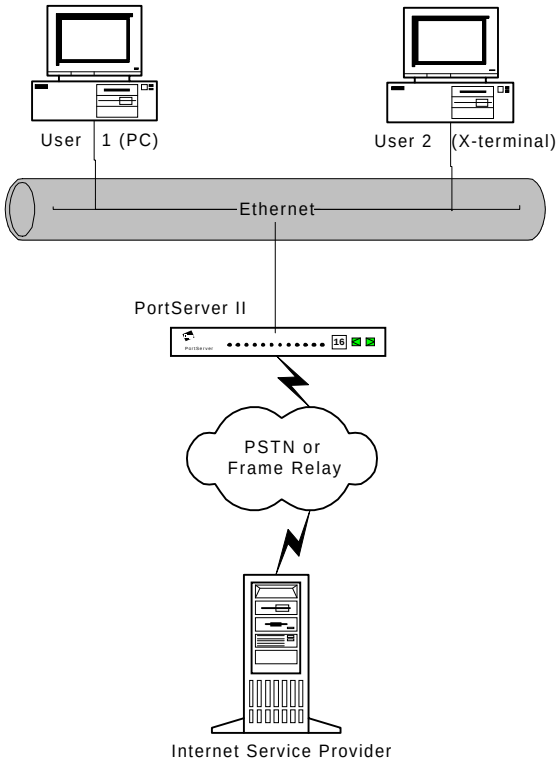

Figure 6 PortServer II - Dial-Out Access to the Internet

In this configuration, several satellite or remote users are provided with access to an ISP (Internet Service Provider) by a PortServer II and the PSTN (Public Service Telephone Network) or Frame Relay (see *Chapter 14*). This permits the user full access to all Internet options, including e-mail and World Wide Web (WWW) if they have the appropriate software installed on their terminals. Features to note include:

- PortServer II is configured to provide a dial-out PPP link to the ISP.
- The link may be available at all times, only at certain times, or only on demand, depending on traffic and billing considerations (see *Chapter 8*).
- PortServer II is connected to the local user's terminals by Ethernet.
- It is not necessary for local users to log into the Internet service individually. PortServer II will log in for all configured users (see *Chapter 9*).

Specifications

Network compatibility

- Ethernet IEEE 802.3 standard
- One 10BaseT twisted-pair Ethernet port with an RJ-45 8-pin connector
- One 10Base2 Ethernet port with a BNC coaxial connector

Ports

- 16 EIA RS-232 synchronous/asynchronous serial ports, each with an RJ-45 connector
- 115.2Kbps is available on all 16 ports. Connection of an expansion module may reduce the available bandwidth.
- Support for TCP, IP, ICMP, UDP, telnet, reverse telnet, rlogin, and ARP.
- One EBI (External Bus Interface) connector, allowing the connection of external modules that provide a total of up to 64 ports

Power Requirements

External 43W universal 50/60Hz power supply included. Internal supplies:

- +5 volts ± 5% 1250 mA typical
- +12 volts ± 5% 250 mA typical
- –12 volts ± 5% 50 mA

Environment requirements

- Ambient temperature 10° C (50° F) to 55° C (130° F)
- Relative humidity 5% to 90%
- Air movement 30 CFM Forced
- Altitude 0 to 3,660 meters (0 to 12,000 feet)

Dimensions

- Length 12 inches (305 mm)
- Width 7 inches (224 mm)
- Height 2.4 inches (57 mm)
- Weight 2.25 lbs (1.0 kg)

Free-standing and rack-mount versions are available.

Limits

PortServer II includes the following limits:

Users	Maximum 64 internal users, regardless of whether expansion modules are connected. Use of an external RADIUS server allows additional users, up to the limit of the RADIUS server.
WANs	Maximum 128. 64 outgoing connections, waiting for traffic, and 64 established via RADIUS.
Links	Maximum 64 of each.
IP Routes	Maximum 50 static routes.
Stanzas	Maximum 24 in each script.
Menus	User navigation menus can have a maximum of 20 lines, including two title lines.

Chapter 2

Operation

When should I read this Chapter?

Read this chapter if you are unfamiliar with PortServer II, and want to learn about its controls, indicators, and connectors.

In this Chapter

This chapter describes the controls, indicators, and connectors on PortServer II, and describes how to interpret the displayed information. It includes the following topics:

PortServer II Front Panel

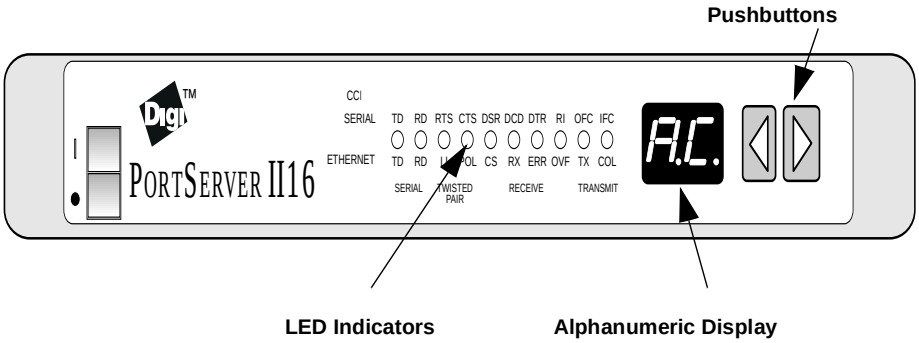

Figure 7 PortServer II Front Panel

LED indicators

These show the current status of a serial port or the Ethernet connection. Details of how to interpret the information displayed are given later in this Chapter. The LEDs can also show diagnostic information, as described in *Chapter 17, Troubleshooting*.

Alphanumeric display

If the alphanumeric display shows a number, the information shown on the LED indicators relates to the serial port of that number. For example, if it shows **16**, the information on the LED indicators is for Port 16.

If the alphanumeric display shows **EA**, the information shown on the LED indicators relates to Ethernet Activity.

If the alphanumeric display shows **PU**, the LED indicators show Processor Utilization in a bar graph-like presentation. The more LEDs that are lit, the greater the CPU activity.

If the alphanumeric display shows **AC**, PortServer II has powering up and is now in normal operating mode.

Other indications on the display identify possible fault conditions. Refer to *Chapter 17, Troubleshooting* for information on fault codes.

Pushbuttons

There are two pushbuttons, one marked with a left arrow and one with a right arrow. You can use these to:

- Select a serial port to monitor. If you press the left arrow, the number on the alphanumeric display decrements and information for that serial port is shown on the LED indicators. Similarly, if you press the right arrow, the number on the alphanumeric display increments and information for that serial port is shown on the LED indicators.
- Select diagnostic tests, as described in *Chapter 17, Troubleshooting*.
- Reset PortServer II to factory default settings, as described in *Chapter 17, Troubleshooting*.

Interpreting the LED Indicators

The LED indicators are used to display two different types of status information. If the alphanumeric display shows a port number, the LED indicators display serial port status information and the labels *above* the LEDs apply. If the alphanumeric display shows **EA**, the LED indicators display Ethernet activity and the labels *below* the LEDs apply. If the alphanumeric display shows **AC** or **PU**, the LED indicators do not display data.

Serial port status

You can display the status of a serial port by pressing either of the pushbuttons until the port number appears on the alphanumeric display. The LED indicators then provide RS-232 line status information (in a similar way to a breakout box) and flow control information, as follows:

Signal	CCITT	Description
TD	103	Transmitted Data
RD	104	Received Data
RTS	105	Request to Send
CTS	106	Clear to Send
DSR	107	Data Set Ready
DCD	109	Data Carrier Detect
DTR	108	Data Terminal Ready
RI	125	Ring Indicator
OFC	-	Output is Flow Controlled
IFC	-	Input is Flow Controlled

Ethernet activity

You can display the status of the Ethernet connection by pressing either of the push-buttons until **EA** appears on the alphanumeric display. The LED indicators then provide Ethernet status information, as follows:

Signal	Function	Description
TD	Serial data	Serial data has been transmitted on any of the serial ports
RD	Serial data	Serial data has been received on any of the serial ports
LG	Twisted pair	Line Good. Indicates a good connection to the Ethernet hub box.
POL	Twisted pair	Polarity is backwards. Indicates that the twisted pair wiring has been installed with the wires transposed. PortServer II will still operate correctly.
CS	Receive	Carrier Sense. Blinks when PortServer II senses the Ethernet carrier signal.
RX	Receive	Blinks when PortServer II detects a packet destined for itself.

Signal	Function	Description
ERR	Receive	Network error on packet received, for example, CRC, Frame, or FIFO overrun.
OVF	Receive	Overflow. Lights when packets arrive faster than PortServer II can process them.
TX	Transmit	Lights when PortServer II is transmitting a packet.
COL	Transmit	Lights when PortServer II detects a collision on the network. This may be a collision between any packets, not necessarily packets destined for or originated by PortServer II devices.

PortServer II Side and Rear Panels

The connectors for the Ethernet cable, the power on/off switch, and the D.C. power supply are located on the left-hand side panel of PortServer II (viewed from the front), while the serial port connectors are on the rear panel. The locations of connectors are shown below:

Side panel

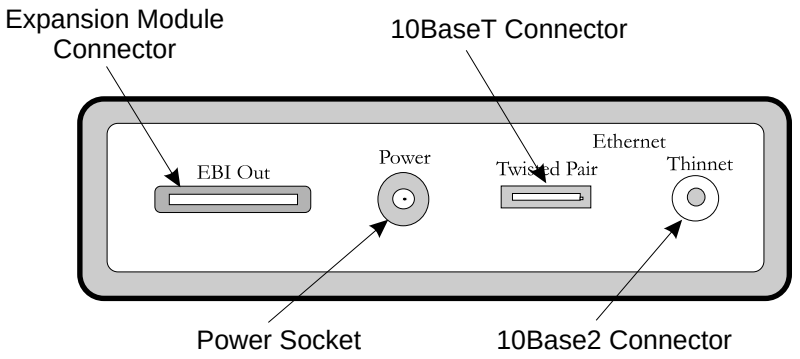

Figure 8 PortServer II Side Panel

Power on/off switch and socket

The D.C. power connector from the power unit provided with PortServer II should be inserted in this socket. Use the switch to turn the PortServer II on or off.

Caution!

This switch does not disconnect power from the PortServer II power unit and there may be 110V/220V power present on the power unit when this switch is set to 0 (off). Take care if you are doing installation or service work.

EBI Out connector

This is provided for the connection of a compatible expansion module from the Digi International PORTS range. Refer to the expansion module documentation for more details.

Thinnet connector, twisted pair connector

These permit a single connection to a compatible Ethernet hub. The internal circuitry senses whether you have made a 10BaseT connection or a 10Base2 (coaxial) connection, and functions accordingly.

Rear panel

The rear panel provides 16 identical RS-232 compatible serial connectors. Port 1 is at the left, viewed from the rear.

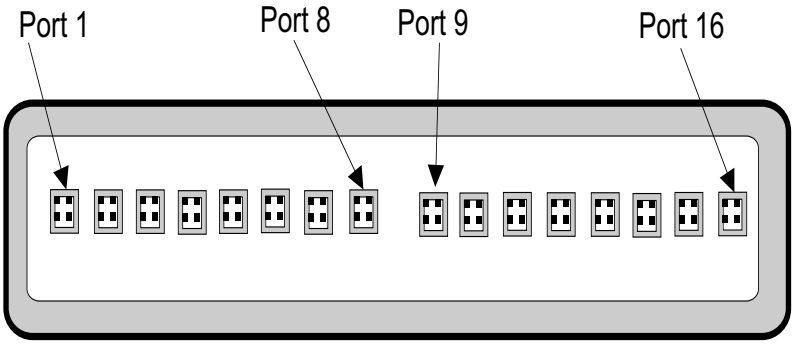

Figure 9 PortServer II Rear Panel

Chapter 3

Installation

When should I read this Chapter?

Read this chapter before you install and connect a PortServer II.

In this Chapter

This chapter describes how to install PortServer II, including site planning procedures and information on basic safety. It also includes descriptions of the cables that you will require. This chapter is divided into the following sections:

Before you Begin

Before you can install PortServer II, you must do the following:

- Select a location.
- Determine cabling requirements and obtain proper cables.
- Have available a terminal (or PC with terminal emulation software) to configure PortServer II.

Surveying the Installation Site

1. Identify the best location to place PortServer II. You may want to review the *Site Environment* requirements and the information given on *Interference Limitation* below.

2. Locate the relevant connection points to the Ethernet and each PC, terminal, modem, and printer to which you will connect PortServer II. Measure the distance from the PortServer II location to each connection point.

3. Decide the length and type of cable required to make each connection. Remember that each cable should be longer than the distance you measured to allow for tidy routing.

4. Check that the cable lengths are within the maximum distance limitations for that type of cable, using the information in *Recommended Maximum Distance Limitations* below. If any of the maximum distances are exceeded, you may have to relocate PortServer II.

5. If the cable lengths do not exceed the maximum specified distance, obtain the required cables.

Interference limitation

Reliable operation of PortServer II can be compromised by electromagnetic interference generated in the connecting cables. Typically, this interference may be derived from three sources:

- Plant and equipment containing electric motors, such as air conditioners or heating pumps, or their connecting cables
- High power radio transmitters
- Lightning strikes

You can minimize the risk of interference by using coaxial or twisted-pair cables with grounding connectors at each end and junction point. You should not exceed the maximum distances specified under *Recommended maximum distance limitations* on Page 30.

If any of your cables pass between buildings or on the outside of buildings, you should give special consideration to the possibility of lightning strikes. The electromagnetic pulse carried by lightning and similar high-energy phenomena can induce enough energy in cables to destroy PortServer II's circuitry. If your installation site is subject to lightning strikes, you may want to consider lightning suppression and shielding.

Recommended maximum distance limitations

The maximum recommended length of each type of cable connection you may use with PortServer II are shown below.

Note: Although PortServer II may operate adequately if you exceed a given limitation, performance may be degraded.

Ethernet

10BaseT: Maximum distance of 330 ft (100 M) at a transmission rate of 10 megabits/second (10 Mbps).

10Base2: Maximum distance of 610 ft (185 M) at a transmission rate of 10 megabits/second (10 Mbps).

EIA RS-232 serial ports

Maximum distance depends on the data rate and the type of cable:

Table 1 RS232 Serial Port Specifications

Data Rate (Baud)	Distance (ft)	Distance (M)
2400	200	60
4800	100	30
9600	50	15
19200	25	7.5
38400	12.5	3.8
57600	6.3	1.9
115200	3.2	0.9

V.24 cable

If you are connecting to a Frame Relay service, you require a fully V.24-compatible cable. Distance limitations are the same as for serial cables.

Inspecting PortServer II

Do not unpack PortServer II until you are ready to install it. When you unpack the shipping box, check that you have the following items:

- One PortServer II unit
- Power supply unit with integral cords
- One RJ-45 to DB-25 cable leg converter. This is intended for connecting the terminal or PC on which you will configure PortServer II. You will require a null-modem cable or adapter to provide full terminal functions using this cable.

  Note: If you require additional cables, contact your Digi International dealer or distributor.

- This *User's Guide* and associated *Command Reference Guide*.
- A warranty card and associated paperwork.

Examine PortServer II and power supply unit for signs of physical damage or breakage.

If anything is missing or appears damaged, or if you encounter problems when installing or configuring PortServer II, contact your dealer or distributor immediately.

Tools and equipment required

No special tools or equipment are required to install PortServer II.

Site Preparation

Site environment

The location you select for your PortServer II is extremely important for its proper operation. If it is placed too close to other equipment or in a hostile environment, you may cause equipment failures or errors that you could otherwise avoid.

Before you begin installation, identify the best location using the following guidelines:

- PortServer II generates heat and consequently requires adequate circulation to maintain its specified operating temperatures. Try to allow at least 12 ft (0.3 M) of ventilation space above and on all sides. Never cover or obstruct the ventilation slots provided on the unit.
- Always follow the ESD prevention procedures given later in this chapter when you work on PortServer II. Damage from static discharge can cause immediate or intermittent failure.
- Do not position PortServer II close to electrical equipment such as electric motors or air conditioners. Interference from electrical equipment may cause intermittent failures.
- Do not install PortServer II in an area where condensation, water, or other liquids may be present. This may cause a safety hazard and cause failure of the equipment.
- Ensure that the cover is secure and that all cables are firmly in place on completion of installation. This will reduce safety hazards and eliminate the chance of failures due to cable disconnection.

Safe installation practices

Read the installation instructions below and the associated warnings completely before beginning.

Warning!

The power supply provided with PortServer II is a sealed unit and contains no user-serviceable parts or adjustments. Do not attempt to open or otherwise tamper with the power supply.

- Locate the power off switch or main circuit breaker for the room in which you are working. If an electrical accident occurs, turn OFF the power immediately.
- Operate PortServer II only from the external power source complying with the requirements indicated in the *Specifications* in *Chapter 1*. If you are not sure of the type of power source, contact your dealer or power company.
- PortServer II's power supply is provided with a 3-wire plug, which includes a ground connection as a safety feature. If you are unable to insert the plug into your outlet, have an electrician replace the obsolete outlet. DO NOT attempt to defeat the safety feature of the plug.
- If you need to use a power extension cord, make sure the total ampere rating of all equipment plugged into the extension cord does not exceed the extension cord ampere rating. Also, make sure the total ampere rating of all equipment plugged into the wall outlets does not exceed the capacity of the outlet.
- If PortServer II exhibits unexpected behavior at any time (for example, smokes or becomes excessively hot), disconnect it from the power source immediately and obtain service assistance.
- If PortServer II is exposed to moisture or condensation, disconnect it from the power source immediately and obtain service assistance.
- Look carefully for potential hazards in your work area, such as damp floors, ungrounded power extension cables, and missing ground connections.

Installing and Connecting PortServer II

General procedure

When you are ready to complete installation, follow this procedure:

1. Place PortServer II in the required location.
2. Locate and identify each connecting cable.
3. Connect each signal cable to PortServer II and to the required destination (for example, terminal, PC, modem, printer, or Ethernet). Detailed procedures are given in *Connecting PortServer II to serial devices* following.
4. Connect the power unit to PortServer II and to the power source. Do not switch on until instructed.

Connecting PortServer II to the Ethernet LAN

Connect your Ethernet network cable to the appropriate connector on the left-hand side of PortServer II. Refer to your network documentation for appropriate procedures and precautions.

If you are using a Thinnet (10Base2) cable, plug the cable into the BNC coaxial connector marked **THINNET**.

If you are using Twisted Pair (10BaseT) cable, plug the RJ-45 connector into the 10BaseT connector marked **TWISTED PAIR**.

Connecting PortServer II to serial devices

General

Each of PortServer II's serial ports is provided with an RJ-45 10-pin jack socket. You can connect a serial device to the port using a cable terminated with any of the following plugs:

- Eight pin RJ-45 plug
- Ten pin RJ-45 plug
- Four pin RJ-11 plug
- Six pin RJ-11 plug

The RJ-45 plugs are the same physical size, but the ten pin version has one additional wire at each end of the row of contacts. Thus pins 1 through 8 of the eight pin version correspond to pins 2 through 9 of the ten pin version.

Similarly, the RJ-11 plugs are the same size, but the six pin version has two extra pins at either end.

RJ-11 plugs are smaller than RJ-45 plugs, but are designed to fit into the center of an RJ-45 socket. The pins are assigned from the center and are always in the same location, regardless of the size of plug. For example, the pins of a six pin RJ-11 plug carry the same signals as the center six pins of an eight or ten pin RJ-45 connector.

Caution!

If you insert an RJ-11 plug into an RJ-45 connector, take care to align the pins correctly.

The wiring of the various plug types is shown on the following pages:

Ten pin RJ-45

The ten pin RJ-45 plug carries all eight of the RS-232 signals supported by PortServer II, together with two ground lines, SG (Signal Ground) and GND (Chassis Ground). It includes two modem control lines, RI (Ring Indicator) and DCD (Data Carrier Detected).

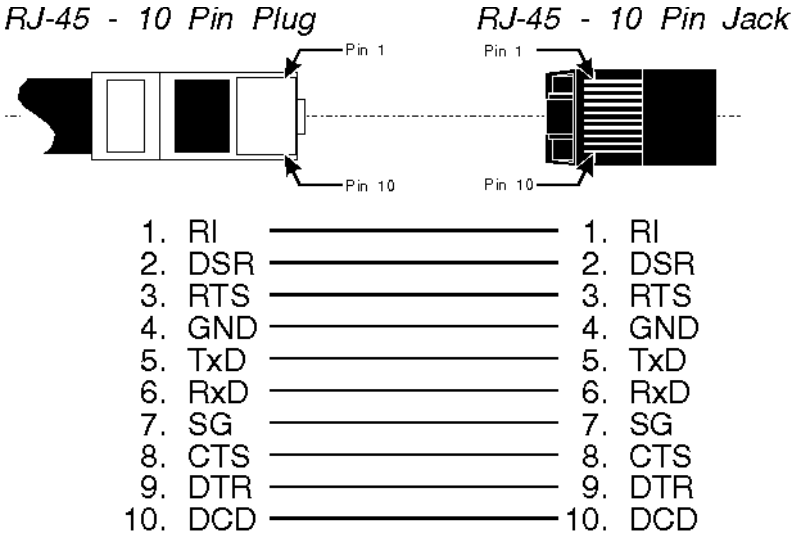

Figure 9 RJ-45 10 Pin Plug and Jack Wiring

Eight pin RJ-45

The eight pin RJ-45 plug carries the RS-232 signals supported by PortServer II, except for the two modem control lines, RI (Ring Indicator) and DCD (Data Carrier Detected). It also provides the two ground lines, SG (Signal Ground) and GND (Chassis Ground).

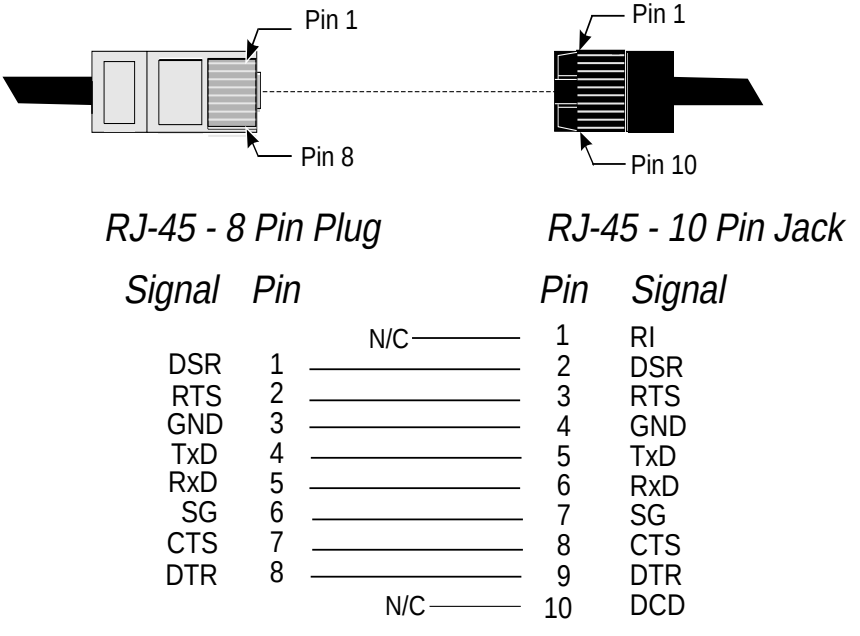

Figure 10 RJ-45 8 Pin Plug and Jack Wiring

Note: PortServer II software provides DCD in eight pin configurations. See *Chapter 9, Configuring Modem Connections*.

Six pin RJ-11

The six pin RJ-11 plug is suitable for connections that require limited control signals, for example modems or printers that employ hardware handshaking.

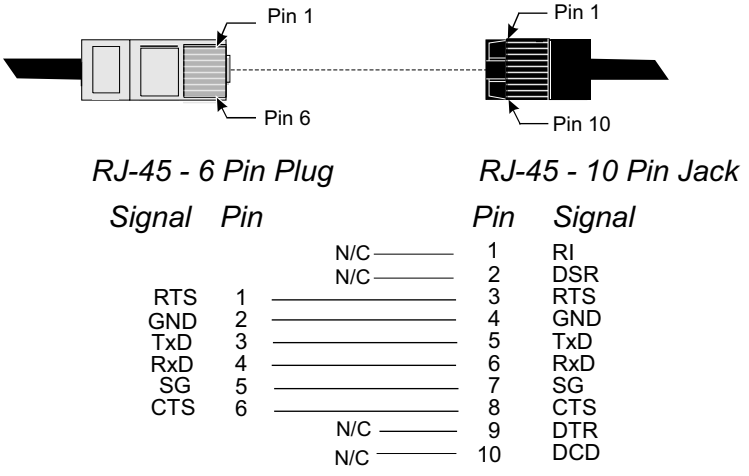

Figure 11 RJ-11 6 Pin Plug and Jack Wiring

Four pin RJ-11

The four pin RJ-11 plug is suitable for connections that require only data signals, for example modems or printers that employ software handshaking. No hardware handshaking is available with this configuration.

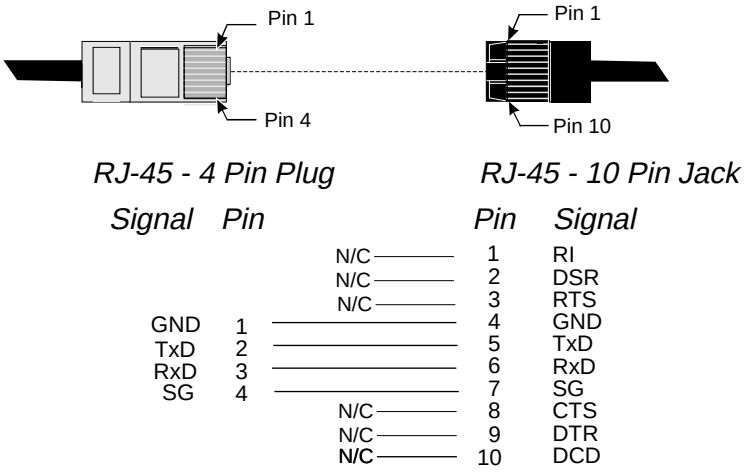

Figure 12 RJ-11 4 Pin Plug and Jack Wiring

Connecting the configuration terminal

Connect a PC or terminal to Port 1 on the rear of PortServer II to use during the configuration procedure described in *Chapter 4, Basic Configuration*. You can subsequently use this terminal for system management or other purposes. If you use a PC, you will require a terminal emulator program.

1. Obtain a suitable cable to connect your terminal to the serial port on PortServer II. The RJ-45 to DB-25 cable leg connector supplied with PortServer II was intended for this purpose. If you prefer, you can also construct or obtain your own cable.

   If you construct or obtain your own cable, you only need to connect the TxD (Transmitted Data), Received Data (RxD), and Signal Ground (SG) cables, as shown below.

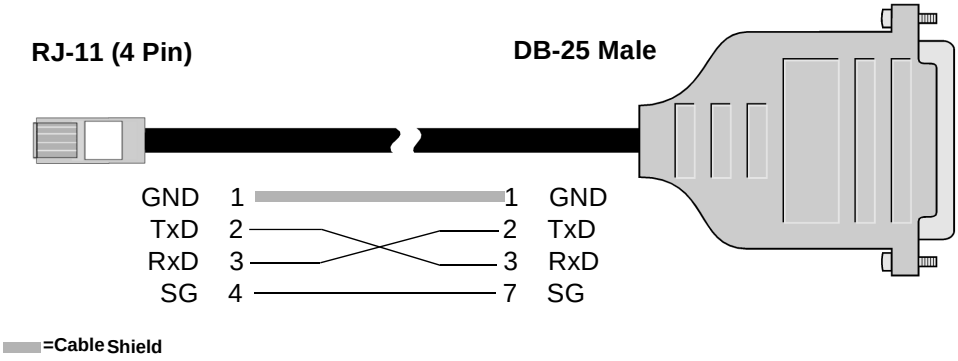

Figure 13 Simple 3 Wire cable

Note: Any cable you construct or obtain must be shielded to comply with FCC certification requirements. The shield must be connected to Chassis Ground (GND) at one end of the cable.

In addition to the configuration illustrated above, you can also use a 6-pin RJ-11 plug, or an 8 pin or 10 pin RJ-45 plug. (See Page 35).

2. Fit the DB-25 connector to the terminal. If your terminal does not have a DB-25 connector, you will also have to obtain an adapter.

3. Fit the RJ-11 or RJ-45 plug to PortServer II. If you use an RJ-11 plug, ensure that you center it on the jack.

Connecting to terminals and PCs

You can connect a terminal to any available serial port on PortServer II.

1. Obtain a suitable cable to connect the terminal to the serial port on PortServer II. The wires that must be available depend on whether the terminal requires hardware handshaking.

   If the terminal does not require hardware handshaking, you can use a simple 3-wire cable as shown below:

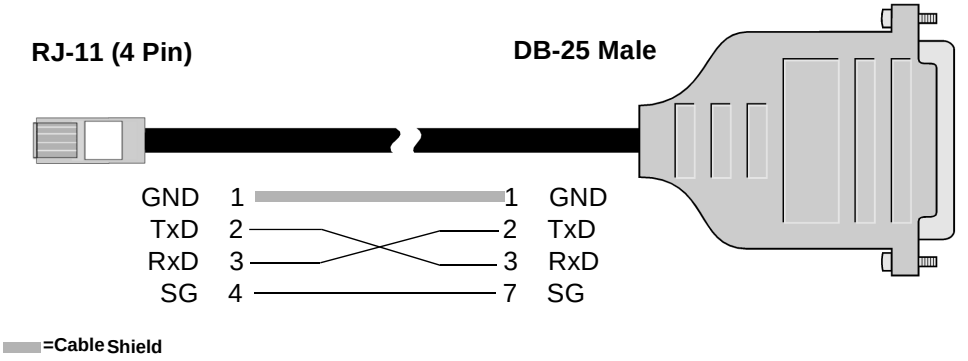

Figure 14 Simple 3 Wire cable

Note: Any cable you construct or obtain must be shielded to comply with FCC certification requirements. The shield must be connected to Chassis Ground (GND) at one end of the cable.

In addition to the configuration illustrated above, you can also use a 6-pin RJ-11 plug, or an 8 pin or 10 pin RJ-45 plug. (See Page 35).

If the terminal requires hardware handshaking, use the cable shown on the following page. This configuration assumes the terminal uses Data Terminal Ready (DTR) for handshaking; check your terminal or PC documentation for further information.

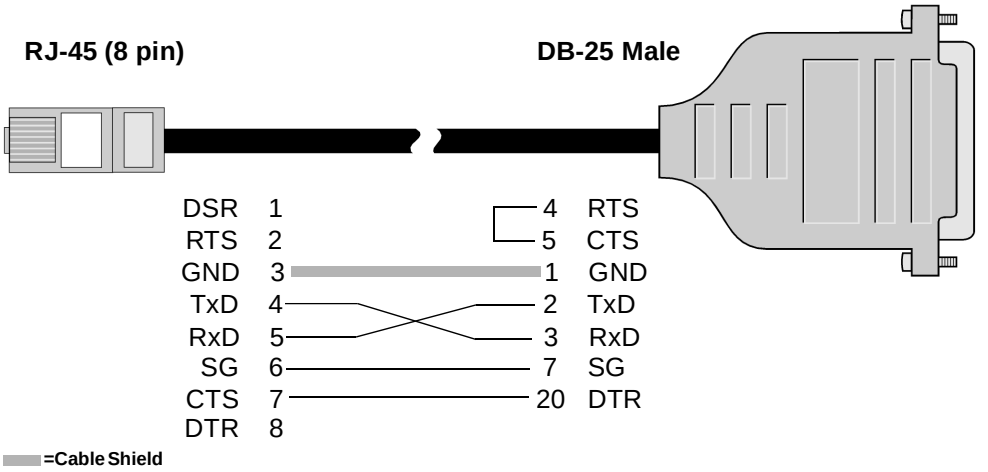

Figure 15 Terminal cable with DTR handshaking

2. Fit the DB-25 connector to the terminal. If your terminal does not have a DB-25 connector, you will also have to obtain an adapter.
3. Fit the RJ-11 or RJ-45 plug to PortServer II. If you use an RJ-11 plug, ensure that you center it on the jack.

Connecting to modems

You can connect a modem to any available serial port on PortServer II.

1. Obtain a suitable cable to connect the modem to the serial port on PortServer II. Most modems require a 10-pin cable; we suggest you obtain one of Digi International's RJ-45 to DB-25 straight-through cables:

Table 2 Available Cables

	DB-25 Male	DB-25 Female	DB-9 Male
24 Inch Cables	61020024	61030024	61070024
48 Inch Cables	61020048	61030048	N/A

This cable type is illustrated below:

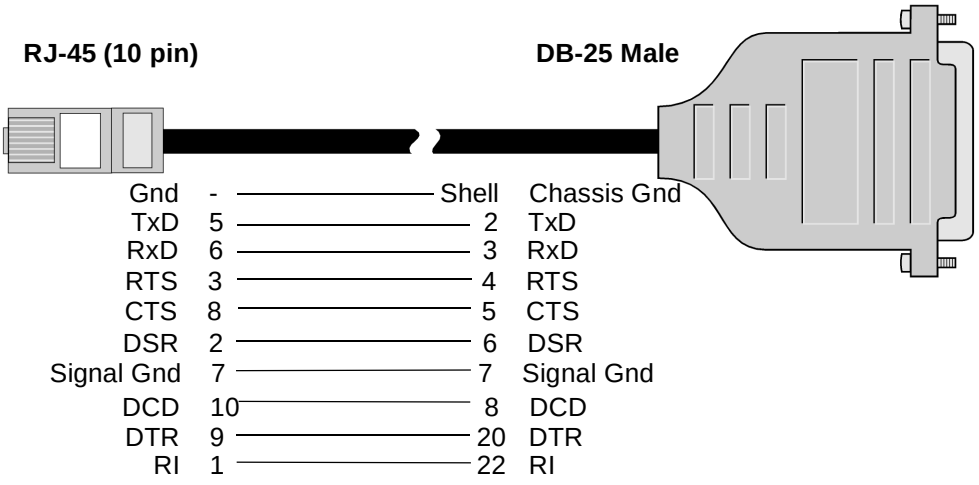

Figure 16 Full 10 Wire Modem cable

Note: Any cable you construct or obtain must be shielded to comply with FCC certification requirements. The shield must be connected to Chassis Ground (GND) at one end of the cable.

2. Fit the DB-25 connector to the modem. If your modem does not have a DB-25 connector, you will also have to obtain an adapter.

3. Fit the RJ-11 or RJ-45 plug to PortServer II. If you use an RJ-11 plug, ensure that you center it on the jack.

ALTPIN

PortServer II includes a feature called ALTPIN that allows you to use 8 pin RJ-45 to DB-25 adapters. ALTPIN swaps pins 2 and 10, making DCD available on pin 1 of an 8 pin RJ-45 connector. If you use ALTPIN, you must also configure the PortServer II port with the command `set flow altpin` (see the *Command Reference Guide* for details).

The cable wiring for ALTPIN is shown below:

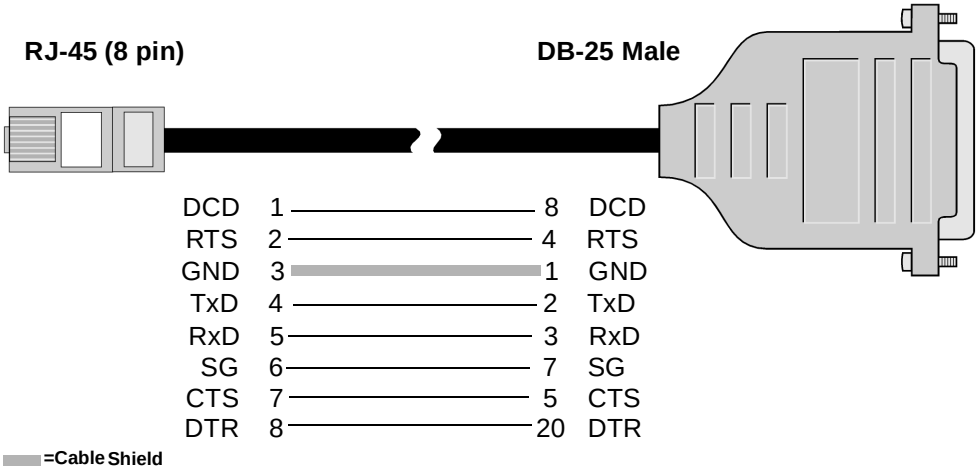

Figure 17 ALTPIN cable

Note: Digi International's software can use the DSR signal in modem control applications. If your modem has auto-answer capability, you do not require the Ring Indicator.

Connecting to printers

You can connect a printer to any available serial port on PortServer II.

1. Obtain a suitable cable to connect the printer to the serial port on PortServer II. The wires that must be available depend on whether the printer requires software (XON/XOFF) or hardware (DTR/DSR) handshaking.

   If the terminal employs software handshaking, you can use a simple 3-wire cable as shown below:

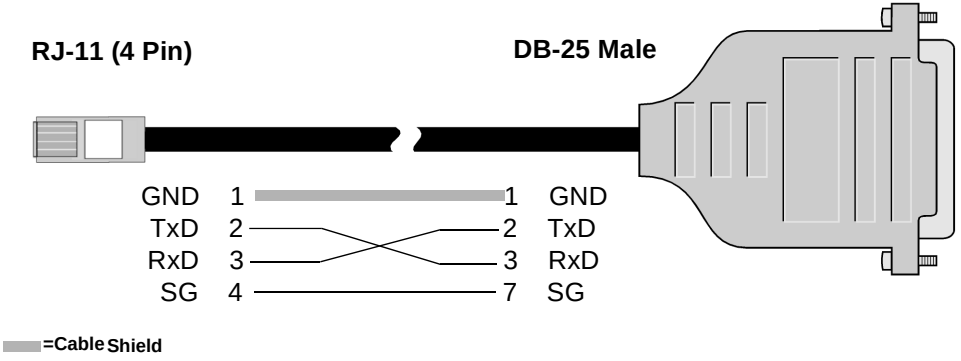

Figure 18 Printer cable with Software handshaking

Note: Some Okidata printers use a control signal called Supervisory Send Data (SSD) on pin 11, instead of DTR. For these printers, connect CTS on the RJ-45 connector to pin 11 of the DB-25 connector, instead of pin 20.

If the terminal requires hardware handshaking, use the cable shown on the following page.

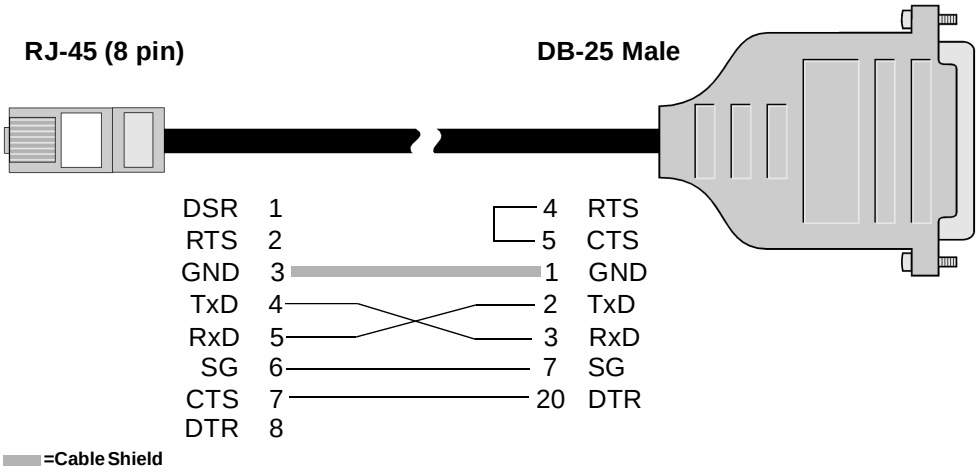

Figure 19 Printer cable with Hardware handshaking

Note: Any cable you construct or obtain must be shielded to comply with FCC certification requirements. The shield must be connected to Chassis Ground (GND) at one end of the cable.

In addition to the configurations illustrated above, you can also use a 6-pin RJ-11 plug, or an 8 pin or 10 pin RJ-45 plug for either application. (See Page 35).

Connecting to Frame Relay

If you plan to use PortServer II with a Frame Relay service, you must use a V24 cable similar to the one shown in Figure 20. One cable of this type is supplied with each PortServer II. An RJ-45 10-pin male connector is provided at the PortServer II end of the cable, and a DB-25 25-pin male connector is provided for the Frame Relay termination.

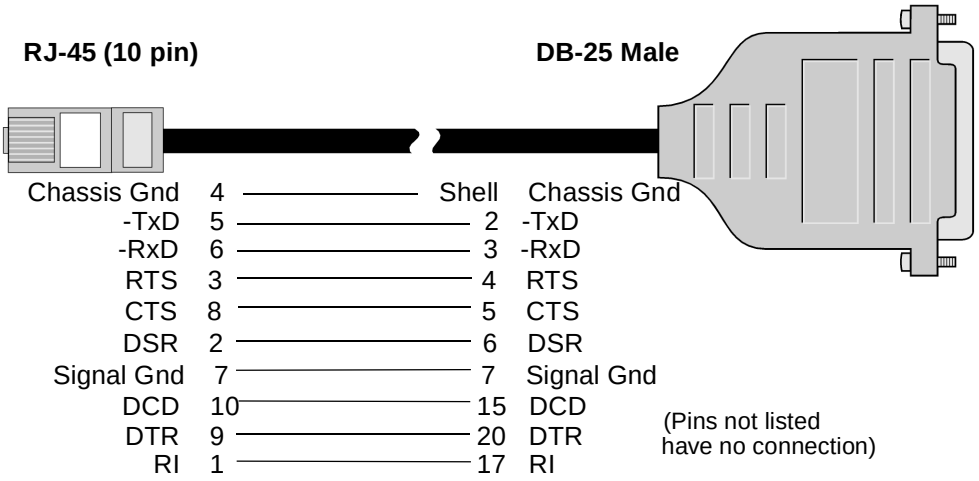

Figure 20 V24 Cable for Frame Relay

Wiring Ports for Specific Devices

The table below summarizes the characteristics of and wiring requirements for the device types available on PortServer II.

Device Type	DTR + RTS when idle	Spawn Login	Remote Connect allowed	Action when DCD drops
term	high	data	no	ignore
prn	low	never	yes	ignore
host	low	never	if DCD	hang up
hdial	low	DCD + data	no	hang up
hio	low	DCD + data	if DCD	hang up
min	high	DCD raises	no	hang up
mout	low	never	yes	hang up
mio	high	DCD raises	yes	hang up

Detailed explanations of these functions are given below.

Note: Device types (**dev=xxx**) are set with the `set ports` command, as described in the *Function Reference Guide*. Procedures are also given in later chapters of this *User's Guide*.

dev=host (Computer or other devices)

Use this device type for connecting another computer to a serial port. A typical use might be connecting a bulletin board system (BBS) to the Ethernet through the PortServer II. DCD and DTR must be cross-connected between the PortServer II and the BBS server.

When the server raises DTR to signal that it can receive calls, the PortServer II sees DCD go high. It then allows incoming connections from the Ethernet through the serial connection to the server. When an incoming connection is requested, PortServer II raises DTR, and the server sees DCD go high, indicating there is a call. (This sequence mimics a modem answering a dial-up call.)

The server can terminate the connection by dropping DTR, which PortServer II sees as DCD going low. PortServer II then terminates the incoming connection. If the incoming connection terminates itself, PortServer II lowers DTR. The server sees DCD drop, and knows the connection is terminated. DTR remains low for two seconds and input is flushed to assure a clean disconnection.

Termination control using DTR and DCD is the difference between **dev=host** and **dev=mout** (described later in this section). Both are used for outgoing connections. **host** requires that DCD is high before the connection is complete, while **mout** does not.

Wiring required: Full 10-wire Null Modem cable.

dev=hdial

Use this device type for connecting another computer to a serial port. DCD and DTR must be cross-connected between PortServer II and the host. When the host raises DTR, DCD goes high at PortServer II. When PortServer II receives data, a login is spawned.

Unlike **dev=min**, there is no two-second delay before the login is spawned. When PortServer II raises DTR, it brings DCD up at the host immediately.

When the user logs out or the host drops DTR, PortServer II lowers its DTR and flushes data for two seconds.

dev=hio

Use this device type for connecting another computer to a serial port when you require bidirectional incoming and outgoing connections. The device behaves like **dev=hdial** or **dev=host**.

When the host raises DTR (raising DCD on the PortServer II), an outgoing connection can be established. Alternately, if data is received, a login is spawned.

dev=term (Terminals)

Use this device type for “dumb” terminals. When the device is set for a terminal, the incoming port ignores DCD. All outgoing connect attempts are refused.

Wiring required: Only TD, RD, and GND connections are needed.

dev=prn (Printers)

PortServer II treats printers as outgoing devices that ignore DCD. Any incoming characters are also ignored, until an outgoing connection is established. When a connection is established DTR is raised, otherwise DTR is low. Unlike modem device types, there is no two second hangup.

This device type can also be used for general purpose outgoing connections

Wiring required: Generally, only TD, RD, and GND connections are needed.

dev=min (Modem In)

Use this device type for a modem that is used for incoming communications only. DTR and DCD are supported. In this mode, PortServer II ignores all input until DCD appears. Then it “flushes” all input for two seconds to discard messages like RING and CONNECT, before it spawns a login or attempts a connection.

Whenever DCD drops, PortServer II drops DTR. All connections terminate, and any user is logged out.

When a user logs out or (on an autoconnect port) the remote system drops the connection, DTR is dropped to hang up the modem. DTR remains low for two seconds to assure a clean modem disconnection, and then input is flushed to discard messages such as DISCONNECT.

Wiring required: Modem connections require that TD, RD, DCD, DTR and GND should always be wired in the cable. With multispeed, data compressing modems, you may also need to wire RTS and CTS for hardware flow control.

dev=mout (Modem Out)

Use this device type for a modem requiring outgoing communications only, with full DTR/DCD modem support.

Until an outgoing connection is made, DTR remains low, keeping the modem from answering. When an outgoing connection succeeds, DTR is raised.

The state of DCD is ignored until there is a high-low transition of DCD. This closes the connection, and drops DTR. DTR then remains low for two seconds to assure a clean modem disconnection, and then input is flushed to discard messages such as DISCONNECT.

Wiring required: Modem connections require that TD, RD, DCD, DTR and GND are always wired in the cable. With multispeed, data compressing modems, you may also need to wire RTS and CTS for hardware flow control.

dev=mio (Modem In & Out)

Use this device type for a bi-directional modem, requiring full DTR/DCD support.

When the modem is idle, PortServer II leaves DTR high so the modem will answer incoming calls.

If DCD goes high when the device is idle, PortServer II assumes an incoming connection has been requested and acts like a **dev=min** device until the device goes idle again.

If an outgoing connection succeeds when the device is idle, PortServer II switches into outgoing mode and acts as a **dev=mout** device until the connection is dropped. DTR remains low for two seconds to assure a clean modem disconnection, and then input is flushed to discard messages such as DISCONNECT messages.

Wiring required: Bidirectional modem connections require that TD, RD, DCD, DTR and GND are wired in the cable. With multispeed, data compressing modems, you may also need to wire RTS and CTS for hardware flow control.

Connecting to the External Bus Interface

PortServer II includes an External Bus Interface that permits connections to up to three external modules providing additional serial ports. Compatible external modules from the Digi International range of PORTS modules include:

- PORTS/16em. Provides sixteen additional serial ports.
- PORTS/8em. Provides eight additional serial ports.
- PORTS/8emp. Provides eight additional serial ports and one parallel port.

The connecting cable required is provided with the PORTS module and you should refer to the PORTS documentation for installation instructions.

An example configuration is shown below, with two PORTS/16em modules connected to a PortServer II. This arrangement provides 48 serial ports, 16 on PortServer II, and 16 on each PORTS module.

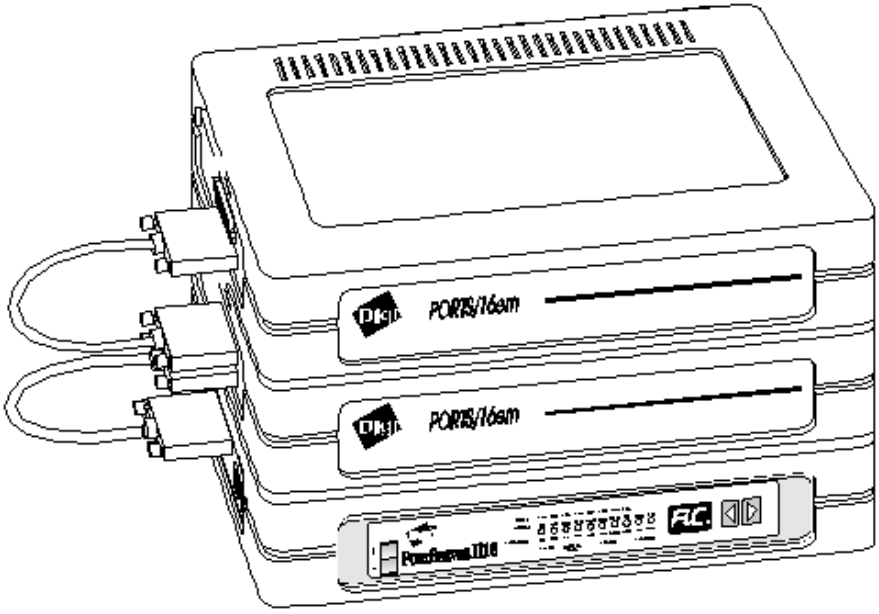

Figure 21 PortServer II and Expansion Modules

Warning!

Do not connect a PORTS module to PortServer II while power is switched on. This may cause severe electrical problems or damage to PortServer II and the PORTS module.

Note: You may experience performance degradation when using expansion modules under heavy data throughput. For example, if you have a total of 64 ports, you can drive some (but not all) ports at 115.2Kbps. The available WAN bandwidth on PortServer II is approximately 3.7Mbps; the maximum port speed for any given port is either 115.2Kbps, or whatever portion of 3.7Mbps is not currently in use by other ports that are simultaneously active.

Chapter 4

Basic Configuration

When should I read this Chapter?

Read this Chapter if you are going to configure a new PortServer II for the first time. If you want to change or add to the configuration of a previously configured PortServer II, refer to subsequent chapters.

In this Chapter

This chapter describes a "quick start" method of setting the basic functions of PortServer II, including its connection to the Ethernet. It also provides a general description of how to use the configuration commands how to assign users.

The Chapter includes the following topics:

Entering Configuration Commands

To configure PortServer II, you enter commands at PortServer II's command line prompt on the configuration terminal. Type the command syntax including any options on the configuration terminal keyboard. The descriptions of the commands in this User's Guide include only the most common options, and for details of all command options you should refer to the *Command Reference Guide*.

Note: Characters that you must type are shown in the format:

```
command syntax [options]
```

Abbreviations

All PortServer II commands may be abbreviated to their shortest *unique* initial letter or letters. For example, you can abbreviate the command:

```
set user name=jill
```

to:

```
set u n=jill
```

If you enter insufficient letters for PortServer II to identify the command, it will return an error message.

You may enter an asterisk ("`*`") in place of "`range=1-`*n*" to specify all of the ports or tables entries of the same type.

Editing keystrokes

You can use any of the following keystrokes when entering commands at the PortServer II command line:

Ctrl + b	Move the cursor left (move back)
Ctrl + f	Move the cursor right (move forward)
<--	Erase (delete character to the left of the cursor)
Delete	Delete the character under cursor
Ctrl + h	Delete the character to the left of the cursor. (You can change "h" to another character with the `set keys` command, as described in the *Command Reference Guide*.)
Ctrl+p	When typed as the first character on a line, this sequence copies the previous line from the command buffer onto the current line.
Ctrl+n	When typed as the first character on a line, this sequence copies the next line from the command buffer onto the current line.
Enter	Executes the currently typed command.
Ctrl [Enter	Escape character. Used to break out of the current session, either to another session or to the PortServer II command line. (You can change "[" to another character with the `set keys` command, as described in the *Command Reference Guide*.)
Ctrl+c	Interrupt character. (You can change "c" to another character with the `set keys` command, as described in the *Command Reference Guide*.)

Specifying the range of set configuration commands

As a part of a `set` command, you specify which port or table entry to which the command applies. The `set flow`, `set framerelay`, `set FrDcli`, `set keys`, `set line`, `set logins`, and `set ports` commands are applied to a port or range of ports, while other set commands are applied to a table index.

Note: If you do not specify a range, the `set` command will affect only the port from which you are logged in.

Specify a table entry, or a range of table entries, with the range option, in one of three formats:

**`set (`*`command`*`) range=`*`value`*

This tells the `set` command that the rest of the command affects the port number *value*. Type in the rest of the command after *value*.

**`set (`*`command`*`) range=`*`startvalue-endvalue`*

This tells the `set` command that the rest of the command affects the port numbers *startvalue* through *endvalue*. Type in the rest of the command after the value for *endvalue*.

It is possible to combine the preceding two methods, separated by commas:

**`set (`*`command`*`) range=`*`value,startvalue-endvalue,value`*

You can include multiple command options in the same line with a single range option. For example:

```
set line range=1-4 baud=2400 parity=e csize=7 stopb=1
```

This example sets the serial characteristics of ports 1 through 4 for 2400 baud, even parity, character size of 7 and 1 stop bit. This could also have been entered as:

```
set line range=1-4 baud=2400
set line range=1-4 parity=e
set line range=1-4 csize=7
set line range=1-4 stopb=1
```

Saving configuration changes to flash ROM

Each `set` command includes an option to save configuration changes into PortServer II's Flash Read Only Memory (ROM). If you logged in as root, `save` is on by default. If you are not logged in as root, `save` is off by default.

Important: Changes made by a non-root user are valid only for the duration of their session.

Non-root users can only set parameters for their own port. If non-root users wish to save their parameters in Flash ROM, the configuration command must include `save=on`. The System Administrator must also configure this port with `set logins write=on`.

Since configuration changes are automatically saved for root users, you must remember to add `save=off` to any temporary changes you make when logged in as `root`.

On-Line Help

On-line help screens are available for the PortServer II's commands. Help is displayed if you enter a "?" by itself or after part of a command line. The help screen will tell you which parameters are available to complete the command line.

Help menu

Enter a "?" after the PortServer II prompt with no other parameters to get the top level help screen, as shown below:

```
login: root
passwd:
#> ?
admin       boot       close    cpconf
exit        info       kill     mode
newpass     ping       quit     rlogin
send        set        snmp     status
telnet      traceroute wan      who
?
#>
```

Command-specific help

For help on a specific command, type the command name followed by a question mark. The example below asks for help for the command `info`:

```
#> info ?

Displays or clears statistics tables.

syntax: info (table_name)|(table_cmd)

tables:

frame       ip       icmp           tcp
udp         clear

#> info
```

Note: After you enter a request for help, PortServer II prints the command before the "?" for you on the next command line, anticipating that you will enter the command next. For example, after the request for help with `info?` above, notice how the next command prompt is followed by the word `info`.

Logging on to PortServer II

To configure PortServer II with this procedure, you will use a terminal or PC that is connected to one of the serial ports, as described in *Chapter 3, Installation*.

Note: If you want to configure PortServer II remotely with RARP (not using a local terminal), refer to Configuring PortServer II over the Ethernet on page 64.

To log on to PortServer II, proceed as follows:

1. Switch on the terminal or PC. If you are using a PC, start the terminal emulation software.
2. Set the terminal parameters to VT-100 emulation, 9600 baud, 8 data, 1 stop, and no parity.

   Note: These are the factory default PortServer II settings and can be changed. If you are reconfiguring a PortServer II whose parameters have been changed, use the new parameters. If you do not know the new parameters, refer to *Chapter 17, Troubleshooting* for details of how to restore the factory-default settings for a port.
3. Switch on PortServer II. The power switch is located on the front of the case. PortServer II's Power On Self Test (POST) sequence will now run automatically. If it does not detect any errors, the alphanumeric display will show **AC**, and the LED indicators will light in sequence.
4. When **AC** is showing on the alphanumeric display, press the **Return** or **Enter** key on your terminal or PC keyboard. The `login:` prompt should display.
5. Type `root` and press **Enter**. Check that the `passwd:` prompt should display.

   Note: Always log on as `root` to configure or reconfigure PortServer II.
6. Type `dbps` and press **Enter**. The password you type is not displayed on the screen. When you see a #> prompt, you have successfully logged in, for example:

```
login: root
passwd:
#>
```

Note: These are the factory default PortServer II login and password, and can be changed. If you are reconfiguring a PortServer II whose login and/or password have been changed, use the new entries. If you do not know the new login and/or password, refer to *Chapter 17, Troubleshooting* for how to restore the factory-default settings.

Configuring the Ethernet Connection

You now should enter basic network configuration information for the PortServer II. The information you enter is stored in the network parameters configuration table.

1. Set the IP address for the PortServer II unit, using the following command:

   #>set config ip=xxx.xxx.xxx.xxx

   Your network administrator can give you an appropriate IP address to use. For example, to set an IP address of 192.83.159.1, enter:

   ```
   #>set config ip=192.83.159.1
   #>
   ```

   Note: If your system supports RARP (Reverse Address Resolution Protocol), you may be able to acquire the IP address automatically. See *Configuring PortServer II over the Ethernet* on Page 64.

2. Set the Node Name for the PortServer II, using the following command:

   #>set config myname=<node_name>

   For example, to set the node name to `termserver`, enter:

   #>set config myname=termserver

3. If applicable, enter the name of the domain that includes the PortServer II unit, using the following command:

   #>set config domain=<domain_name>

   For example, to set `dgii.com` as the domain name, enter:

   ```
   #>set config domain=dgii.com
   #>
   ```

4. If you have one, enter the IP address of the Domain Name Service machine that will translate domain names to and from IP addresses, using the following command.

   #>set config nameserv=xxx.xxx.xxx.xxx

   For example, to set 192.83.159.2 as the Domain Name Server, enter:

   ```
   #>set config nameserv=192.83.159.2
   #>
   ```

Note: If you do not have a Domain Name Server, you can use the PortServer II's `set host` command to build a table of names and IP addresses. (See the *Command Reference Guide* for the command syntax.)

5. Set your network sub-net mask, using the following command:

   ```
   #>set config submask=xxx.xxx.xxx.xxx
   ```

   For example to set the mask to 255.255.255.240, enter:

   ```
   #>set config submask=255.255.255.240
   #>
   ```

   Note: The default submask is 255.255.255.0.

6. If you have a gateway, set its IP address, using the following command:

   ```
   #>set config gateway=xxx.xxx.xxx.xxx
   ```

   For example, to set the gateway address to 198.83.159.3, enter:

   ```
   #>set config gateway=198.83.159.3
   #>
   ```

7. Check the time and date, by entering:

   ```
   set time
   ```

   Check that the correct time and date are displayed. If the correct values are not displayed, enter:

   ```
   set time time=hour.minute.second date=month.day.year
   ```

   where the time and date values are the current values, for example:

   ```
   set time time=11.32.00 date=4.1.96
   ```

Configuring PortServer II over the Ethernet

If you do not set the IP address during configuration, PortServer II will attempt to use RARP (Reverse Address Resolution Protocol) to determine its IP address. If this succeeds, you can configure the remaining PortServer II parameters by connecting to it using telnet from an administration terminal, instead of using a local terminal connected directly to PortServer II.

If you want to use this method of configuration, a RARP server should be available on the network. Consult your operating system and network software documentation for further information. (On many UNIX systems, this information can be found in the manual entry for `rarpd`).

Note: If you do not have a RARP server, you can configure most PortServer II parameters using **telnet**. However, you must set PortServer II's IP address with a local terminal before you can use **telnet**.

Running RARP on the server

To use RARP to assign IP addresses to PortServer II's on your network, the RARP daemon must be running on the network host. The RARP server maintains a database of mappings between Ethernet (MAC) addresses and protocol (IP) addresses. When a client such as PortServer II requests an IP address, the RARP daemon accesses the database and returns the appropriate address.

Note: The example below is for an SCO server; the file names and responses on your system may differ.

To enable RARP on a server:

1. Ensure that TCP/IP is running on the server.
2. Add each PortServer II on your network to the **/etc/hosts** file, in the following format:

```
# @(#)hosts  1.2 Lachman System V STREAMS TCP source
# SCCS IDENTIFICATION
127.0.0.1  localhost
132.147.144.246  dgii dgii.com
```

3. Add each PortServer II on your network to the **/etc/ethers** file, in the following format:

```
# @(#)ethers  4.1 Lachman System V STREAMS TCP source
# SCCS IDENTIFICATION
#
```

```
#sample ethers file
#
8:0:20:0:fc:6f  dgii
```

4. Remove the comments (# symbols) from the following lines in the **/etc/rc2.d/S85tcp** file, to ensure the RARP daemon runs each time the server is brought up in multi-user state:

SCO Unix Version 4.2 and below:

```
#  if [ -x/etc/rarpd -a -f /etc/ethers ]; then
#          /etc/rarpd -a ; echo “rarpd\c”
# fi
```

SCO Unix Version 5:

```
#  if [ -x/etc/rarpd -a -f /etc/ethers ]; then
#          /etc/rarpd -a &
#  echo “rarpd\c”
# fi
```

Note: If you need to uncomment these lines in either of the examples above, you must start RARPD manually or reboot your server.

5. Reboot PortServer II, and check that you can telnet or rlogin to PortServer II.

Testing the network connection

You should now check that you have configured PortServer II correctly by establishing communications with another network machine.

Enter the `ping` command to test the connection. For example, to test the connection by pinging the machine with IP address 192.83.159.227, enter:

```
#>ping 192.83.159.227
```

Check that you see a response similar to:

```
192.83.159.227 is alive
#>
```

If you do not see this message, check that you have set the IP address for PortServer II correctly, and have entered the address of the target machine correctly. Check also that you have connected the network cable to PortServer II correctly. You can also use PortServer II’s front panel to monitor Ethernet activity – see *Chapter 2, Operation*. If you are unable to resolve the problem, refer to *Chapter 17, Troubleshooting*.

PortServer II TCP/IP Port Numbers

PortServer II provides two ranges of TCP/IP service numbers to which you can connect from other systems:

2001 - 2099 Telnet Connect

2101 - 2199 Raw Connect

A connection made to any 20*XX* port uses the Telnet protocol, providing full Telnet compatibility. Use the 20*XX* ports in commands like:

```
pr myfile|telnet dbps-nodename 2001
```

A connection to any 21*XX* service is a raw connection, passing eight bit "clean" data. Use the 21XX ports with RTTY (see *Chapter 9*) and custom applications.

The last two digits in each number indicate the port or group the user is connected to. If the last two digits are in the range 01 through 64, the user is connected to the specified port. Numbers greater than 64 indicate a **port group number**, as configured with `set port group=`*group_number* (See *Command Reference Guide*).

When several PortServer II ports are configured with the same group number, they work like a telephone "hunt group." Any attempted connection goes to the lowest numerical port number that can currently accept a connection.

The `set altip` command (See *Command Reference Guide*) can be used to define alternate IP addresses for telnetting to a port or hunt group. (This command does not support raw connections.) For example, you could replace `telnet abc 2088` with `telnet abcalt`, if that `abcalt` has an IP address associated with group 88 in the `altip` table, and `abcalt` is defined in a host table.

Note: Choosing the wrong range of service numbers, (for example, 20*xx* instead of 21*xx*) is a common source of user problems.

Configuring a User

For each PortServer II you can configure between one and 64 users, and a password for each user. These users typically have their own logins, and do not have access to the PortServer II configuration options. Users may log in via dial-up links or at terminals. The configuration options available depend on their method of access, as described below.

If you do not have users connecting to your PortServer II (for example, if you are using it as a printer server), you can ignore the rest of this chapter.

Note: If a user can log in by more than one method (for example, through a local terminal or a dial-up link), you must create a user profile for each.

Creating a new user

You must create an account for each user who can connect through PortServer II, and configure their basic access and network parameters as follows:

1. Create a user by entering a command in the format:

   `set user name=<user_name>`

   <user_name> would typically be the same user name as on your host systems. For example, to create a user called **Bill**, enter:

   `set user name=Bill`

   PortServer II displays the list of basic access parameters that you will configure for the user. If this is a new user, default settings are shown for the parameters, as shown below:

```
User 1              Name=Bill
Access:
CommandLine=off     Password=off         Ports=00000000 00000000 (1-16)
AutoConnect=off     SessionTimeout=86400       0x0x0x00 00000000 (17-32)
NetService=off      IdleTimeout=86400          00000000 00000000 (33-48)
Menu=off            MaxPorts=64                00000000 00000000 (49-64)
DefaultAccess=CommandLine
AccessTime=
AutoPort=20513      AutoService=Default      AutoHost=199.186.118.253
```

2. Enter commands to configure the Basic Access parameters for the user, by entering commands in the format:

   `set user name=Bill <parameter>=<value>`

   You can include more than one parameter in each command. For example:

```
set user name=Bill AutoPort=20513 AutoService=Default Auto-
Host=199.186.118.253
```

Has the same effect as entering:

```
set user name=Bill AutoPort=20513
set user name=Bill AutoService=Default
set user name=Bill AutoHost=199.186.118.253
```

The Basic Access parameters are as follows:

AccessTime The user is only permitted access during the period(s) configured. If this is an outgoing connection, the link is only kept up during this period. Enter a timestring containing any combination of the following:

time - day or day hours
day - `mo`, `tu`, `we`, `th`, `fr`, `sa`, `su`, `wk`
hours - "`hr:min - hr:min`"
hr - 0 through 23
min - 0 through 59

Note: `wk` indicates the "working week", Monday through Friday

Examples:

```
AccessTime="wk sa su"
```

This allows access 24 hours a day, 7 days a week.

```
AccessTime="wk9:00-17:00"
```

This allows access between 9AM and 5PM, Monday through Friday

```
AccessTime="wk0:00-8:59 wk18-23:59 sa su"
```

This allows access any time *except* during the working week.

AutoConnect If this is set to `yes`, the user may be automatically connected to another machine, without getting a local PortServer II command line prompt.

AutoHost If **AutoConnect** is set to `yes`, this contains the IP address of the machine that the user is connected to.

AutoPort If **AutoConnect** is set to `yes`, this contains the TCP/IP port for the connection. If `0`, the TCP port is based on the **AutoService** entry below. `513` sets Rlogin (default), `23` sets Telnet, or `9` sets "raw".

AutoService The protocol for the automatic connection, which may be `Telnet`, `Rlogin`, `Raw`, or `Default`. `Default` sets the protocol based on the AutoPort entry. `Raw` passes data between the serial port and the TCP stream with no modification.

Command-Line If this is set to `on`, the user has access to the PortServer II command line. If it is set to `off`, the user cannot issue PortServer II commands.

DefaultAccess Selects the service that is presented to the user, and may be set to `CommandLine`, `Menu`, `AutoConnect`, `NetService`, or `Outgoing`:

- `CommandLine` presents the PortServer II command line
- `Menu` presents the menu configured for the user
- `AutoConnect` makes an automatic connection, as specified by the options above
- `NetService` starts a PPP or SLIP/CSLIP session
- `Outgoing` indicates that this user is permitted only outgoing connections

IdleTimeout The maximum time in seconds for which the connection may be idle before the user is disconnected. `0` means there is no limit.

MaxPorts The maximum number of ports that the user can log into at the same time. `0` means the user can log into an unlimited number of ports.

Menu The number of the menu that is presented to the user, if any. `0` or `no` means no menu is presented.

NetService If this is set to `on`, the user can initiate an incoming PPP or SLIP/CSLIP session. If this is set to `off`, the user does not have incoming network access.

Password If this is set to `on`, the user must enter a password when logging in. If set to `off`, no password is required.

Ports Sets the ports that the user has access to. You can also enter a range of values, for example, `Ports=1-16` permits access on all of ports 1 through 16.

SessionTimeout The maximum time in seconds for which the user may be connected before the user is disconnected. `0` means there is no limit.

3. Enter commands to configure the Network parameters for the user, by entering commands in the format:

   `set user name=Bill <parameter>=<value>`

   As in step 2, you can include more than one parameter in each command. PortServer II displays the list of network parameters that you will configure for the user. If this is a new user, default settings are provided for the parameters, as shown below:

```
User 1     Name=Bill
Network:
IPAddr=123.123.123.123      Protocol=PPP         BringUp=Filter_name
IPMask=255.255.255.255      Compression=none     KeepUp=Filter_name
LocalIPAddr=123.123.132.132 VJSlots=16           PassPacketFilter_name
NetRouting=receive          MTU=1500             LogPacket=Filter_name

PPPAuth=both                Passive=off          Dialout=off
PapID=identifier            AddrCompress=off     LocalBusyDly=01
PapPasswd=password          ProtoCompress=off    RmtBusyDly=01
ChapID=identifier                                DownDly=01
ChapKey=key_value                                Device=device_name
                                                 LoginScript=script_name
index             phonenumber               parameters
1                 470-9777                  ppnx1
2                 470-9778                  midnight
3
4
5
6
7
8
9
```

The Network parameters are as follows:

AddrCompress If set to `on`, PortServer II attempts to negotiate address compression on any PPP connection. If set to `off`, no negotiation is attempted.

BringUp The name of a BringUp filter that PortServer II will use to initiate a remote connection. See *Chapter 9, Configuring Modem Connections* for information on filters.

ChapID The CHAP authentication identifier that PortServer II sends to any remote system that requests it. Any identifier that the remote system sends must match the user name.

ChapKey The CHAP key that PortServer II uses to generate a reply to the remote system.

Compression If set to `VJ`, Van Jacobsen TCP compression is applied to packets on SLIP and PPP connections. If set to `none`, no compression is applied.

Note: SLIP connections employing Van Jacobsen TCP compression are referred to as Compressed SLIP or CSLIP connections.

Device The name of the device or device pool (typically a modem pool) that is used for outgoing connections.

Dialout If set to `on`, outgoing network connections are enabled. If set to `off`, the user is not permitted to initiate outgoing network connections.

DownDly If set to a value, the dialer script waits for the specified number of seconds before attempting to re-establish a host that is reported as down.

IPAddr If set to an IP address, this specifies the remote host to connect to. If set to `0.0.0.0` or `negotiated`, the remote host negotiates the address to use. If set to `ippool`, the address is obtained from the dynamic IP address pool (see *Configuring the IP Pool* later in this chapter for more details).

IPMask The IP mask that is applied to the address specified in IPAddr. If the mask is not `255.255.255.255`, PortServer II assumes a router is present on the other end of the connection, and the router can forward packets to all other addresses on the network or subnet specified.

Keepup The name of a KeepUp filter that PortServer II will use to maintain the connection. See *Chapter 9, Configuring Modem Connections* for information on filters.

LocalBusyDly Sets the number of seconds that PortServer II delays before trying to establish a connection when no ports were available at the previous attempt.

LocalIPAdr Assigns an IP address to the local end of a point-to-point link. If this is set to `0.0.0.0`, the IP address of PortServer II's Ethernet interface is used.

Loginscript The name of a script that PortServer II uses to log in to the remote system and start the network connection. See *Chapter 9, Configuring Modem Connections* for information on scripts.

LogPacket If you enter the name of a filter, any packets passing through the WAN interface and matching this filter are logged by PortServer II. See *Chapter 17, Troubleshooting*.

MTU Maximum Transmit Unit, which is the maximum size of the packet that can cross the WAN interface. For a PPP connection, the maximum packet size is negotiated, and you should enter the largest packet size that PortServer II will permit the remote system to send.

phonenumber Alternative phone numbers to dial to request the connection. If you enter more than one number, PortServer II tries them in turn if it receives a Busy signal from the remote system.

NetRouting If set to `off`, routing protocol packets are not sent and are ignored if received. If set to `send`, routing packets are sent, but ignored if received. If set to `receive`, routing packets are not sent, but are acted on if received. If set to `both`, routing packets are both sent and acted on.

Note: No routing packets are sent unless you also turn forwarding on, by entering:

```
set forwarding state=active
```

parameters Parameters for the login script. Typically, **p1** is the login name and **p2** is the password to send, but this depends on the script. See *Chapter 9, Configuring Modem Connections* for information on scripts.

PapID The PAP authentication identifier that PortServer II will send to a remote system if requested. (Any identifier that the remote system returns must match the user name.)

pappasswd The PAP password that PortServer II will send to a remote system if requested. (Any password that the remote system returns must match the user's password.)

Passive If set to `on`, PortServer II waits for the remote system to begin PPP negotiations. If set to `off`, PortServer II may initiate negotiations.

PassPacket If you enter the name of a filter, any packets not passing the filter are blocked by PortServer II. See *Chapter 9, Configuring Modem Connections* for information on filters.

PPPAuth If set to `none`, the remote system is not authenticated. If set to `PAP`, the remote system is authenticated with PAP. If set to `CHAP`, the remote system is authenticated with CHAP. If set to `both`, the remote system negotiates what authentication is used.

Protocol Sets the protocol on the connection, and may be `PPP`, `SLIP`, or `FR` (Frame Relay).

ProtoCompress If set to `on`, PortServer II tries to negotiate protocol compression. If set to `off`, no protocol compression is attempted.

RmtBusyDly Sets the number of seconds that PortServer II delays before trying to establish a connection to a host that is reported as busy.

VJSlots Sets the maximum number of slots to use for Van Jacobsen header compression (see **Compression** above). Each active TCP transaction on a point-to-point connection requires one VJ slot to hold information about the compression state. If there are more transactions than slots, some compression information is lost.

Note: For information on the Frame Relay parameters attached to the Network options, refer to *Chapter 14, Configuring Frame Relay.*

Using the IP Pool

You can configure PortServer II to use a “pool” of IP addresses for devices connected to it. PortServer II then assigns an address from the pool each time a device requests a session.

Creating a pool

To provide a range of IP addresses to use as a pool, enter a command similar to the following:

```
set ippool ip=2.4.1.30 count=15
```

In this example, **2.4.1.30** is the first address in the pool and **15** is the number of addresses in the pool.

Important: The IP pool uses a series of consecutive addresses and all addresses in the pool must be available to PortServer II. For instance, in the example above, all addresses between **2.4.1.30** and **2.4.1.44** will be assigned to PortServer II.

Assigning a device to use an address from the IP pool

To configure a device to take an address from the IP pool, enter its IP address as `255.255.255.254` or `ippool` each time you enter a configuration command containing its IP address.

Note: The device may have a different IP address for each session.

Configuring a user for manual or automatic login and connection to a host

The procedures given below are typical methods of configuring a user's connections to a host. The first procedure configures a user who must log in for each session with their own login and password. The second procedure configures a user who will automatically be connected to another machine. The third procedure describes how to provide the user with a menu of connection options each time they log in.

Configuring a user for manual login

1. If you have not already done so, create a new "regular" user by entering the following command:

   `set user name=<username> [options]`

   A "regular" user is any user other than `root`. The options specify whether the user is required to enter a password and whether successful login results in automatic connection to another machine. For example, to create a regular user called **Linda**, enter:

   `#>set user name=Linda`

   For detailed information, refer to `set user` in the *Command Reference Guide*.

   The user names are stored in a table of users that you can view by typing `set user` without any other parameters.

   Note: User names are case sensitive; **Linda** is not the same user as **linda** or **LINDA**.

   Important: For new users, the user's password is the same as the user name, with the same capitalization. In the example, Linda's initial default password is the same as her username, **Linda**. The password can be changed as described in Step 3.

2. By default, a user must enter their password each time he or she logs in. We suggest you check that the option is set correctly by specifically entering the command described below. Alternatively, if security is not a concern, you may want to remove the requirement for a password.

   `set user name=username passwd=[on][off]`

   For example, to ensure Linda enters a password each time she logs in, enter:

   `#>set user name=Linda passwd=on`

Note: When you are more familiar with PortServer II commands, you can specify the password requirement when you define the user name (step 1).

3. For additional security, you may want to change the new user's password immediately. A password can be any combination of letters and numbers up to a maximum of 16 characters, and is case-sensitive. To change a password:

   a. Enter:

   `newpass name=<user_name>`

   where `<user_name>` is the name of the user whose password you want to change. Check that the following prompt is displayed:

   `current passwd:`

   b. Type in the new password and press `Enter`.

   Note: The password you type is not echoed onto the screen.

   Check that a prompt for the new password is displayed.

   c. Type in the new password again and press `Enter`.

   Note: Each user can change his or her own password, using the `newpass` command as described in the procedure above. Assuming the user logged in with his or her own user name (under his or her old, original password), it is only necessary to enter `newpass` without options.

   For detailed information, refer to `newpass` in the *Command Reference Guide*.

4. You can change the login prompt for each user on one port or a range of ports from the default **login**> by entering:

   `set logins logprompt=[options]`

   For example, to change the login prompt to **Digi Int**> on all 16 ports, enter:

   `#>set logins logprompt="Digi Int>" range=1-16`

   The maximum login prompt length is 10 characters, and may include both letters and numbers. Include quotation marks around the prompt if it includes a space.

   You can change the login prompt for a single port by specifying its number in the **range=** option, for example, **range=2**.

   If your terminal supports multiple sessions, you can now refer to *Chapter 7, Configuring Multiple Sessions and Multiple Screens* for details of how to configure sessions.

Configuring a new user for automatic login and connection

PortServer II includes automatic login and connection options, which are described in *Chapter 6, Configuring Security.* The following procedure describes how to create a user who is automatically connected to a server:

1. Create the new user and set the password, as described in the previous section.

2. Configure the user for automatic connection and enter the destination host, using the following commands:

   ```
   set user name=<user_name> AutoConnect=Yes
   set user name=<user_name> AutoHost=xxx.xxx.xxx.xxx
   set user name=<user_name> AutoPort=xx
   set user name=<user_name> DefaultAccess=AutoConnect
   ```

   For example, to configure user “Peter” as an *auto* user, who is automatically connected to the destination host **corporate** that we defined in step 2, enter the following sequence of commands:

   ```
   #>set user name=Peter AutoConnect=Yes
   #>set user name=Peter AutoHost=152.83.151.1
   #>set user name=Peter AutoPort=telnet
   #>set user name=Peter DefaultAccess=AutoConnect
   ```

   Setting `AutoPort` is necessary to establish a terminal session with the host. The example configures a telnet-type session, and Peter will be asked to log in again when the connection to **corporate** is established. If you specify an `Rlogin` session, Peter will automatically be logged in to **corporate** *if his login on* ***corporate*** *is the same user name as his PortServer II login*. You can also set `AutoPort=Default` to use the user’s default protocol or `raw` (no modification to data passing between the serial port and the TCP stream).

Providing a navigation menu for each user

You can configure a menu for each user that is displayed automatically when he or she logs in or enters `menu` at the command line prompt. The menu would typically display a list of servers or systems to which the user can connect. Each menu can have up to 20 lines, including two titles lines. You define each line of the menu with a separate command.

1. Create each line of the menu, by entering the following command as many times as necessary:

   `set menu range=<menu> [options]`

   <menu> is a unique number that identifies this menu. The options specify a command to execute or an informational message to display.

   For example, the following five commands specify a complete menu:

   ```
   set menu range=1 t1="     Digi International Inc."
   set menu range=1 t2="Networking Products and Solutions"
   set menu range=1 m1="Choose one of the following systems:"
   set menu range=1 c3="telnet corporate"
   set menu range=1 c5="telnet engineering"
   ```

   In this example, **t1** and **t2** are title lines, **m1** is an informational line, and **c3** and **c5** are command lines. For detailed information, refer to `set menu` in the *Command Reference Guide*.

   Note: To remove a line, type `rmentry=<line_number>`.

2. Check the menu seen by the user is correct by logging in with their normal user name and password:

   For the example in step 1, the menu should appear as shown below:

   ```
        Digi International Inc.
   Networking Products and Solutions

   Choose one of the following systems:

   1 - telnet corporate

   2 - telnet engineering


   0 - exit
   ```

Note: Notice the double spacing between lines.

When the user selects a command line and presses `Enter`, the command specified on that line is executed.

Note: You can include blank lines in a menu.

3. Configure a user to the menu. For example:

```
set user name=fred menu=1 DefaultAccess=menu
```

Removing a user

To remove a user's login and associated information from PortServer II, enter the following command:

```
set user rmuser=<user_name>
```

For example, to remove the user called **Peter**, enter:

```
set user rmuser=Peter
```

You can also remove several users by specifying a range of index numbers:

```
set user rmuser=on range=<range>
```

You can identify the range of index numbers to use from the User Table. To view the User Table, enter `set user` with no options.

For detailed information, refer to `set user` in the *Command Reference Guide*.

Changing a user's name

To change a user's name without affecting other definitions such as the password, enter the following command:

```
set user newname=<newusername> name=<oldusername>
```

For example, to change Peter to Linda, enter:

```
set user newname=Linda name=Peter
```

For detailed information, refer to `set user` in the *Command Reference Guide*.

Users logging on to PortServer II

When you have configured a user, he or she can log in to PortServer II (if permitted). Alternatively, PortServer II may route the login request to any autoconnect host that you have defined. To log in, the user enters a command in the format:

```
login (options)
```

To use the default access login type, the user enters a command similar to the following example:

```
login:steve
```

To access an autoconnect host, the user enters a command similar to the following example:

```
login:steve autoconnect
```

The autoconnect option can only be used if you have set `AutoConnect=on` or `DefaultAccess=AutoConnect` for the user.

To access the PortServer II command line, the user enters a command similar to the following example:

```
login:steve CommandLine
```

The CommandLine option can only be used if you have set `CommandLine=on` or `DefaultAccess=CommandLine` for the user.

To display the login menu (if configured), the user enters a command similar to the following example:

```
login:steve Menu
```

To access the configured network service, the user enters a command similar to the following example:

```
login:steve netservice
```

Note: The `autoconnect`, `CommandLine`, `Menu`, and `netservice` options can be abbreviated, provided sufficient unique characters are typed to allow PortServer II to identify the command.

Note: The **login** command is also passed to the RADIUS server, if one is available.

Using configuration commands

You can use the `set` command with different variables to change individual parameters or to add new devices and functionality. The variables available are summarized below, and their use is described in subsequent chapters. You can also find information on `set` command syntax in the *Command Reference Guide*.

altip	Defines an alternate IP address for a PortServer II port or a group of ports, allowing (for example) it to be included in hunt groups.
arp	Sets ARP (Address Resolution Protocol) Table entries, which map IP addresses to Ethernet addresses.
auth	Sets the access permissions for PortServer II's serial ports.
chat	Defines telephone number strings that PortServer II will translate.
config	Sets configuration options, including an IP address for PortServer II.
device	Sets devices for PortServer II to use when making outgoing connections.
filter	Sets filters that control and record traffic over PPP, SLIP, and CSLIP links.
flow	Sets hardware and software flow control parameters for RS-232 serial ports.
forward	Sets IP packet routing options for connections that use PPP, SLIP, and CSLIP links.
frame relay	Sets parameters for connecting a port to a Frame Relay service.
FrDLCI	Sets the configuration for a single virtual connection on a serial port.
host	Sets hosts that PortServer II can communicate with.
keys	Sets local key combinations that have specific functions on the PortServer II local terminal, for example, `kill` and `erase`.
line	Sets or views the parameters for a single serial port or group of ports.
logins	Sets login, password, and command line prompts.
menu	Creates a menu from which the user can select connection options.
modem	Sets initialization and tests scripts to use with a modem.

ports	Sets a serial port or group of ports for a particular device type.
radius	Sets RADIUS authentication options, if you have a RADIUS server on your network.
route	Sets routing parameters for connections that support IP routing.
script	Defines a modem or login script.
service	Associates common TCP and UDP service ports with names for use with filters.
terms	Sets terminal types and parameters for multiple sessions.
time	Sets or displays the time and date held by PortServer II.
trace	Records and displays trace messages that are held in the system log.
user	Sets the names of users who can log into PortServer II.

Chapter 5

Configuring Terminals

When should I read this Chapter?

Read this chapter if you want to connect an additional terminal to a previously-configured PortServer II, or if you want to change the configuration of a terminal that is already connected.

In this Chapter

This chapter describes how to configure PortServer II for connection to a terminal.

The Chapter includes the following topics:

Setting Up a Terminal

The following procedure assumes that you have connected the terminal that you wish to configure to PortServer II, following the procedures in *Chapter 3, Installation*. If the terminal is connected to PortServer II via a modem or a network, use the appropriate procedure from Chapters 8 through 10.

1. Check that there is an administrative terminal or PC connected to one of the serial ports, as described in *Chapter 3 Installation*. If you do not have a terminal or PC connected, connect one now. Alternatively, you can telnet into PortServer II from another machine connected to the Ethernet.

2. Log into the PortServer II as `root`, as described in *Chapter 4, Basic Configuration*.

3. Set the terminal type for the port that your terminal is connected to, using the following command:

   `set ports termtype [options]`

   The options specify the type of terminal that you have connected. For example, if your terminal is a Wyse 60 connected to Port 2, enter:

   `#>set port termtype=wy60 range=2`

   The terminal type name (`termtype`) depends on the conventions accepted by operating systems on your network hosts; refer to your host documentation for more details.

   The `range=2` parameter tells the PortServer II software that this command is to affect Port 2. If you do not specify the range, the port that the administrative terminal is connected to is changed to Wyse 60 operation.

   Note: You can specify other options in the `set ports` command in addition to the terminal type, including automatic connection when the user logs in, time delays between packets sent to the user, groups of like devices, and the maximum number of sessions available to the user. For detailed information, refer to `set ports` in the *Command Reference Guide*. Take particular note of the `save` option that determines whether your changes are save permanently in PortServer II's memory.

4. Set the baud rate and operational parameters of the port that the terminal is connected to using the following command:

   `set line [options]`

The options specify baud rate and character processing for the serial port, and depend on the terminal's specification. For example, to set Port 2 to a baud rate of 38400, enter:

```
#>set line baud=38400 range=2
```

For detailed information, refer to `set ports` in the *Command Reference Guide*. Take particular note of the save option that determines whether your changes are save permanently in PortServer II's memory.

5. Set the flow control parameters using the following command:

   ```
   set flow [option]
   ```

   The options specify flow control, control character usage, and ALTPIN usage. For example to set Port 2 to software flow control, enter:

   ```
   #>set flow ixon=on range=2
   #>set flow ixoff=off range=2
   ```

   (In this example, `set flow ixon=on` sets Port 2 to use software flow control (typically XON/XOFF) for output data, and `set flow ixoff=off` sets Port 2 to turn off software flow control for input data (For a terminal, input data will be from the keyboard and should not require flow control).

   For detailed information, refer to `set flow` in the *Command Reference Guide*.

   Note: The two set flow commands in the example could have been entered with one command:

   ```
   #> set flow ixon=on ixoff=off range=2
   ```

   These steps are sufficient to configure a terminal that is connected directly to PortServer II and will use **telnet** or **Rlogin** to request a connection.

   Note: You can change the key sequences for the terminal with the `set terms` command, as described in the *Command Reference Guide*.

Chapter 6

Configuring Security

When should I read this Chapter?

Read this Chapter if you want to change the security provided for a previously configured user.

In this Chapter

Chapter 4 Basic Configuration briefly described how you can provide each user with a password to increase security. This chapter covers the subject of security and includes information on configuring automatic logins to PortServer II.

The Chapter includes the following topics:

Levels of Security

PortServer II allows one of four types of security access levels to be associated with each user, as follows:

- **Root login**. The root user has access to all PortServer II's configurable parameters and consequently can change any of its functions. Root login is normally restricted to system administrators and managers.
- **Regular user login**. A regular user does not have direct access to PortServer II configuration parameters or functions, but connects to another host or system from a menu or command line prompt. PortServer II can be configured to allow a regular user to login with or without a password. A regular user can only make configuration changes to his or her own port.
- **Autoconnect user**. An autoconnect user is always connected by PortServer II to a specified host, regardless of the port or terminal from which he or she logs in. Any autoconnect user can be configured to login with or without a password. An autoconnect user does not have direct access to PortServer II parameters or functions.
- **Autoconnect port**. Any user who logs into an autoconnect port always connects to a specified host, regardless of who they are. An autoconnect port user does not have direct access to PortServer II parameters or functions.

In addition to these options, additional security can be provided for dial-in users by PPP, SLIP, or CSLIP features, and by RADIUS. These additional security options can be configured in addition to the basic login security described above. For more information, refer to *Chapter 4, Basic Configuration* (`set user` options), *Chapter 8, Configuring WAN Connections*, and *Chapter 15, Configuring RADIUS*.

You can configure a user with any of these levels of security either from the configuration menu or with command line entries, as described below. You can configure user security at the same time you configure the user (see *Chapter 4, Basic Configuration*) or as a separate procedure as described in this Chapter.

Root Login

`root` is provided as the only user when PortServer II is shipped. A default password of `dbps` is provided for the root user (see *Chapter 4, Basic Configuration*). The root login must always be present, and must always have a password. It is not possible to remove the root user entry, or delete the root user password.

However, for additional security, you can change the root password:

1. Log in as root at any terminal or PC, using the default password.
2. Enter the command:

   `newpass user=root`

3. Type in the current password and press **Enter**. The password is not displayed on the screen, but a request to enter the new password again should appear.
4. Type in the new password in exactly the same format and case as you entered in Step 3. If the two passwords match, the login is changed.

Regular User Login

This is the most common way of logging into a system. A regular user login identifies a user by name, restricts access by password, and provides limited access to the PortServer II command line after login. A regular user *cannot* change any of the PortServer II parameters, but can make certain changes to the PortServer II port at which they are logged in.

If the `set login write=on` option is selected for a user's port, the user can save changes to the parameters of his or her own PortServer II port. The new parameters become the defaults available to other users who subsequently log in on that port. If the `set login write=off` option is selected, only the root user can save port settings. For further details, refer to `set login` in the *Command Reference Guide*.

A regular user can temporarily gain root privileges if he or she knows the root password, by entering the following command:

```
admin
```

The user is prompted to enter the root password.

Regular user login with password authentication

To configure a regular user who will be asked to enter a password then he or she logs in, enter the following command:

```
set user name=<user_name> passwd=on
```

For example:

```
set user name=george passwd=on
```

Note: For details of how to create the user, refer to *Chapter 4, Basic Configuration*.

The password for a newly configured user is initially the same as their user name (`george` in the example). For additional security, the user can change his or her password with the `newpass` command (see the *Command Reference Guide* for details).

Regular user login without password

This configuration option does not require the user to enter his or her password when logging in. This feature is most useful in small, secure environments that do not have dial-up ports. It is not suitable for systems where security is a concern or where there are external network or dial-up connections.

To configure a regular user who will not be asked to enter a password when he or she logs in, enter the following command:

```
set user name=<user_name> passwd=off
```

For example:

```
set user name=george passwd=off
```

Note: For details of how to create the user, refer to *Chapter 4, Basic Configuration*.

Autoconnect User Login

An Autoconnect User is always connected to a specified host or system, regardless of the port or terminal that the user logs in from. An auto-connected user cannot ever escape to the PortServer II command line, run multiple sessions, or connect to a system other than the one configured for automatic connection.

To create auto-connect users, you must log in as `root`. You can then create three types of auto-connect users:

Auto-connection of any user on one or more ports

In this configuration, one or more ports may be configured to allow any user to log in at any time without a login or password. Any user can gain access to the PortServer II command line prompt. The configuration is only suitable for systems where security is not a major concern or if there is no possibility of unauthorized dial-up access.

To configure this mode of operation, configure the port or ports on which you want to allow access by any user, by entering the following command:

```
set login range=<port_numbers> login=off passwd=off
```

For example:

```
set login range=1-4 login=off passwd=off
```

This example allows any user access on ports 1 through 4.

Auto connection of a user with password protection

In this configuration, a particular user is connected to a specific system after entering a login and password. This user cannot access the PortServer II command line prompt. You can specify that the connection is made using the **rlogin** protocol or **telnet**. Enter the following command:

```
set user name=<user_name> Password=on AutoPort=<tcp_port#>
AutoHost=<IP_address> AutoConnect=on
DefaultAccess=AutoConnect CommandLine=off
```

This configures the user called `<user_name>` for automatic connection (`AutoConnect=on`) to the host specified by `<IP_address>`. The user is required to log in (`Password=on`).

For example, **rlogin** uses TCP port 0, so you should enter a command similar to the following:

```
set user name=martha Password=on AutoPort=0
AutoHost=192.1.2.3 AutoConnect=on DefaultAccess=AutoConnect
CommandLine=off
```

Telnet uses TCP port 23,so you should enter a command similar to the following:

```
set user name=martha Password=on AutoPort=23
AutoHost=192.1.2.3 AutoConnect=on DefaultAccess=AutoConnect
CommandLine=off
```

Auto connection of a user without a password

In this configuration, a particular user is connected to a specific system without entering a login or password. This user cannot access the PortServer II command line prompt. You can specify that the connection is made using the **rlogin** protocol or **telnet**. Enter the following command:

```
set user name=<user_name> Password=off AutoPort=<tcp_port#>
AutoHost=<IP_address> AutoConnect=on
DefaultAccess=AutoConnect CommandLine=off
```

This configures the user called `<user_name>` for automatic connection (`AutoConnect=on`) to the host specified by `<IP_address>`. The user is not required to enter a password (`Password=off`).

For example, **rlogin** uses TCP port 0, so you should enter a command similar to the following:

```
set user name=martha Password=off AutoPort=0
```

```
AutoHost=192.1.2.3 AutoConnect=on DefaultAccess=AutoConnect
CommandLine=off
```

Telnet uses TCP port 23, so you should enter a command similar to the following:

```
set user name=martha Password=off AutoPort=23
AutoHost=192.1.2.3 AutoConnect=on DefaultAccess=AutoConnect
CommandLine=off
```

Autoconnect Port

In this configuration, a port connects *any* user who logs in through it to a specified host. The port is dedicated to making a connection to a specific system and cannot connect to any other system. A user on an auto-connected port can never escape to the PortServer II command line, run multiple sessions, or connect to another system. You can also specify a range of ports that are automatically connected to a host. You can specify that the connection is made using the **rlogin** protocol or **telnet**.

On an Auto-Connect port, any port activity (such as a received character on a TTY port, or a DCD high transition on a modem port) causes an automatic connection to the specified host or system.

Enter the following command:

```
set port range=<port_numbers> dport=<tcp_port#>
uid=<user_index> dest=<IP_address> auto=on
```

This configures the range of ports specified by `<port_numbers>` for automatic connection (`auto=on`) to the host specified by `<IP_address>`.

For example, **rlogin** uses TCP port 0, so you should enter a command similar to:

```
set port range=8-9 auto=on dport=0 uid=7 dest=192.1.2.3
```

If `dport=0`, the user's uid is checked against the user index table, and the user is only permitted access if the user index is valid.

Alternatively, you could enter:

```
set port range=8-9 auto=on dport=513 uid=7 dest=192.1.2.3
```

If `dport=513`, the user's uid is not checked against the user index table.

Telnet uses TCP port 23, so you should enter a command similar to the following:

```
set port range=5-7 auto=on dport=23 dest=192.1.2.3
```

Chapter 7

Configuring Multiple Sessions and Multiple Screens

When should I read this Chapter?

Read this Chapter if you want to configure PortServer II to support multiple screens or multiple sessions on a previously-connected terminal. Read *Chapter 5, Configuring Terminals* before you read this chapter.

In This Chapter

Chapter 5, Configuring Terminals described how you can configure a simple terminal that allows the user to run a single session, or run multiple sessions and multiple screens.

The Chapter includes the following topics:

General

Any PortServer II user can run more than one session on a terminal if their permissions and operating system configuration allow this.

Multiple Sessions

Any user can start multiple sessions if his or her port is configured to allow multiple sessions. However, any user who is configured as an autoconnect user or who logs in to an autoconnect port cannot run multiple sessions through PortServer II because, to maintain security, the autoconnect configuration options restrict a user or port to a certain host.

Within these limits, PortServer II supports up to nine separate login sessions per port. For example, if your company has host computers named **Corporate**, **Engineering,** and **Production**, you can log into all three hosts, then switch between the different sessions, and to the PortServer II command prompt, using **telnet** escape commands.

Configuring multiple sessions

Configure a port for multiple sessions by entering the following command:

```
set ports sess=<number_of_sessions> range=<port numbers>
```

`<number_of_sessions>` specifies the maximum number of sessions permitted on the port and `<port_numbers>` specifies the port(s) to which the entry applies. If you do not enter a range, the entry is applied to the port at which you are currently logged in.

For example, the following command allows six concurrent sessions on Port 2:

```
set ports sess=6 range=2
```

For additional information, refer to `set ports` in the *Command Reference Guide*.

Starting multiple sessions

You initiate multiple sessions one at a time from the command line. (Start up one session, escape to the command line prompt, start another session, escape to the command line, start another session, and so on, up to the maximum number of sessions configured for that port).

To temporarily escape from a session to the PortServer II command prompt to start another session, type the **telnet** escape character followed by a `Carriage Return`. You may now start another session.

Controlling multiple sessions

Telnet sessions

In the following commands, `TESC` refers to the **telnet** escape character. The default telnet escape character is `Ctrl-]` (Control-Right Bracket), but this can be changed using the `set keys` command. See `set keys` in the *Command Reference Guide* for more details.

To leave a **telnet** session and return to the PortServer II command line, enter:

```
TESC Carriage Return
```

To switch to another **telnet** session, enter:

```
TESC <digit>
```

where `<digit>` is a valid active session number in the range 1-9.

To return to the previous session, enter:

```
TESC TESC
```

To terminate the current session, enter:

```
TESC Carriage Return close Carriage Return
```

Rlogin session

To leave an **rlogin** session and return to the PortServer II command line, enter:

Carriage Return ~ TESC Carriage Return

To switch to another **rlogin** session, enter:

Carriage Return ~ TESC <digit>

where `<digit>` is a valid active session number in the range 1-9.

To return to the previous session, enter:

Carriage Return ~ TESC TESC

To terminate the current session, enter:

Carriage Return ~ .

Switching to another session

To switch to another active session, type the session number as the first character of a PortServer II command line. You do not have to press `Carriage Return` after typing the number.

To go directly from one session to another without using the PortServer II command line, type the telnet escape character followed by the number of the session you require. For example, type the telnet escape character followed by `2` to go directly to the second session.

Type the telnet escape character followed by `Carriage Return` to go from a session to the PortServer II command shell.

As a shortcut, you can also toggle between your last two telnet sessions by pressing `Ctrl-] Ctrl-]`.

When you switch between sessions, the current session will continue on the next line of your screen unless your terminal is capable of multiple screen operation (as described later in this chapter). You may need to press the `Carriage Return` key between sessions to see your prompt.

Closing a session

To close a **telnet** session or an **rlogin** session, you can:

- Log out of the remote system, which causes the remote system to close the session.
- Escape to the command line and type the `close` command.

You may also close **rlogin** sessions using the **rlogin** escape sequence:

`Carriage Return ~ .  (Carriage Return tilde period).`

An example of multiple telnet sessions

The following example shows a user called **Linda** logging in on a terminal, logging into two hosts, switching between the two sessions, then logging out.

Note: The default **telnet** escape character is `Ctrl-]` (Control-Right Bracket), but this can be changed using the `set keys` command. See `set keys` in the *Command Reference Guide* for more details.

1. The user logs in as `Linda`:

   a. PortServer II is already up and running. Linda switches on her terminal, presses `Carriage Return` (if the login prompt is not already showing), and sees the PortServer II prompt, for example:

      `Digi Int.>`

   b. She types in her user name `Linda` and presses `Carriage Return`.

   Note: User names are case sensitive.

      She sees the password prompt, for example:

      `passwd:`

   c. Linda enters her password.

   Note: Passwords are case-sensitive. If a user enters an incorrect password or a password that includes incorrect case letters, PortServer II responds by displaying "`login invalid`...". The user has to re-enter both user name and password.

2. Linda starts a session on **Corporate**.

   At the PortServer II prompt, Linda starts a session with the host called **Corporate** by typing:

   `#>telnet corporate`

   Note: She does not have to give an Ethernet address for **Corporate** if her network has domain name service. If it does not, she could still enter `telnet corporate` if **Corporate** and its IP address have been entered in PortServer II's host table, as described in *Chapter 4, Basic Configuration, Configuring the Ethernet connection.* If neither of these conditions apply, the syntax for the telnet command is:

      `telnet hostip [tcp port#]`

3. Linda logs into **Corporate** using her login and password set up on that host.

4. Linda starts a new session on **Engineering**:

   Linda is currently logged into **Corporate** through PortServer II. She escapes the **Corporate** login prompt and returns to the PortServer II command line prompt using the **telnet** escape sequence:

   `Ctrl–]`

5. At the PortServer II command line prompt, Linda starts a session on the host computer called **Engineering** by typing:

   `#>telnet engineering`

6. Linda logs into **Engineering** using her login and password set up on that host.

7. Linda switches between sessions:

   Linda's session on **Corporate** is session 1, and her session on **Engineering** is session 2. From her **Engineering** session, she can switch to **Corporate** by typing:

   `Ctrl–] 1`

   (The telnet escape character `Ctrl–]` followed by the number `1`).

8. From her session on **Corporate**, she can switch to the **Engineering** session by typing:

   Ctrl–] `2`

   Note: She can switch between the two sessions by pressing `Ctrl–] ].`

9. From *either* session, Linda can go back to the PortServer II command line prompt by typing the telnet escape sequence `Ctrl-]` and pressing `Carriage Return`.

   Once at the PortServer II command line prompt, Linda can start other sessions (up to a maximum of nine), or enter other commands. She can go to switch either to session 1 (**Corporate**) or to session 2 (**Engineering**) by typing the number `1` or `2` respectively as the *first* character on a line.

10. Linda logs out of **Corporate**:

    Linda now switches back to her **Corporate** session (using one of the methods described in the previous step), and logs out of that session using the command sequence required by the host.

11. Linda logs out of **Engineering**:

    Linda switches to her **Engineering** session, (which is now session 1 because the original **Corporate** session 1 was closed in step 10), logs out and closes the **telnet** session.

12. Linda logs out of PortServer II.

    The terminal displays the PortServer II command line prompt. Linda exits her session by entering:

    `exit` or **`quit`**

    Note: The `exit` and `quit` commands close all active **`telnet`** and/or **`rlogin`** sessions before exiting.

Multiple Screens

Configuring terminals for multiple screens

To allow multiple screen operation, you must configure the port for operation with this type of terminal by entering the following command:

```
set ports termtype <terminal_name>
```

The PortServer II allows up to sixteen different terminal types to be configured. Each terminal type must have a distinct name, although a particular terminal may have several names if it can be used in more than one mode. For example, a Wyse 60 can be used in the following modes, each of which should have its own name:

- 43 line mode (no screen switching)
- 24 line standard mode (two screens available)
- 24 line ECON-80 mode (three screens available)
- WYSE 50+ mode (six screens available)

For each terminal, you must define the name, the number of screens used, the terminal screen clear sequence, and the sequence to switch to each screen page, using the `set terms` command (see the *Command Reference Guide*).

Note: Your terminal must support multiple screen operation for this configuration to succeed. You cannot configure a basic terminal for multiple screen operation. Refer to your terminal documentation for further information.

How to use multiple screen sessions

If your port is configured for a terminal that supports multiple screen sessions, PortServer II automatically switches between screens on your terminal when you switch between sessions.

The PortServer II screen switching algorithms were designed to maintain as much screen context as possible. It is, however, not possible to maintain screen context if you start more sessions than you have screens available. For example, you cannot display four sessions on a terminal with only three screens. In this situation, PortServer II maintains screen context on the most recently used sessions, and looses context on the unused sessions.

If your terminal is configured for three screens (for example, Wyse 60 in ECON-80 mode) you may run up to three simultaneous sessions, and switch between sessions without losing screen context. However, if you escape to the PortServer II command line or open a fourth session, PortServer II will clear and reuse the earliest screen to display the requested session. You will then be able to switch between the new session and the two most recent sessions without losing screen context. When you return to the session that was previously contained in the earliest screen, PortServer II will again clear and reuse the earliest screen to show the requested session.

Chapter 8

Configuring WAN Connections

When should I read this Chapter?

Read this chapter if you want to modify a user's connection from one of PortServer II's serial ports to a WAN (Wide Area Network). If you are modifying connections via a modem, you should read *Chapter 9, Configuring Modem Connections* before you read this chapter.

In this Chapter

This chapter describes how to configure a user's connection to a WAN, which may be an intra-company network or the Internet.

The Chapter includes the following topics:

WAN Connections Explained

Each user can connect via PortServer II to other networks or to the Internet, in addition to the local Ethernet host(s). For each user, you must define a WAN (Wide Area Network) connection that specifies parameters such as the protocol to use. This information is stored by PortServer II in the WAN Connection Table, where it is referenced by a unique index number. This index number (less 1) provides an IP address for the WAN interface. WAN connections can use PPP (Point to Point Protocol), SLIP (Serial Line Interface Protocol), or CSLIP (Compressed Serial Line Interface Protocol).

You entered the initial WAN configuration when you defined the user, as described in *Chapter 4, Basic Configuration*. Use the procedures in this chapter if you want to change the WAN configuration for a user.

Each connection may be incoming, outgoing, or both (bidirectional). For each connection, you must configure incoming connection parameters and outgoing connection parameters; some parameters are shared between the incoming and outgoing connections.

Any WAN connection requires:

- A **User Table** entry. This must include the user name, the network addresses and protocol information used to set up the connection, timeouts, and filters (if any) to use. This information may be obtained from PortServer II's internal tables or from a RADIUS server.
- **Filters Table** entry(ies) (optional), if you are using filters to restrict incoming access by the user.
- **Service Table** entry(ies), to describe any service names you are using in the filters. Actual service names are not required – a filter can be created by port numbers, rather than service names, but service names make a filter more readable.

Incoming connections also require the following:

- A **Port** configured to receive the incoming connection. You configure a port with appropriate `set` commands, including `set ports`, `set line`, and `set flow`, as described later in this chapter.

Outgoing connections also require the following:

- A **Port** configured to originate the outgoing connection. You configure a port with appropriate `set` commands, including `set ports`, `set line`, and `set flow`, as described later in this chapter.

A dialout connection also requires:

- A **Device Table** entry. This defines the port that each modem is on, and any script that will cause the modem to dial out.

A direct serial port connection also requires:

- A **Device Table** entry. This defines the port the outgoing connection is made.

A Frame Relay connection also requires:

- An **FrDLCI Table** entry that specifies options to use for the particular port/DLCI combination.
- A **User Table** entry.
    - For a dialout connection, this should also include the name of the device or device pool, and phone number (if any) to use if this is a dial-up connection.
    - For a direct connection, the name of the **Device Table** entry that specifies the port the connection should be made on.
    - For a Frame Relay connection, the Frame Relay port and DLCI numbers to use for making the connection.
    - The name of the login script and login/password parameters the script would use when logging in, if this connection requires a login chat session.
    - In the **Scripts Table,** entry(ies) that define the dialer and login scripts used by the device and user entries mentioned above, if any.

Note: Modems require additional configuration as described in *Chapter 9, Configuring Modems*.

After configuring the outgoing and incoming WAN connections, use the `wan verify` command to ensure that the configuration information you entered is correct. You can also use this command to monitor the status of an active WAN connection.

Incoming WAN Connections

How incoming connections are established

When PortServer II receives an incoming WAN connection request, it verifies that the user's login and password are contained in the User Table. If the verification is successful, it obtains the index called **uid** from the User Table. If it does not find a match, the request is referred to the RADIUS server, if one is defined.

It then brings up the incoming WAN connection that is specified by the User Table entry. It uses the remote host's IP address to create a static Route Table entry, which is numbered if the local IP address is defined, otherwise unnumbered.

On PPP connections, as part of the IP address negotiation process, the dial-in user may override the IP address for the port.

Configuring incoming connections

The following procedure gives a typical sequence of commands to modify an incoming connection. Certain of these commands are optional and you can omit them where indicated, if they are not appropriate to your connection. The order in which the commands are given below are the sequence recommended by Digi International, although you may be able to enter a different sequence of commands without affecting the functionality of the connection.

1. Configure the user of the connection, if you have not already done so. See *Chapter 4, Basic Configuration* for detailed procedures.

2. Configure the PortServer II port that will receive the incoming connection requests, by entering a command similar to the following:

   `set ports dev=<value> sess=<number_of_sessions> range=<value>`

   `dev=<value>` specifies the device type used for the connection; the most frequently used option is `mio` (modem for both input and output).

   For example:

   `set ports dev=mio sess=3 range=5`

   Note: The `set ports` command includes options that set autoconnection and modem parameters. For more information on these and additional details on the options described above, refer to `set ports` in the *Command Reference Guide*.

3. Configure the required network options for the user, by entering a command similar to the following:

```
set user name=<username> IpAddr=<ipaddress> IpMask=<ipmask>
LocalIpAddr=<local_ip_address> NetRouting=<option>
Protocol=<protocol> Compression=<type> [VJSlots=<slots>]
MTU=<size> PPPAuth=<type> [PapId=<string>]
[PapPasswd=<password>][ChapId=<string>] [ChapId=<key_value>]
Passive=<on/off> AddrCompress=<on/off>
ProtoCompress=<on/off>
```

The parameters for this command are described below:

IPAddr — If set to an IP address, this specifies the remote host that will originate the connection request. If set to `0.0.0.0` or `negotiated`, the host negotiates the address to use with PortServer II; this option is useful if connection requests may be originated by more than one host or the host addresses are unknown, for example, if your PortServer II is connected to an anonymous dial-up user. If set to `ippool`, the address is obtained from the dynamic IP address pool.

IPMask — The IP mask that is applied to the address specified in IPAddr. If the mask is not `255.255.255.255`, PortServer II assumes a router is present on the other end of the connection, and the router can forward packets to all other addresses on the network or subnet specified.

LocalIPAdr — Assigns an IP address to the local end of a point-to-point link. If this is set to `0.0.0.0`, the IP address of PortServer II's Ethernet interface is used ("unnumbered operation"), unless the remote host uses PPP to negotiate a different address.

NetRouting If set to `off`, routing protocol packets are not sent and are ignored if received. If set to `send`, routing packets are sent, but ignored if received. If set to `receive`, routing packets are not sent, but are acted on if received. If set to `both`, routing packets are both sent and acted on.

Note: No routing packets are sent unless you also turn forwarding on, by entering:

```
set forwarding state=active
```

Protocol Sets the protocol on the connection, and may be `PPP`, `SLIP`, or `FR` (Frame Relay).

Compression If set to `VJ`, Van Jacobsen TCP compression is applied to packets on SLIP and PPP connections. If set to `none`, no compression is applied.

Note: SLIP connections employing Van Jacobsen TCP compression are referred to as Compressed SLIP or CSLIP connections.

VJSlots Sets the maximum number of slots to use for Van Jacobsen header compression (see **Compression** above). Each active TCP transaction on a point-to-point connection requires one VJ slot to hold information about the compression state. If there are more transactions than slots, some compression information is lost.

MTU Maximum Transmit Unit, which is the maximum size of the packet that can cross the WAN interface. For a PPP connection, the maximum packet size is negotiated, and you should enter the largest packet size that PortServer II will permit the remote system to send.

PPPAuth If set to `none`, the remote system does not perform PPP authentication. If set to `PAP`, the remote system is authenticated with PAP. If set to `CHAP`, the remote system is authenticated with CHAP. If set to `both`, the remote system negotiates what authentication is used.

PapID The PAP authentication identifier that PortServer II will send to a remote system if requested. (Any identifier that the remote system returns must match the user name.)

pappasswd	The PAP password that PortServer II will send to a remote system if requested. (Any password that the remote system returns must match the user's password.)
ChapID	The CHAP authentication identifier that PortServer II sends to any remote system that requests it. (Any identifier that the remote system sends must match the user name.)
ChapKey	The CHAP key that PortServer II uses to generate a reply to the remote system.
Passive	If set to `on`, PortServer II waits for the remote system to begin PPP negotiations. If set to `off`, PortServer II may initiate negotiations.
AddrCompress	If set to `on`, PortServer II attempts to negotiate address compression on any PPP connection. If set to `off`, no negotiation is attempted.
ProtoCompress	If set to `on`, PortServer II tries to negotiate protocol compression. If set to `off`, no protocol compression is attempted.

For example, to reconfigure an incoming WAN connection for user `linda`, enter:

```
set user name=linda IpAddr=ippool IpMask=255.255.255.255
LocalIpAddr=0.0.0.0 NetRouting=off Protocol=ppp
Compression=VJ VJSlots=16 MTU=1500 PPPAuth=CHAP
ChapId=password ChapKey=random_numbers Passive=off
AddrCompress=on ProtoCompress=on
```

In this example, linda obtains an address from the IP pool. The remote system is not a router, so the IP mask is **`255.255.255.255`**. The local IP address is set to PortServer II's Ethernet interface. The protocol is PPP, with Van Jacobsen compression on. CHAP authentication will be used, and identifier `password` and the key `randomnumbers` will be used if the remote host authenticates the PortServer II. The remote server will begin PPP negotiations, and Address Compression and Protocol Compression are enabled.

Note: For more information on the `set user` command options, refer to `set user` in the *Command Reference Guide*.

Tip: If the user is dialing in from Windows 95, include the following options:

```
set user name=<username> network ipaddr=on Compression=vj
Passive=on AddrCompress=on ProtoCompress=on
```

4. Optionally, PortServer II may use **logpacket** and **passpacket** filters. **Logpacket** filters define IP packets that are logged for tracing. **Passpacket** filters define IP packets that are passed over the connection. Information on how to create filters is given later in this Chapter. To use these filters, enter the following command:

   `set user name=<username> LogPacket=<name> PassPacket=<name>`

   For example:

   `set user name=linda LogPacket=filter1 PassPacket=filter2`

5. Optionally, you can configure PortServer II to drop the connection if there is no activity for a defined period. To do this, enter the following commands:

   `set user name=<username> IdleTimeout=<timeout>`

   `<timeout>` is the maximum period of inactivity before PortServer II drops the connection. The following example will drop the connection if no packets are detected for 100 seconds:

   `set user name=linda IdleTimeout=100`

6. Optionally, you can configure a **keepup** filter that restricts which packets restart the idle timer. If no keepup filter is specified, all packets restart the Idle Timer. To do this, enter the following command:

   `set user name=<username> Keepup=<filter_name>`

   `<filter_name>` is the name of the `keepup` filter. Information on configuring filters is given later in this Chapter. For example:

   `set user name=linda Keepup=filter3`

Note: The optional parameters in steps 4 and 6 could be combined with the main configuration command in step 3 if you wish.

Verifying the incoming connection

When you have completed configuration of the incoming connection, verify that the information you entered is correct by entering the following command:

`wan verify=<username>`

Check that you are returned directly to the command line prompt. If there are any errors in the configuration, more details are displayed.

Outgoing WAN Connections

How outgoing connections are established

Outgoing WAN connections can be initiated two ways:

1. If the `set user option dialout=on` and no ActivateTime (described later) is defined, the connection is created immediately the user requests it. If a bringup filter is defined, PortServer II waits for the filter to trigger before dialing; if there is no bringup filter, the number is dialed immediately.

   Note: Even if a bringup filter is defined, but the NetRouting parameter is `rec` or `both`, an initial connection is made to attempt to learn about routes from the remote end of the connection.

2. Using the ActivateTime option. Outgoing connections can be initiated at specific times or on certain days using the `set user` basic options command with the `AccessTimes` option. If the current time is out of the specified range, PortServer II waits until the time enters the allowable range before bringing up the connection.

3. Entering the command `wan start=name` immediately brings up the connection specified by `name`.

An outgoing WAN connection remains active until the period during which the connection is permitted expires, an idle timeout or hang-up is detected, or you issue a `wan close` command. Entering the following command also closes any current connections on the named WAN:

```
set wan name=name dialout=off
```

Note: For more information on the `wan` command options, refer to `wan` in the *Command Reference Guide*.

Outgoing connections can also be closed by an **idle timeout**. The idle timer monitors activity on the connection and, if it does not detect IP packets, before the timer delay expires, it disconnects the user. The link is returned to the bringup state, awaiting IP packet activity to re-establish the connection.

How ports are used

When a connection is requested, the outgoing port is opened and initialized using the configured port settings.

You can configure a script which, when executed, changes the port settings to 8 data bits, 1 stop bit, no parity, and no software flow control. The script may then optionally set different port settings for baud rate, flow control, parity and stop bits. The changed baud settings remain in effect after the script has been executed.

SLIP and PPP connections also use 8 data bits, 1 stop bit, and no parity. Software flow control is not used for SLIP; however, you can enable it for PPP with the `set flow` command. Use the default software flow control characters (0x11, 0x13) and the asynchronous transparency set to `000a0000`. Hardware flow control can be used with either SLIP or PPP, and is recommended.

Configuring outgoing connections

1. Create any dialer and login scripts that will be used by your modem or other connection device. Any script **must** be configured before you create the device. See *Chapter 9, Configuring Modem Connections* for detailed information.

   `set device name=vfast dialer=vfastscript ports=4`

2. Configure the PortServer II port that will originate the outgoing connection requests, by entering a command similar to the following:

   `set ports dev=<value> range=<value>`

   `dev=<value>` specifies the device type used for the connection; the most frequently used option is `mio` (modem for both input and output).

   For example:

   `set ports dev=mio range=4`

   Note: The `set ports` command includes options that set autoconnection and modem parameters. For more information on these and additional details on the options described above, refer to `set ports` in the *Command Reference Guide*.

3. Configure the required network options for the user, by entering a command similar to the following:

   `set user name=<username> IpAddr=<ipaddress> IpMask=<ipmask>`
   `LocalIpAddr=<local_ip_address> Device=<device_name>`
   `Dialout=on NetRouting=<option> Protocol=<protocol>`

```
Compression=<type> [VJSlots=<slots>] MTU=<size>
PPPAuth=<type> [PapId=<string>]
[PapPasswd=<password>][ChapId=<string>] [ChapId=<key_value>]
Passive=<on/off> AddrCompress=<on/off>
ProtoCompress=<on/off>
```

The parameters for this command are described below:

IPAddr — If set to an IP address, this specifies the remote host to connect to. If set to `0.0.0.0` or `negotiated`, the host and PortServer II negotiate the address to use; this option is useful if the host address is unknown, for example, if the host system uses an IP address pool.

IPMask — The IP mask that is applied to the address specified in IPAddr. If the mask is not `255.255.255.255`, PortServer II assumes a router is present on the other end of the connection, and the router can forward packets to all other addresses on the network or subnet specified.

LocalIPAdr — Assigns an IP address to the local end of a point-to-point link. If this is set to `0.0.0.0`, the IP address of PortServer II's Ethernet interface is used ("unnumbered operation"), unless the remote host uses PPP to negotiate a different address.

Device — Contains the name of the device or device pool that makes the connection.

Dialout — Always set to `on` to enable outgoing connections.

NetRouting — If set to `off`, routing protocol packets are not sent and are ignored if received. If set to `send`, routing packets are sent, but ignored if received. If set to `receive`, routing packets are not sent, but are acted on if received. If set to `both`, routing packets are both sent and acted on.

Note: No routing packets are sent unless you also turn forwarding on, by entering:

```
set forwarding state=active
```

Protocol — Sets the protocol on the connection, and may be `PPP`, `SLIP`, or `FR` (Frame Relay).

Compression If set to VJ, Van Jacobsen TCP compression is applied to packets on SLIP and PPP connections. If set to none, no compression is applied.

Note: SLIP connections employing Van Jacobsen TCP compression are referred to as Compressed SLIP or CSLIP connections.

VJSlots Sets the maximum number of slots to use for Van Jacobsen header compression (see **Compression** above). Each active TCP transaction on a point-to-point connection requires one VJ slot to hold information about the compression state. If there are more transactions than slots, some compression information is lost.

MTU Maximum Transmit Unit, which is the maximum size of the packet that can cross the WAN interface. For a PPP connection, the maximum packet size is negotiated, and you should enter the largest packet size that PortServer II will permit the remote system to send.

PPPAuth If set to none, the remote system is not authenticated with PPP. If set to PAP, the remote system is authenticated with PAP. If set to CHAP, the remote system is authenticated with CHAP. If set to both, the remote system negotiates what authentication is used.

PapID The PAP authentication identifier that PortServer II will send to a remote system if requested. (Any identifier that the remote system returns must match the user name.)

pappasswd The PAP password that PortServer II will send to a remote system if requested. (Any password that the remote system returns must match the user's password.)

ChapID The CHAP authentication identifier that PortServer II sends to any remote system that requests it. (Any identifier that the remote system sends must match the user name.)

ChapKey The CHAP key that PortServer II uses to generate a reply to the remote system.

Passive If set to on, PortServer II waits for the remote system to begin PPP negotiations. If set to off, PortServer II may initiate negotiations.

AddrCompress — If set to `on`, PortServer II attempts to negotiate address compression on any PPP connection. If set to `off`, no negotiation is attempted.

ProtoCompress — If set to `on`, PortServer II tries to negotiate protocol compression. If set to `off`, no protocol compression is attempted.

For example, to reconfigure an outgoing WAN connection for user `linda`, enter:

```
set user name=linda IpAddr=1.2.3.4 IpMask=255.255.255.255
LocalIpAddr=0.0.0.0 Device=modem_pool Dialout=on
NetRouting=both Protocol=ppp Compression=VJ VJSlots=16
MTU=1500 PPPAuth=CHAP ChapId=password ChapKey=random_numbers
Passive=off AddrCompress=on ProtoCompress=on
```

In this example, the IP address used is **`1.2.3.4`**. The remote system is not a router, so the IP mask is **`255.255.255.255`**. The local IP address is set to PortServer II's Ethernet interface. The connection is made via a device called `modem_pool`. The protocol is PPP, with Van Jacobsen compression `on`. CHAP authentication will be used, and identifier `password` and the key `randomnumbers` will be sent to the remote host. The remote host will begin PPP negotiations, and Address Compression and Protocol Compression are enabled.

Note: For more information on the `set user` command options, refer to `set user` in the *Command Reference Guide*.

4. Optionally, PortServer II may use **logpacket** and **passpacket** filters. **Logpacket** filters define IP packets that are logged for tracing. **Passpacket** filters define IP packets that are passed over the connection. Information on how to create filters is given later in this Chapter. To use these filters, enter the following command:

   ```
   set user name=<username> LogPacket=<name> PassPacket=<name>
   ```

   For example:

   ```
   set user name=linda LogPacket=filter1 PassPacket=filter2
   ```

5. Optionally, you can configure PortServer II to drop the connection if there is no activity for a defined period. To do this, enter the following commands:

   ```
   set user name=<username> IdleTimeout=<timeout>
   ```

   `<timeout>` is the maximum period of inactivity before PortServer II drops the connection. The following example will drop the connection if no packets are detected for 100 seconds:

   ```
   set user name=linda IdleTimer=100
   ```

6. If you configure an **Idle Timer** in the previous step, you should also configure a **bringup** filter. To do this, enter the following command:

   set user name=<username> Bringup=<filter_name>

   <filter_name> is the name of the bringup filter. For example:

   set user name=linda network Bringup=filter3

7. Optionally, you can configure a **keepup** filter that restricts which packets restart the idle timer. To do this, enter the following commands:

   set user name=<username> Keepup=<filter_name>

   <filter_name> is the name of the keepup filter. Information on configuring filters is given later in this Chapter. For example:

   set user name=linda Keepup=filter3

8. You can also restrict connections to certain times or certain days. To do this, enter the following command:

   set user name=<username> AccessTime=<timestring>

   <AccessTime> specifies when the link will be available. For example, if AccessTime=tu, the connection will remain active for 24 hours, but only on Tuesdays. If AccessTime=tu12:00-13:00, then the connection is only available between 12.00 and 13.00 hours on Tuesday. The following example establishes the outgoing connection for linda through the "working" week (Monday through Friday) for 24 hours each day, but drops it at weekends:

   set user name=linda AccessTime==wk

Note: The optional parameters in steps 4 through 8 could be combined with the main configuration command in step 3 if you wish.

Note: If both a bringup filter and the AccessTime option are defined, when AccessTime makes the link active, it also enables the bringup filter. If the filter then detects traffic, it brings up the link. If an AccessTime option is defined, but no filter, the link comes up immediately AccessTime triggers.

9. If the user makes the outgoing connection to a Frame Relay service, configure the Frame Relay parameters by entering the following command:

   set user name=<username> FRPort=<number> FRDLCI=<number>

   - FRPort= is the port or range of ports assigned to the user.
   - FRDLCI= is the local Data Link Channel Identifier assigned by your service provider.

For example:

```
set user name=linda FRport=1 FRdlci=128
```

Note: Other Frame Relay parameters are defined with the `set Frdlci` command. Refer to *Chapter 14, Configuring Frame Relay* for details.

Verifying the outgoing connection

1. When you have completed configuration of the incoming connection, verify that the information you entered is correct by entering the following command:

   ```
   wan verify=<username>
   ```

   If there are no errors, the command line prompt is displayed. If there are any errors in the configuration, details are displayed.

2. If you use timed bring-up filters, reboot PortServer II to establish any outgoing connections. (See *Chapter 17, Troubleshooting* for reboot procedures). PortServer II then scans its User Table for outgoing connections that must be started because of an active `AccessTime` configuration, then establishes those connections.

Bidirectional WAN Connections

You can define a WAN connection that can be both incoming or outgoing. For example, suppose you want to establish a connection between two PortServer IIs on which either detects packets destined for the other. If *PortServer A* detects packets for *PortServer B,* then *A* contacts *B;* if *B* detects packets for *A,* then *B* contacts *A*.

To create a bidirectional connection, use a `set user` command that includes both incoming WAN and outgoing WAN connection parameters, as described above. For further information, see `set user` in the *Command Reference Guide*.

Filters

General

Filters can be used in PPP, SLIP, and CSLIP connections to bringup a connection, to keep the connection up, to pass or block IP packets, and to log IP packets.

Creating a filter

To configure a filter, enter the following command:

```
set filter name=<filtername> [stanza#]
```

Each filter comprises up to 32 stanzas. When a filter is active, stanza s1 is processed first, s2 second, and so on. As soon as a stanza is found that matches the packet being processed, the action specified in that stanza is taken, and the remaining stanzas are ignored. Each stanza may contain several tokens or conditions; all conditions must be true for the stanza to be true.

Important: It is not possible to add a stanza into the middle of a filter that has been completed. You can only add a line to the end, or replace an existing line. Otherwise, you should create a new filter that includes the additional stanza.

Syntax for filter stanzas

Configuring actions that will not *be taken*

The exclamation mark ("!") changes the sense of the filter action. If the packet matches a stanza and the stanza is preceded by "!", then the action is *not* taken, and no other stanzas are processed. Also, if the last stanza is preceded by "!", the action taken on exiting the filter differs to that taken if the "!" is omitted (see the example under *A filter that will bring up a connection when it detects any IP packet except DNS* later in this chapter).

Applying actions to source or destination

If an IP address or port is included in a filter, any match is valid for packets to and from the address or port, unless qualified with `src` (source only) or `dst` (destination only). For example:

```
s1= 199.86.8.33    //Match if IP source or destination address
                   is 199.86.8.33

s1=src/199.86.8.33//Match if IP source address is 199.86.8.33

s1=tcp/60-80/dst   //Match if tcp destination port is in range
                   of 60-80

s1=udp/199.86.8.33/0xffff0000/src //udp source address of
                                    199.86.xxx.xxx
```

Applying actions to inbound or outbound packets

You can use `send` and `rcv` to distinguish between inbound and outbound packets. For example:

```
s1=send/3/icmp     //Match outbound icmp type 3 packet

s1=!rcv/telnet     //Match if not an inbound telnet packet

                   //where telnet defined in services table
```

Applying actions to specific types of packet

You can use `syn` and `fin` as qualifiers for TCP packets. `syn` allows the filtering of packets starting a TCP connection, and `fin` can be used to log the end of a TCP session. For example:

```
s1=send/syn/telnet//outbound, telnet startup ip packet
```

Examples of filters that perform common functions

Building a firewall with passpacket filters

Filters can be defined to selectively pass or block IP packets based on:

- Inbound or outbound packet IP address
- Source or destination IP address
- TCP/UDP port
- Protocol

You can configure **passpacket** filters using any or all of these criteria to build a security firewall between the Internet and a local network.

For example, if your WWW server has an IP address of 199.86.8.33, configure a filter similar to that shown below and call it *filter 1*:

```
s1= 199.86.8.33 //Match if IP source or destination address is
                  199.86.8.33
```

You can then enter a command similar to the following:

```
   set user name=webconnection network PassPacket=filter1
```

This will pass packets that match the WWW server's IP address and block all others.

A filter that will block all except specific ftp packets

The following filter blocks all incoming **ftp** packets except those to host `199.86.8.22` and allows other packets. You must define **ftp** in the Service Table, using the `set service` command:

```
s1=ftp/syn/recv/dst/199.86.8.22//allow incoming ftp with dest
                                   addr of 199.86.8.22
s2=!ftp/syn/recv              //allow all other packets except
                                   incoming ftp
```

A filter that will bring up a connection when it detects IP packets

The following filter brings up a connection when it detects telnet or rlogin IP packets:

```
s1=telnet
s2=rlogin
```

A filter that will bring up a connection when it detects any IP packet except DNS

The following filter brings up a connection for any packet but `dns`. The first stanza is preceded by a "`!`", so the filter brings up the connection when it detects IP packets that are *not* domain packets. The second stanza is also preceded by a "`!`", so TCP packets that *are* not domain packets are passed. All other packets (including DNS packets) are dropped.

```
s1=! 53/udp
s2=! 53/tcp
```

Note: If the exclamation marks were omitted from this filter, it would pass only DNS packets and drop all others.

Tracing messages

A **logpacket** filter can specify packets that will be logged when the `set trace` options are set appropriately (see `set trace` in the *Command Reference Guide* and *Chapter 17, Troubleshooting* for more information).

For example, if want to log all traffic to your WWW server and the server has an IP address of 199.86.8.33, configure a filter similar to that shown below as *filter 2*:

```
s1= /recv/199.86.8.33   //Match if IP destination address is
                        199.86.8.33
```

You can then enter a command similar to the following:

```
set user name=webconnection LogPacket=filter2
```

This will log all packets on the WAN whose user is called *webconnection* that are addressed to the WWW server.

A **PassPacket** filter can specify packets that will be passed over the WAN connection. If you omit a **PassPacket** filter, all packets will be passed, including trace messages.

Chapter 9

Configuring Modem Connections

When should I read this chapter?

Read this chapter before you add a dial-in or dial-out connection to a previously-configured PortServer II. If you have this type of connection to configure, you should read this chapter before you read *Chapter 8, Configuring WAN Connections*.

In this Chapter

This chapter describes how to configure PortServer II to operate over modem links.

The Chapter includes the following topics:

About Modem Connections

If you use a modem or similar device on a connection to a remote host or system, you must configure the serial port to operate with the modem. This Chapter describes how to configure PortServer II for communications between the PortServer II and the modem. You will also have to configure your modem settings for communication with PortServer II and should refer to your modem documentation for details.

Note: Configuring communications by the modem over telephone or data lines is outside the scope of this User's Guide. Refer to your modem documentation for assistance.

You will also need to configure PortServer II for the protocol to use over the modem connection after you have configured the modem. Information on this is given in *Chapter 8, Configuring WAN Connections*.

Configuring your Modem

The configuration procedure for your modem depends on the type of modem and your system arrangement. However, to ensure that the modem operates with PortServer II, set your modem to meet the requirements listed below:

- Make sure signals TD, RD, DCD, DTR and GND are wired in the cable between PortServer II and the modem. Refer to the cabling diagrams in *Chapter 3, Installation*.
- Configure the modem so that DCD goes high when it receives an incoming connection request. For Hayes-compatible modems, this command is `AT &C1`.
- Configure the modem to answer an incoming call only when DTR is high, and to drop the line when DTR goes low. For Hayes-compatible modems, this command is `AT S0=1 &D3`.
- For the best results in bidirectional mode, configure the non-volatile parameters in the modem for incoming calls. Also, configure the modem to reset to these parameters each time DTR is dropped. For Hayes-compatible modems, this command is `AT &D3`.
- PortServer II cannot switch the baud rate of the serial line for different connections. You should configure the modem to lock the serial line speed at the highest baud rate the modem will accept for reliable data transfer.
- Generally, use hardware flow control on modem lines, and set software flow control off.

  Note: UUCP and XMODEM protocols do not work with software flow control.

- Many modems have external or internal jumpers that override the normal operation of DTR, DCD, RTS and CTS. Check the settings of these jumpers on your modem by referring to your modem manual.
- You can watch the LEDs on the PortServer II front panel to monitor correct modem operation. In particular, verify that DCD is off when the modem is not connected. Verify that the modem does *not* answer a call when DTR is low, and hangs up when DTR is dropped.

Configuring the Modem Connection

1. Set the relevant PortServer II serial port to the highest speed that the modem can accept without data corruption, by entering the following command:

   `set line baud=<speed> range=<ports> [options]`

   `<speed>` is the baud rate of the serial port, which can only be set to certain legal values (refer to `set line` in the *Command Reference Guide*). `<ports>` specifies the port(s) you are setting. [`options`] are control and parity settings and must correspond to your modem's settings.

   For example, to set serial port 4 to 2400 baud and tell it to ignore parity errors, enter:

   `set line baud=2400 range=4 error=ignore`

   Note: Remember that this command sets the PortServer II-to-Modem connection, not the speed at which the modem communicates over the telephone lines. If you are uncertain of the best value, we suggest you try the default setting of 9600 baud. If this does not allow reliable data transfer, *lower* this value until you obtain satisfactory results. If the default setting allows reliable transfer, *increase* the value until you obtain the highest setting that allows reliable transfer.

2. If you have not already done so, set the serial port for the modem device type, by entering:

   `set port dev=<modem_type> range=<ports>`

   `<modem_type>` may be `mio` for a bidirectional modem, `min` for an input-only modem, or `mout` for an output-only modem. `<ports>` specifies the port(s) you are setting. For example, to use a bidirectional modem on port 4, enter:

   `set port dev=mio range=4`

Note: For detailed information on the `set port` command, see the *Command Reference Guide.*

3. Set the flow control for the serial port by entering:

   `set flow signals=<state> [options] range=<ports>`

   Using `signals=<state>`, you can set PortServer II to use CTS, DCD, DSR, and/or RI for output data flow control. You can also set it to use DTR and RTS for input data flow control. The [`options`] allow you select software flow control and **altpin** options.

   For example, to use RTS and CTS for flow control on Port 4, enter:

   ```
   set flow rts=on range=4
   set flow cts=on range=4
   ```

   Note: In this example, the modem must be set for hardware flow control. You must also ensure that software flow control is off, by entering:

   ```
   #>set flow ixon=off range=4
   #>set flow ixoff=off range=4
   ```

   Note: For detailed information on the `set flow` command, see the *Command Reference Guide.*

4. Optionally, you can also set PortServer II to run initialization and test scripts, by entering:

   `set modem option=<value> range=<ports>`

   <value> is the name of the script to run. For example, to assign a test script called `test1` to Port 4, enter:

   `set modem test=test1 range=1`

   Note: For detailed information on the `set modem` command, see the *Command Reference Guide*. Information on how to create and edit scripts is given below.

Dialer and Login Scripts

Creating or editing a script

Scripts can dial modems on outgoing connections and login to remote systems. (See *Chapter 8, Configuring WAN Connections*). For example, if you dial in to PortServer II from a PC running Windows 95, you may need a script to work with your dialer program – refer to your PC documentation for details. They can also initialize, and test modems (see above). Scripts are stored in the Scripts Table.

Each script contains one or more strings and each string describes a machine state. You can configure up to 24 different machine states. Each machine state contains a sequence of commands performed in order from left to right. The available commands are listed below.

To create a new script, enter the following command:

```
set script name=<script_name> s{1-24}=statedefinition
```

<script_name> is the name you give to the script. s{1-24} is the number of the machine state, and <statedefinition> is the sequence of commands that corresponds to that machine state. For example, the following script tells the modem to pause for one second, send parameter 2 on the outgoing port, then go to state 5:

```
set script name=waitscript s3="P1 M{%2\r} G5"
```

Important: It is not possible to add a line into the middle of a script that has been completed. You can only add a line to the end, or replace an existing line. Otherwise, you should create a new script that includes the additional line.

Script commands

You can use the following commands in your scripts:

Note: `s` indicates a state parameter. State parameters are described after the list of commands.

A*np*	Sets character size to *n*, and parity according to *p*. The value *n* must be 7 or 8. The value *p* may be `0` (no parity), `1` (odd parity), `2` (even parity), or `3` (mark parity).
B*n*	Transmits a break signal *n* milliseconds in length. If *n* is not specified, transmits a 250 millisecond break.
C*n*	Sets carrier loss detection. If *n* is `0`, carrier loss is not detected. If *n* is `1`, the modem hangs up if the port loses DCD.
E{*string*}	If running interactively, writes *string* to the user's terminal. If running in background, writes *string* to the trace buffer. (`set trace mask=wan:info` must have been previously executed).
F*n*	Pauses for *n* seconds, then flushes input data. *n* is optional and defaults to `0` if you do not enter a value.
G*s*	Immediately transfers control to state *s*.
H*s*	Sets the carrier lost (hangup) recovery state to *s*. If carrier is lost (and detection set with command `C`*n*) transfers to state *s*.
D+*m*	Raises a modem signal. If *m* is `1`, raises DTR. If *m* is `2`, raises RTS.
D-*m*	Lowers a modem signal. If *m* is `1`, drops DTR. If *m* is `2`, drops RTS.
M{*string*}	Writes *string* to the modem.
N*b*	Changes the baud rate to speed *b*.
P*n*	Pauses for *n* seconds. *n* is optional and defaults to 1 second if you do not enter a value.
Q*n*	Sets flow control. If *n* is `0`, disables all regular port flow control. If *n* is 1, enables flow control.
R*s*	Decrements the retry count. If count < 0, switches to state *s*.
S*n*	Declares the timeout used when waiting for a modem signal or input data is *n* seconds.
T*s*	Timeout recovery state. Any timeout waiting for input data, or waiting for a modem signal will transfer to this state.

U*n*	Immediately executes (uses) the text of state definition *n* as if it were inserted to replace this command. You can nest this command, up to a maximum of 10.
W+*m*	Waits for a modem signal to become high. If *m* is 1, waits for DCD high. If *m* is 2, waits for CTS high.
W-*m*	Waits for a modem signal to become low. If *m* is 1, waits for DCD low. If *m* is 2 waits for CTS low.
[*string*]*s*	Declares that a transfer to state *s* should be performed when *string* is received on the communication line.

State parameters

You can use any of the following state parameters in place of *s* in the commands listed above.

n	Transfer to state *n*, where n is a decimal number in the range 1 to 24.
+	Exit, indicating success.
-	Exit, indicating general failure.
*	Exit, indicating remote system is busy.
=	Exit, indicating remote system is down.

PortServer II take action(s) if any of these exit commands are issued. For example, a retry timer is provided for both remote system busy and remote system down exit conditions. Remote system busy causes PortServer II to try the next number in the dialer script.

Escape commands

You can use any of the following escape commands in `E`, `M`, and `[]` command strings:

`^c`	The control character derived by the logical and-ing of ASCII character `c` and octal mask 037. This is the character transmitted by a standard ASCII keyboard when the **Ctrl** key is held down and the character `c` is depressed.
`\c`	Standard C language escapes: `\b` (backspace), `\f` (form feed), `\t` (tab), `\n` (new line), `\\` (backslash) and `\r` (return).
`\`*nnn*	The octal byte value *nnn*.
`\x`*hh*	The hexadecimal byte value *hh*.
`%`*p*	Parameter substitution where *p* is an integer from 1 to 9. Parameters 1-9 are provided by the `set user` parameter options for login scripts or the `set device` options for dialer scripts. The parameter is substituted for the `%`*p* expression.
`%n`	Phone number substitution where the number provided in the User Table is substituted for `%n`
`%N`	Phone number substitution with special character translation. The numbers from the system table are substituted for `%N` after special characters are replaced.

Special characters

The following characters have special meaning in `%`*N* phonenumber strings (see *Escape Commands* above), and are replaced with corresponding strings from the `set chat` command:

`*`	(star) Generates a tone equivalent to dialing * on a touch-tone telephone.
`#`	(pound or hash) Generates a tone equivalent to dialing # on a touch-tone telephone.
`=`	(pause) Pause 2 seconds.
`w`	(wait) Wait for secondary dial tone.
`-`	Completely ignored and not passed to modem.

Running a script

When a script is invoked, the modem initializes to the state specified by commands `C1 A80 H- D+1 D+2 Q0 T- S10` and the script executes beginning at line `s1`.

As each state is entered, previous `[]` command strings are cleared, and execution proceeds from left to right.

If a command is encountered which causes a change of state (for example, `Gs`), any remaining commands in the current state are not performed, and execution resumes in the new state.

Otherwise, when all commands in a state have been executed, and at least one `[]` command has been seen, the communication line is read. Incoming characters are then matched against previously declared `[]` command strings. If a match is found, the corresponding state transfer occurs. If carrier sense was enabled by the `C` command and carrier is lost, the last `H` command is honored. If neither of these events occur within the timeout specified by the last `S` command, a timeout is detected. Action is then taken according to the last `T` command.

Examples of scripts that perform specific functions

A login script

The following script uses parameters `p1` and `p2` as login name and password. The script writes a success or failure message to the log file for each access attempt, using the trace facility (see *Chapter 17, Troubleshooting*):

```
s1="[ogin:]2 S20 T4"      //wait 20 seconds for ogin:
                           /if timeout goto s4;if ogin: goto s2
s2='P2 M{%1\r} P1 [sword:]3 T4" //pause 2 seconds, send param1
                           to //port; wait 1 second for sword:
s3="M{%2\r} G5"           //send param 2 to port,go to state 5
s4="E{login failed} G-"   //Write Fail message, Exit indicating
                          //General failure
s5="E{login complete} G+" //Write Success message, Exit indi
                           //cating Success
```

A script that tries alternate numbers

The following dialer script uses the "*" return code to advance to the next phone number in the System Table, when a busy signal is encountered.

```
s1="M{atdt%n\r} G2"
s2="[BUSY]* [CONNECT]+ S50 T-"
s3="P1 G+"
```

This script exits with:

- "remote busy" when it receives a busy signal.
- "success" when it receives a connect signal.
- "general failure" when there is a timeout.

If "remote busy" is generated, PortServer II tries the next phone number from the System Table in the same script. If there are no more phone numbers, either because all numbers have been tried or there are no alternates, the connection attempt fails.

Note: You can use the `set user` command to configure a `RemoteBusyDelay` timer. If you do this, the script is executed again when the timer expires.

A script that tries the same number multiple times

You can configure a dialer script that repeats each phone number several times before advancing to the next number. For example:

```
s1="M{atdt%n\r} [BUSY]2 [CONNECT]+ s10 T="
s2="P2 R* G1"
```

When a busy signal is received, state `s2` is entered. The script pauses for two seconds, then decrements the retry counter. When the retry counter is negative, the script returns with "remote busy"; otherwise it goes to state `s1` and retries the phone number. If a timeout occurs, the script exits with "remote system down."

The example above tries each phone number in the User Table twice. The initial retry count is equal to the retry number defined in a Chat Table entry, or 1 if a Chat Table entry is not defined. Refer to `set chat` in the *Command Reference Guide* for information on how to set these values.

A script to initialize a Hayes-compatible modem

The following script initializes the modem and sets the data speed for 115.2Kbps. If the modem does not initialize correctly, it prints **setup failed**, otherwise it prints **setup succeeded**.

```
s1="M{at\r} [OK} s3 T4"
s2="M{at$k1&y0&f\r} [OK]3 T4"
s3="M{at13&d3$B11520m1s0=1s2=0s11=50*w\r} [OK]5 T4"
s4="E{setup failed} G-"
s5="E{setup succeeded} G+"
```

A script to test a specific modem

The following script tests a Microcomm DeskPort Fast modem and connecting cable.

Note: If you are using the `altpin` option, temporarily disable it using the `set flow altpin=off port=<number>` command, otherwise the test will fail.

```
s1="C0 F M{at&f\r} [OK]2 S5 T3" //no hangup, flush, set modem to
                                //factory defaults, wait for OK
s2="T4 W-1 T5 W+2 G6"           //wait for DCD low, CTS high
s3="E{Error Condition} G-"      // s1 didn't get OK, or other
                                //error
s4="E{DCD stuck high} G-"
s5="E{CTS stuck low} G-"
s6="M{at\\q0 \\d2 \r} T7 W-2 G8" //no flow control, CTS follows
                                //DCD, detect CTS low
s7="E{CTS stuck high} G-"
s8="M{at &C0 \\d0 \r} T9 W+1 G10"//DCD on, detect DCD high
s9="E{DCD stuck low} G-"
s10="M{at &f\r} {OK}11 S5 T3"  //reset to factory defaults
s11="M{at &d3s0=7so?\r} [7]12 T3" //dtr transition clears
                                //modem, set so=7, read so
s12="D-1 P2 D+1 M{at so?\r} [0] 14 T13"//raise and lower dtr to
                                //clear modem, s0 goes to 0
s13="E{DTR undetected} G-"
s14="M{at &f\r} E{Cable test complete} G+"
```

To run the script, enter the following commands:

```
set port devtype=mout range=port#
set modem test=testmicrocom range=port#
wan testmodem=port#
```

If the test is successful, it will display "Cable test complete".

Modem Pools

As an alternative to assigning a dedicated modem to each connection, you can configure a modem “pool”. Each PortServer II can be configured to look on a range of ports for an available modem.

In the example command below, the PortServer II is configured for two modem pools, one using Ports 1 through 10, and the other Ports 11 and 12. If the remote host’s incoming connection (see *Chapter 8*) is configured to search the Device Table for a modem device called `vfast`, it initially finds `vfasta`. If ports 1–10 are busy or the dialer script fails, it continues the search and finds device `vfastb`.

```
set device name=vfasta dialer=vfastscripta ports=1-10
set device name=vfastb dialer=vfastscriptb ports=11,12
```

Telnet and Modems

This section contains an example of how you can telnet from a terminal that is connected to Port 2 to a modem that is connected to Port 4 on the same PortServer II. You will telnet from the terminal on Port 2 to the modem on Port 4, use "AT" type modem commands to set a fixed DTE interface and baud rate, then call an information service.

The telnet command is:

```
telnet hostname|hostip [tcp port#]
```

`hostname` is PortServer II's *nodename* that you configured when you set up PortServer II's basic configuration (see *Chapter 4, Basic Configuration*). The `tcp port#` parameter is used to access a specific device connected to the PortServer II.

Note: By convention, Digi adds *2000* to the PortServer II's port number to create its TCP port number for telnet. To access a device on port 4, use the port number 2004.

1. Log into the terminal on Port 2, then connect to the modem installed on Port 4 by entering the telnet command:

   ```
   telnet termserver 2004
   ```

   (`termserver` is the PortServer II's node name in this example)

   You are now connected directly to the modem, and can enter commands to the modem (including "AT" commands to Hayes or Hayes-compatible modems).

2. Set the modem for RTS/CTS hardware handshaking on the PortServer II to modem connection, by entering:

   ```
   AT&K3
   ```

   You can enter other "AT" commands to configure the modem in the same way.

   Note: You are connected directly to the modem, so there is no command prompt.

3. Set the modem baud rate. The exact command depends upon the brand and model of modem; refer to your modem documentation for information.

4. Connect to the on-line information service by instructing the modem to dial its telephone number:

   ```
   ATDT 123-4567
   ```

In this example, `123-4567` above is the telephone number of the service.

5. Log off from the on-line service, close the connection to the modem, and log off from the terminal by entering:

```
^] <Carriage Return>
close
exit
```

Configuring CU and UUCP to dial out

Note: As an alternative to the method described below, you may be able to use RealPort, as described in *Chapter 11*.

If you want to use CU or UUCP on your UNIX system without RealPort, you should install RTTY. RTTY works by associating a pseudo TTY port with a particular port or port "group" on the PortServer II. The pseudo TTY operates similarly to a regular port, allowing you to use kermit, UUCP, CU, and similar programs.

You can obtain a copy of RTTY from the Digi International FTP server or BBS. (Refer to *Contacting Digi* in Chapter 18 of this User's Guide for numbers and protocols). The source code is available to customers, and you can also download binaries for most popular operating systems.

RTTY is provided on an "as-is" basis by Digi International, but it is not covered by our unlimited technical support policy. Digi provides the source, internal documentation in the source, and an unlimited distribution license when used with our products.

RTTY does not work with STREAMS-based pseudo TTYs. Also, pseudo TTYs associated with "clone" devices normally must be opened by the clone device. However, even systems with clone devices often maintain BSD-style devices.

We recommend that you read your UNIX man pages to find out what your system supports. If your system supports BSD-style pseudo TTYs, you can display the devices available by typing:

```
ls /dev/pty*
```

Description of operation

The pseudo TTY driver will not allow access to the “master” side of a pseudo TTY by more than one program at a time, so once RTTY accesses a pseudo TTY, it retains control and excludes other programs.

To drop a connection, RTTY must close the device and then reopen it. Although the device is closed for only a few milliseconds, it is still possible for another program to gain control of the device before RTTY can resecure it. If that happens, CU and UUCP may not operate correctly. We recommend that you select a high port number to minimize problems.

Several computer systems on a network can be configured with pseudo-TTYs associated with the same port (or port group) on a PortServer II. This is possible because RTTY only attempts a connection when data is written to the pseudo-TTY. If the PortServer II port is available, the connection is made and the program accessing the port can continue. If the port is already in use by another system, the connection fail (RTTY signals the application by momentarily closing and re-opening the pseudo-TTY port). This method of operation is compatible with UUCP and CU on most systems.

Configuring your system

1. If, for example, you found `/dev/pty[pqrs][0-f]` on your system (64 ptys), and you want to configure four modem ports, you could use `/dev/ptys[cdef]` for your pseudo-modem ports.

   If you want to use PortServer II ports 1-4, add the lines shown below to your “rc” startup script. The parameter “`1`” adds a one second delay after connection before sending data. (This ensures DTR high is established before the modem looks for “AT” commands). The “`d`” parameter runs the program as a daemon, and the “`s`” parameter suppresses the Copyright Notice during bootup.

   ```
   rtty -1ds /dev/ttysc dbps-nodename 2101
   rtty -1ds /dev/ttysd dbps-nodename 2102
   rtty -1ds /dev/ttyse dbps-nodename 2103
   rtty -1ds /dev/ttysf dbps-nodename 2104
   ```

   Note: Use **`21xx`** raw connect port numbers, not **`20xx`** series telnet connect numbers.

2. Add the modems to your UUCP Devices file, as shown below. On some systems, you can omit the `hayes` keyword; on other systems, you must replace it with the name of the dialer script your modem uses.

Note: The actual baud rates you enter are ignored by the pseudo-TTY device driver, but they must match the values in your Systems file.

```
ACU ttysc ttysc 38400 hayes
ACU ttysd ttysd 38400 hayes
ACU ttyse ttyse 38400 hayes
ACU ttysf ttysf 38400 hayes
```

3. The modems now appear directly connected to CU and UUCP. Test your configuration by typing a command similar to the following example:

```
cu -l /dev/ttysc -b 38400 555-1212
```

RTTY program

The following is an extract from the RTTY source code and explains the command usage.

```
/*********************************************************
 * NAME
 * rtty - Connect a tty to a remote TCP port.
 *
 * SYNOPSIS
 * rtty [-dhw] tty host port
 *
 * DESCRIPTION
 *rtty attaches the master side of a named pseudo tty
 *to a TCP session.
 *
 *This is most often useful to allow a TCP terminal server
 *port to appear as a local tty on some host computer.
 *
 *-[0-9] Wait for data to be written to the slave side of
 *  the pseudo tty before opening the connection;
 *  open the connection and sleep the given number
 *  seconds [0-9] before writing the data to the port.
 *
 *-dDaemonize. A detached child process is spawned
 *  to perform the program function, ignoring all
 *  signals.
 *
 *-hHold the tty open so "stty" settings are not
 *  disrupted and EOF's are not sent when the tty
 *  is locally closed.
 *
 *-qQuit after one session is complete. Normally the
 *  program loops to handle multiple sessions.
 *
 *-sSuppress the copyright notice.
 *
 *-wWait for data to be written to the tty port before
 *  executing the rsh command.
 *
 *-xOutput debugging information. Specifying "x"
 *  twice produces even more output.
 *
 *
 *To associate "ttypf" with the dedicated printer device
 *connected to (PortServer II) node ncx, port 4.
 *
 * rtty -dh ttypf ncx 2104
 *
 *To associate "ttysf" with a dial-in/dial-out (type mio)
 *modem attached to (PortServer II) node dbps port 13.
 *
 * rtty -1d ttysf dbps 2113
 */
```

```
static char *copyright[] =
{
 "@(#)Copyright 1992, Digi International, All Rights Reserved.",
 "@(#)An unlimited use and distribution license is granted for use with,",
 "@(#)and only with, Digi terminal servers and other network products.",
  0
} ;
```

Chapter 10

Configuring TCP/IP Routing

When should I read this Chapter?

Read this Chapter if you are adding TCP/IP routing to a previously configured PortServer II, or if you want to modify an existing route.

In this Chapter

This Chapter describes how to configure the different types of routing that PortServer II supports.

The Chapter includes the following topics:

Types Of Routing Available

You can configure PortServer II to act as an IP packet router (see *Chapter 1* for a typical system configuration, in which branch office users are connected to an Internet service provider).

PortServer II supports two types of routing, Passive and Active. With passive routing, it takes packets in from hosts and routers connected to it, and decides where to forward them. During active routing, it performs passive routing, but also informs other systems about destinations they can reach by forwarding packets to PortServer II.

PortServer II uses three sources of information to create a Route Table, which decides where packets are forwarded. You can view the contents of the Route Table by entering the `set route` command with no options.

The first source is the configuration information you enter for each interface. The address of the Ethernet interface is used with any subnet mask to determine which destination addresses are reachable through that interface. For example, if the Ethernet interface has the address 192.83.159.15 and a subnet mask set to 255.255.255.0, any IP address from 192.83.159.0 through 192.83.159.255 can be reached through the Ethernet interface. For WAN connections, the IP address of the remote system and any subnet mask determine that addresses can be reached through the WAN connection. For example, if a WAN connection has a remote IP address of 192.83.159.17 and a subnet mask of 255.255.255.255, PortServer II assumes it can only reach IP address 192.83.159.17 through the WAN interface. However, if the subnet mask is 255.255.255.240, PortServer II can reach IP addresses 192.83.159.16 through 192.83.159.31 through the WAN interface.

The second source of routing information is static routes. PortServer II can be manually configured with static routes using the **set route** command. For example, if the **set route** command is entered with net=192.83.160.1, mask=255.255.255.255, and gateway=192.83.159.17, PortServer II will route packets destined for IP addresses 192.83.160.1 through 192.83.159.17. PortServer II allows dependency routes, in which routes associated with the WAN may not be permanently established. While the WAN connection is present, the route is considered active and is used to route packets. When the WAN goes inactive, the route is generally ignored.

Note: For routing purposes, a dial-out or bi-directional WAN connection that is set to connect on network traffic by a bringup filter is considered always active.

The third source of routing information is routes that other routers have told PortServer II about, using RIP (Routing Information Protocol).

During passive routing, there are two ways by which PortServer II can inform other systems about routes that pass through its interfaces.

One is to send RIP (Routing Information Protocol) packets to other systems. RIP packets are only sent to other systems if you configure **active** forwarding.

The second method is by means of the Proxy ARP (Address Resolution Protocol), which determines a route to an unknown IP address via a known IP address on the same subnet. If proxy ARP is used, PortServer II can respond to ARP requests for addresses if it knows of a route to the subnet's IP address through any of its WAN interfaces or the Ethernet interface. For example, if a dial-in user has an address on the same subnet as PortServer II, hosts on the same Ethernet interface as PortServer II can "ARP" for the IP address of the dial-in user, and PortServer II will respond to the ARP request with its Ethernet address.

By default, PortServer II provides static routing transferring packets to and from a single device connected to a serial port.

Description of Passive routing

PortServer II routes packets through the Ethernet connection and serial ports. It routes default routes, static routes, and routes learned from other routes, but does not propagate these routes. The proxy ARP may be used to allow devices on the Ethernet to communicate with devices on PortServer II's serial ports. The proxy ARP handles routing tasks such as the conversion of IP addresses to Ethernet addresses.

Description of Active routing

PortServer II learns the routes being propagated, stores them in the Route Table, and propagates static routes, passive routes, and routes learned from other routers. PortServer II sets and propagates dynamic Route Table entries using RIP. Three timers determine how these entries are manipulated - an **advertise** timer, a **timeout** timer, and an internal timer. The **advertise** timer controls how frequently RIP packets are broadcast to advertise the Route Table state. The **timeout** timer controls how long an entry remains in the Route Table if PortServer II does not receive an update from the route. If it is not updated in the set time, the entry is marked as invalid and, after a further two minutes, removed by the internal timer.

Note: Each WAN connection can function differently, depending on whether it is configured to send, listen, or both:

Table 3: WAN Functions

Forwarding:	None	Send	Listen	Both
Active:	No activity	RIP packets sent	RIP packets received	RIP packets sent and received
Passive:	No activity	No activity	RIP packets received	RIP packets received

Passive Routing

To configure a PortServer II port for passive routing, proceed as follows:

1. Configure a WAN connection, as described in *Chapter 8, Configuring WAN Connections*.

2. If static routes are needed beyond those contained in the WAN connection, configure each route by entering a command in the format:

   `set route net=<network_address> mask=<ipmask> gateway=<ip_address> metric=<numhops>`

   `<network_address>` is the address of the network or host that PortServer II will route IP packets to. `<ipmask>` is the subnet mask, if used, or `255.255.255.255` if you are configuring a route to a single host. `<ipaddress>` is the IP address of the gateway that IP packets covered by this routing entry should be forwarded to. `<numhops>` is the number of hops (gateways) between PortServer II's network and the final destination.

   For example,

   `set route net=192.83.159.65 mask=255.255.255.192 gateway=192.83.160.1 metric=3`

3. Set forwarding to **passive**, by entering the following command:

   `set forwarding state=passive [proxyarp=<state>]`

   `<state>` should be `on` if you are using the Proxy ARP, otherwise `off`.

Active Routing

To configure a PortServer II port for active routing, proceed as follows:

1. Configure a WAN connection, as described in *Chapter 8, Configuring WAN Connections*.

2. If static routes are needed beyond those contained in the WAN connection, configure each route by entering a command in the format:

   `set route net=<network_address> mask=<ipmask> gateway=<ip_address> metric=<numhops>`

   `<network_address>` is the address of the network or host that PortServer II will route IP packets to. `<ipmask>` is the subnet mask, if used, or `255.255.255.255` if you are configuring a route to a single host. `<ipaddress>` is the IP address of the gateway that IP packets covered by this routing entry should be forwarded to. `<numhops>` is the number of hops (gateways) between PortServer II's network and the final destination.

   For example,

   `set route net=192.83.159.2 mask=255.255.255.128 gateway=192.83.160.1 metric=3`

3. Set forwarding to **active**, by entering the following command:

   `set forwarding state=active [proxyarp=<state>]`

   `<state>` should be `on` if you are using the Proxy ARP, otherwise `off`.

   For example:

   `set forwarding state=active advertise=30 timeout=180`

   Note: `set forwarding` allows you to set ICMP Router Discovery packet parameters. For information on this and other `set forwarding` options, refer to the *Command Reference Guide*.

4. Check that the Route Table is displayed and that new routes are added periodically as they are received from other routers.

Chapter 11

Configuring RealPort Connections

When should I read this Chapter?

Read this Chapter if you are connecting a terminal, printer, modem, or serial line to PortServer II and the associated host accesses the device using Digi International's RealPort protocol, or if you want to modify an existing RealPort connection.

In this Chapter

This Chapter describes how to configure a connection to a device when the associated host uses the RealPort protocol.

The Chapter includes the following topics:

RealPort Basics

RealPort is a protocol developed by Digi International that, when used in conjunction with device driver software on a host operating system, allows ports on the PortServer II to be used as if they were connected directly to the host system. RealPort supports all standard host system TTY interfaces, so that any system utility or user application that works with a local serial port works immediately with RealPort.

One advantage that RealPort protocol has over TCP/IP protocols such as telnet is the host operating system can directly change port parameters, such as baud rate, hardware flow control and software flow control in the same way as it changes these settings on a local port.

The ports on the PortServer II may be split among hosts; for example, one host can use the odd-numbered ports while another is using the even numbered ports. RealPort also allows one host to handle incoming modem calls on a port, while allowing other hosts to place outgoing calls if there is no current incoming call. (This feature must be supported by the host driver software.)

You must have the correct RealPort driver installed on each of your host systems to take advantage of RealPort protocol.

RealPort drivers include a transparent print feature called DigiPRINT and a multiple screen utility called DigiSCREEN. You should refer to the *RealPort Device Driver Software Manual* for your operating system for full set-up and user instructions for these features.

Configuring PortServer II for RealPort Operation

To configure a PortServer II port to connect a “real” TTY to a host running RealPort:

1. Set authorization to allow the host running RealPort access to the serial port, by entering a command in the following format:

   `set auth ip=<ipaddress> mask=<ip_mask> realport=<range>`

   `<ipaddress>` is the IP address of the host, <ipmask> is the subnet mask, and `<range>` is the port number or range of numbers.

   For example, to allow only host `198.83.159.2` to use RealPort with the devices connected to ports 1 and 2 on PortServer II, enter:

   `set auth ip=198.83.159.2 mask=255.255.255.255 realport=1-2`

   To allow all the hosts on the same network as the host in the previous example to use RealPort on the same ports, enter:

   `set auth ip=198.83.159.0 mask=255.255.255.0 realport=1-2`

   To allow all hosts on any network to connect, enter:

   `set auth ip=0.0.0.0 mask=0.0.0.0 realport=1-2`

   Note: For more information on `set auth` options, refer to the *Command Reference Guide*.

2. If necessary because of conflicts, change the RealPort TCP port number, by entering a command in the format:

   `set config realport=<tcp_port_number>`

   Important: The RealPort TCP port number (`<tcp_port_number>`) has a default value of 771. If you have to change this value because of conflicts, you must change the port number used by the RealPort drivers to the new value. Refer to the *RealPort Device Driver Software Manual* for your operating system for more information.

3. Set the device type for the port, by entering a command in the format:

   `set port dev=<type>`

   If the port has a three-wire connection that does not need modem signals, set the device type set to `prn`. (The RealPort driver will see the actual state of the modem signals regardless of the device type setting.)

If the port is used with a modem, set the device type set to `mout` if it is used exclusively for RealPort connections, or to `mio` if you want incoming calls to go directly to PortServer II's normal login interface. For other types of connection that need modem signals, set the device type to `host`. All of these device types ensure that RTS and DTR modem signals go low and stay low for at least two seconds between connections, to allow modems and other devices to reset properly. For information on configuring modems, refer to *Chapter 9, Configuring Modem Connections*.

Important: Do not use device types of `term` or `min` for RealPort devices.

Chapter 12

Configuring SNMP

When should I read this Chapter?

Read this Chapter if your system supports SNMP (Standard Network Management Protocol) and you want PortServer II to respond to SNMP requests.

In This Chapter

This Chapter describes how to configure PortServer II's SNMP Agent.

The Chapter includes the following topics:

General

PortServer II includes a Simple Network Management Protocol (SNMP) agent. This agent implements the standard Management Information Base II (MIB-II) as defined in RFC 1213. In addition, the agent also supports character-based MIB (RFC 1316) and RS232-like MIB (RFC1317). The SNMP agent provides information, for example on error rates, to the SNMP host.

The SNMP agent supports the **Get**, **GetNext**, **Set**, and **Trap** messages as defined in RFC 1157. These messages are used as follows:

Get The SNMP host uses this message to retrieve the value of a specific object from one of the MIBs supported by PortServer II.

GetNext The SNMP host uses this message to retrieve the value of the object following a specific object in the MIB list. This allows a management station to traverse or “walk” the MIB variables.

Set The SNMP host uses this message to modifies the value of a MIB object. PortServer II does not allow sets.

Trap PortServer II uses this message to asynchronously report a significant event to the SNMP host.

A list of specific variables that PortServer II supports for each MIB is given at the end of this chapter.

Configuring the SNMP Agent

To configure the SNMP Agent on a PortServer II, enter a command in the following format:

```
snmp [run=off|on] [auth_trap=off|on] [trap_dest=<ipaddress>]
[location=<portserver_id>] [snmp_name=<portserver_name>]
[snmp_contact=<administrator_name>]
```

- If you set `run=off`, the SNMP daemon is stopped and all SNMP reporting for the PortServer II is disabled. If you set `run=on`, the SNMP daemon is started and SNMP reporting is active.
- If you set `auth_trap` to off, PortServer II silently ignores SNMP requests that fail authentication. If you set `auth_trap` to on, it sends an authentication trap whenever an authentication error occurs.
- `<portserver_id>` is a text string that describes the location or identity of the PortServer II in a format that is meaningful to the administrator. Put quotation marks around the entry if it includes spaces.
- `<portserver_name>` is the name of the PortServer II and is normally the same as its network name or address.
- `<administrator_name>` is the name of the person who is responsible for administering the PortServer II.

For example:

```
snmp run=on auth_trap=off trap_dest=199.99.88.1
location="portserver #101" snmp_name=blaze
snmp_contact="bill jones"
```

Monitoring SNMP status on PortServer II

To check the current SNMP agent configuration at any time, enter the following command:

```
snmp
```

This prints (displays) the current SNMP configuration. For example, if we check the configuration after entering the command in the previous section, we should see the following display:

```
#> snmp


snmp status


snmp daemon running: on
authentication traps being sent: off
trap destination: 199.99.88.1
syslocation: portserver #101
sysname: blaze
syscontact: bill jones
```

In this example, the SNMP daemon is running (`on`), so PortServer II will answer SNMP requests from a host or management station. Authentication traps are not being sent (`off`), but other standard traps are being sent to IP address 199.99.88.1. The values for the System portion of MIB II are set to identify the PortServer II, its name, and the name of the responsible administrator.

Supported SMNP Variables

System MIB	Interfaces MIB	IP MIB	ICMP MIB
sysDescr	ifNumber	ipForwarding	icmpInMsgs
sysObjectID	ifIndex	ipDefaultTTL	icmpInErrors
sysUpTime	ifDescr	ipInReceives	icmpInDestUnreachs
sysContact	ifType	ipInHdrErrors	icmpInTimeExcds
sysName	ifMtu	ipInAddrErrors	icmpInParmProbs
sysLocation	ifSpeed	ipForwDatagrams	icmpInSrcQuenchs
sysServices	ifPhysAddress	ipInUnknownProtos	icmpInRedirects
	ifAdminStatus	ipInDiscards	icmpInEchos
	ifOperStatus	ipInDelivers	icmpInEchoReps
	ifLastChange	ipOutRequests	icmpInTimestamps
	ifInOctets	ipOutDiscards	icmpInTimestampReps
	ifInUcastPkts	ipOutNoRoutes	icmpInAddrMasks
	ifInNUcastPkts	ipReasmTimeout	icmpInAddrMaskReps
	ifInDiscards	ipReasmReqds	icmpOutMsgs
	ifInErrors	ipReasmOKs	icmpOutErrors
	ifInUnknownProtos	ipReasmFails	icmpOutDestUnreachs
	ifOutOctets	ipFragOKs	icmpOutTimeExcds
	ifOutUcastPkts	ipFragFails	icmpOutParmProbs
	ifOutNUcastPkts	ipFragCreates	icmpOutSrcQuenchs
	ifOutDiscards	ipAdEntAddr	icmpOutRedirects
	ifOutErrors	ipAdEntIfIndex	icmpOutEchos
	ifOutQLen	ipAdEntNetMask	icmpOutEchoReps
	ifSpecific	ipAdEntBcastAddr	icmpOutTimestamps
		ipAdEntReasmMaxSize	icmpOutTimestampReps
		ipRouteDest	icmpOutAddrMasks
		ipRouteIfIndex	icmpOutAddrMaskReps
		ipRouteMetric1	
		ipRouteMetric2	
		ipRouteMetric3	
		ipRouteMetric4	
		ipRouteNextHop	
		ipRouteType	
		ipRouteProto	
		ipRouteAge	
		ipRouteMask	
		ipRouteMetric5	
		ipRouteInfo	

System MIB	Interfaces MIB	IP MIB	ICMP MIB
		ipNetToMediaIfIndex	
		ipNetToMediaPhysAddress	
		ipNetToMediaNetAddress	
		ipNetToMediaType	
		ipRoutingDiscards	

Frame Relay DTE MIB	Address Translation MIB
frDlcmiIfIndex	atIfIndex
frDlcmiState	atPhysAddress
frDlcmiAddress	atNetAddress
frDlcmiAddressLen	
frDlcmiPollingInterval	
frDlcmiFullEnquiryInterval	
frDlcmiErrorThreshold	
frDlcmiMonitoredEvents	
frDlcmiMaxSupportedVCs	
frDlcmiMulticast	
frCircuitIfIndex	
frCircuitDlci	
frCircuitState	
frCircuitReceivedFECNs	
frCircuitReceivedBECNs	
frCircuitSentFrames	
frCircuitSentOctets	
frCircuitReceivedFrames	
frCircuitReceivedOctets	
frCircuitCreationTime	
frCircuitLastTimeChange	
frCircuitCommittedBurst	
frCircuitExcessBurst	
frCircuitThroughput	
frErrIfIndex	
frErrType	
frErrData	
frErrTime	
frTrapState	
frDLCIStatusChange	

TCP MIB	UDP MIB	RS232 MIB	SNMP MIB	Character MIB
tcpRtoAlgorithm	udpInDatagrams	rs232Number	snmpInPkts	charNumber
tcpRtoMin	udpNoPorts	rs232PortIndex	snmpOutPkts	charPortIndex
tcpRtoMax	udpInErrors	rs232PortType	snmpInBadVersions	charPortName
tcpMaxConn	udpOutDatagrams	rs232PortInSigNumber	snmpInBadCommunityNames	charPortType
tcpActiveOpens	udpLocalAddress	rs232PortOutSigNumber	snmpInBadCommunityUses	charPortHardware
tcpPassiveOpens	udpLocalPort	rs232PortInSpeed	snmpInASNParseErrs	charPortReset
tcpAttemptFails		rs232PortOutSpeed	snmpInTooBigs	charPortAdminStatus
tcpEstabResets		rs232AsyncPortIndex	snmpInNoSuchNames	charPortOperStatus
tcpCurrEstab		rs232AsyncPortBits	snmpInBadValues	charPortLastChange
tcpInSegs		rs232AsyncPortStopBits	snmpInReadOnlys	charPortInFlowType
tcpOutSegs		rs232AsyncPortParity	snmpInGenErrs	charPortOutFlowType
tcpRetransSegs		rs232AsyncPortAutobaud	snmpInTotalReqVars	charPortInFlowState
tcpConnState		rs232AsyncPortParityErrs	snmpInTotalSetVars	charPortOutFlowState
tcpConnLocalAddress		rs232AsyncPortFramingErrs	snmpInGetRequests	charPortInCharacters
tcpConnLocalPort		rs232AsyncPortOverrunErrs	snmpInGetNexts	charPortOutCharacters
tcpConnRemAddress		rs232SyncPortIndex	snmpInSetRequests	charPortAdminOrigin
tcpConnRemPort		rs232SyncPortClockSource	snmpInGetResponses	charPortSessionMaximum
tcpInErrs		rs232SyncPortFrameCheckErrs	snmpInTraps	charPortSessionNumber
tcpOutRsts		rs232SyncPortTransmitUnderrunErrs	snmpOutTooBigs	charPortSessionIndex
		rs232SyncPortReceiveOverrunErrs	snmpOutNoSuchNames	charSessPortIndex
		rs232SyncPortInterruptedFrames	snmpOutBadValues	charSessIndex
		rs232SyncPortAbortedFrames	snmpOutGenErrs	charSessKill
		rs232InSigPortIndex	snmpOutGetRequests	charSessState
		rs232InSigName	snmpOutGetNexts	charSessProtocol
		rs232InSigState	snmpOutSetRequests	charSessOperOrigin
		rs232InSigChanges	snmpOutGetResponses	charSessInCharacters
		rs232OutSigPortIndex	snmpOutTraps	charSessOutCharacters
		rs232OutSigName	snmpEnableAuthenTraps	charSessConnectionId
		rs232OutSigState		charSessStartTime
		rs232OutSigChanges		

Chapter 13

Configuring Printer Connections

When should I read this Chapter?

Read this Chapter if you want to configure a printer that you have connected to a PortServer II, or if you want to modify an existing printer connection.

In this Chapter

This Chapter describes how to configure a serial port to use with a printer.

The Chapter includes the following topics:

General

Before you configure a printer that you have connected to PortServer II, check the printer documentation and determine if the printer uses software flow control or hardware flow control. If you set flow control incorrectly, the printer may not print all data that is sent to it.

Note: Certain printers do not use flow control. If you have one of these printers, you can ignore the `set flow` command described below.

Note: Parallel printers normally function correctly with PortServer II's default flow control settings.

Most printers that use hardware flow control issue the DTR (Data Terminal Ready) signal when they are ready for data. If so, the DTR line from the printer must be wired to an input to the PortServer II (usually CTS or DCD) that can be used for flow control. Some printers use Printer Busy (pin 11) for output flow control. Refer to *Chapter 3, Installation* and your printer documentation for more information.

Configuring a Printer Connection

1. Set the speed and parameters of the serial port that the printer is connected to, by entering a command in the format:

   `set line baud=<baudrate> csize=<value> onlcr=<option> otab=<option> parity=<option> range=<port_numbers> stopb=<value>`

   - `baud` is the baud rate for the port.
   - `csize` is the characters size, and may be `5`, `6`, `7`, or `8` characters.
   - `onlcr` may be `on` to convert Line Feed characters to Carriage Return & Line Feed characters, otherwise `off`.
   - `otab` may be `on` to convert output tabs to eight spaces, otherwise `off`.
   - `parity` may be `e` for even parity, `o` for odd parity, or `n` for no parity checking.
   - `range` identifies the port or ports you are configuring.
   - `stopb` may be `1` or `2` to set the number of stop bits.

   Note: The default port settings for printing are 9600 baud, 8 data, 1 stop, and no parity. If you are uncertain of your printer's requirements, try these settings first.

For example, to reset Port 3 to the default values identified in the note above, enter:

```
set line baud=9600 csize=8 stopb=1 parity=n range=3
```

Note: For more information on the `set line` command, refer to the *Command Reference Guide*.

Alternatively, you can enter the following command to reset the port to the factory default:

```
kill tty =3 action=eewrite
```

2. Set the port for a printer, by entering a command in the format:

   ```
   set port dev=prn range=<port_numbers>
   ```

   `<port_numbers>` specifies the port or ports you are configuring. For example, to configure port 3 for a printer, enter:

   ```
   set port dev=prn range=3
   ```

3. Set the flow control required by the printer, by entering a command in the format:

   ```
   set flow aixon=<option> altpin=<option> cts=<option>
   dcd=<option> dsr=<option> dtr=<option> itoss=<option>
   ixoff=<option> ixany=<option> ixon=<option> ri=<option>
   rts=<option> range <port_numbers>
   ```

   In each case, `<option>` may be either `on` to use the associated signal or function for flow control, or `off` otherwise. `<port_numbers>` specifies the port or ports you are configuring. For example, to configure Port 3 for hardware flow control for *output* data and software flow control off, enter:

   ```
   set flow cts=on ixon=off ixoff=off range=3
   ```

   CTS on PortServer II must be connected to DTR on the printer in this configuration.

   Important: If flow control is necessary, set the printer's flow control to match the flow control on PortServer II.

   Note: For more information on `set flow` options, refer to the *Command Reference Guide*.

Printing a File using telnet or rsh

The following section shows examples of commands you can use to print a file on a printer that is connected to Port 1 on the PortServer:

Using telnet

```
pr myfile|telnet ncx 2001
```

Using rsh

```
pr myfile|rsh ncx 1
```

Troubleshooting

- If you are not getting the proper line feed after a carriage return, turn on `onlcr` (converts Carriage Return to Carriage Return & Line Feed). The command is `set line onlcr=on`.
- A tab (ASCII character 9) can be converted to eight spaces. Use `set line otab=on` if you want tabs converted.
- Some versions of telnet can cut off the end of print jobs.

Note: If telnet cuts off print jobs, try **rsh**. (rsh is the name of the command in BSD Unix. On some systems, the command is called **rcmd** (for example, SCO Unix) or **remsh** (for example, HP-UX) to eliminate confusion with the restricted shell called **rsh**.

Printing using lpd Protocol

Configuring the printer

To use the **lpd** protocol to print to a printer connected to PortServer II, configure the print spooler on your system to print to a remote printer. Give the name or IP address of PortServer II as the remote system's name. Use the remote printer's queue name to specify what PortServer II port to print to, and how to print, as follows:

Note: Refer to your operating system documentation for information on how to access the print queue.

1. The printer queue name for PortServer II begins with either `ascii` or `raw`. If the name begins with `ascii`, PortServer II substitutes Carriage Return & Line Feed character pairs for each Line Feed the system sends (this substitution is needed with many UNIX systems). If the name begins with `raw,` no substitution is performed.

2. After the queue name, insert an underscore character and the port number to print to.

3. Finally, add another underscore and any options, in the form of single letters. Two option letters are available:

   `f` - appends a Form Feed character to the end of each file in a print job.

   `d` - adds <Ctrl–d> is added to the end of each file in a print job. (This is often required by PostScript printers.)

Queue name examples

ascii_8_f	This queue prints on a printer connected to Port 8, translates Carriage Returns to Carriage Return/Line Feed pairs, and prints a Form Feed at the end of the job.
raw_10_d	This queue prints on a printer connected to Port 10, with no Carriage Return/Line Feed translation, and appends <Ctrl-D> to the end of the print stream.
ascii_1	Prints on a printer connected to Port 1, translates Carriage returns to Carriage Return/Line Feed pairs, and adds no extra characters to the end of the print stream.

An example entry into the BSD style `/etc/printcap` file, with the PortServer II named `nx5` and using `ascii_1_f` as the print queue name is shown below:

```
lpi|portserver line printer port 1:\

:lp=:rm=nx5:rp=ascii_1_f:sd=/usr/spool/lpd/nx5:lf=/usr/adm/lpd-errs:
```

Note: The number of copies option with lpr (lpr-#<*number of copies*>) is not supported.

Note: Banner pages are not supported.

Chapter 14

Configuring Frame Relay

When should I read this Chapter?

Read this Chapter if you want to use a Frame Relay service with PortServer II.

In This Chapter

This Chapter describes how to configure a serial port for Frame Relay service. The port must have already been configured as a WAN connection.

The Chapter includes the following topics:

What is Frame Relay?

Frame Relay was adapted from the Link Access Protocol D-Channel (LAPD) of the ISDN model. It is a switched digital service, in which several **virtual** or **logical** circuits share a common **physical** circuit. Each physical circuit connects to a Frame Relay service provided by the service provider; this service is sometimes referred to as the "cloud". This arrangement allows a server to make a "logical" connection to several other servers that are also connected to the service, but utilizing only one physical circuit.

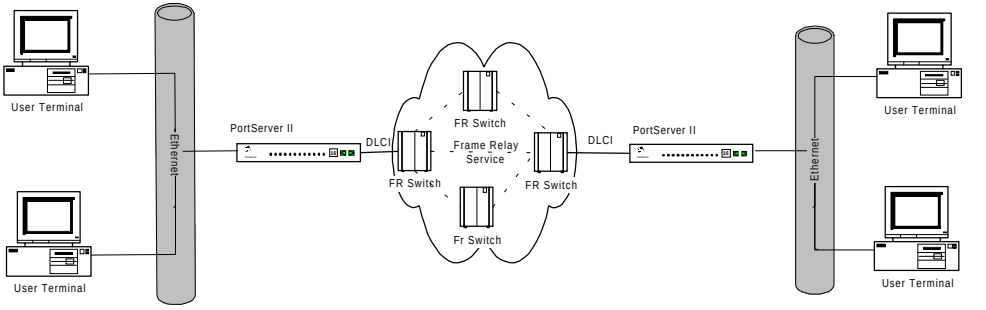

Figure 21 Typical Frame Relay Configuration

Each logical connection is referred to as a Permanent Virtual Circuit (PVC). Each PVC has a Data Link Channel Identifier (DLCI) associated with it. The DLCI is used locally to identify the connection, but does not traverse the network. The DLCI is converted and mapped between its origin and its destination, according to a scheme specified by the network service provider, and may change each time it passes through a network switch. A DLCI differs from an address because each data frame only identifies a circuit in each direction, not a source and destination.

Frame Relay networks incorporate error detection, but not error recovery, to permit faster data throughput. Because this scheme does not guarantee delivery of packets, several flow control mechanisms are provided to minimize the possibility of network overflow. (If the network overflows due to buffer pools being full, it may drop packets because of the lack of error recovery.) Three mechanisms are available to control flow:

1. The network service provider specifies the Committed Information Rate (CIR), which is the rate in bytes-per-second that data should be successfully delivered by the network, under normal conditions.

Depending on capacity, the network may be able to accept data "bursts" at faster rates than the CIR. Such information is generally sent with a Discard Eligibility (DE) bit set, indicating that it may be discarded first if the network buffers are filled. The maximum burst rate is specified as a **Be** rate.

2. If the network becomes congested, it may issue a Forward Explicit Congestion Notification (FECN), which tells a receiver to request the sender to temporarily reduce the transmission rate. An FECN is issued as a bit in a header, and consequently does not specify how much to reduce transmission.

3. If the network becomes congested, it may also issue a Backward Explicit Congestion Notification (BECN). This is similar to an FECN, but tells the sender to reduce their rate below the CIR.

In general, high-level error correcting protocols, such as TCP/IP, are most suitable for running over a Frame Relay link, because they can detect and receive any data losses.

Specifying Frame Relay

When you implement Frame Relay service, you will need the following information:

- The **Link Management Interface** (LMI). Each PVC that you implement is assigned an LMI specification. The LMI determines how often PortServer II will poll the network to obtain updates. The network service provider may specify one of several LMIs, including ANSI T1.617 Annex D, ITU (CCITT) Q.933 Annex A, or Cisco/Stratacom Rev. 1.
- The **Line Speed**, which is the maximum bandwidth available from the hardware connection to the network.
- The **Port Speed**, which is the maximum bandwidth that the network service provider will accept from your connection, and is typically the maximum "burst" rate available above the CIR. It is specified by a **BcMax** rate.
- The **Fallback Speed** (**BcMin**), which is the rate used if flow control is effected.

Note: When you order Frame Relay service, remember that port speed is a more limiting factor than CIR. The CIR stated by the service provider is a guaranteed minimum throughput, and can generally be exceeded on a regular basis. It is common to oversubscribe CIRs; that is, the sum of all defined CIRs may exceed that defined port speed. To determine the amount of oversubscription that is possible requires knowledge of your data traffic patterns; guidance on determining this is given later in this chapter.

Designing a network to use with Frame Relay

This section provides some guidelines on upgrading your network to use a Frame Relay service.

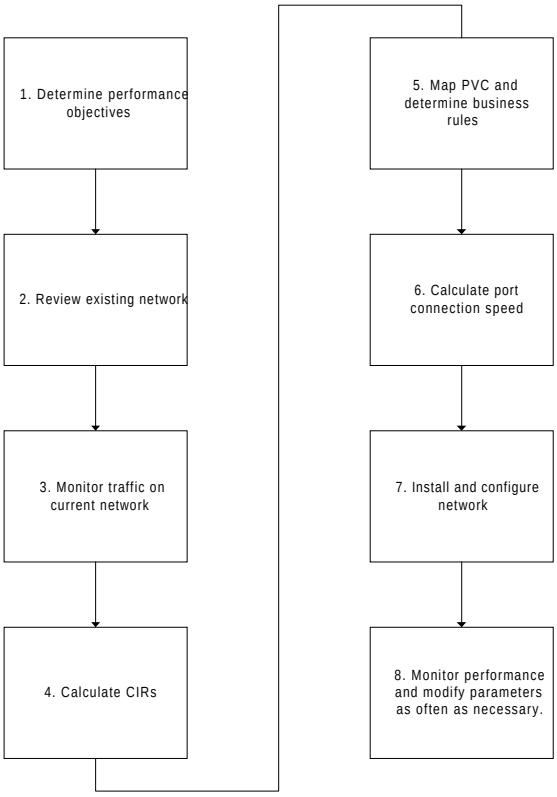

Figure 22 Designing a Frame Relay Network

1. Determine your performance objectives, such as response times, file transfer times, and network availability.
2. Review the configuration of your existing network.
   - Decide which sites or users require access to the Frame Relay service, and which do not.
   - Identify locations for backup servers and files.
3. Monitor the traffic on your network.

- Identify the number of users, the total number of packets transferred, and any variations at different times or on different days.
- Determine the protocols used.
- Note any points of congestion or slow responses experienced by users.

4. Map your PVC (physical circuit) and lay down business rules.

   - Plot the locations of the network nodes and overlay the traffic patterns that you identified in step 3.
   - If the volume of traffic on a particular node is significantly higher in one direction than in another, consider using different connections for incoming and outgoing traffic.
   - Determine rules that will ensure satisfactory performance, for example, the maximum number of users on each 56Kbps channel.

5. Calculate the CIR necessary to support the expected **average** volume of traffic, taking into account the business rules you defined in step 4.

   Note: Do not use the **maximum** volume of traffic to calculate the CIR. Frame Relay's capability of handling bursts of data allows it to accommodate higher values for short periods.

   - You can use the CIR to allocate priority to certain traffic streams multiplexed on the same access channel.

6. Calculate the port connection speed necessary to support the Inbound CIR. However, if the Inbound and Outbound CIR values differ greatly, and the Outbound CIR is greater, adjust the port speed accordingly. (Buffering at the router and the port connection allows Outbound PVCs to support higher values for short periods; buffering is not available to Inbound PVCs).

   - Do not add together the Inbound CIR and the Outbound CIR to calculate the port speed. The inbound and outbound traffic streams do not contend for the same port capacity.
   - The CIR of an individual PVC is limited by the port connection, and you can oversubscribe the port by 100% to 200%. For example, a 256 Kbps port may be able to handle up to eight 64 Kbps CIRs (total 512 Kbps).

7. If possible, install your Frame Relay implementation on a small number of sites and users for initial trials.

8. Monitor performance against the goals you identified in step 2 and the business rules you determined in step 4, and against the values obtained from the previous configuration. Modify PVC, CIR, and port connection speed as often as necessary to obtain optimum performance.

Configuring a Frame Relay port

To configure a port for connection to a Frame Relay service:

Note: Each port can support more than one virtual connection (DLCI).

1. Configure the port for connection to a synchronous CSU/DSU. Refer to your CSU/DSU documentation for the required parameters.

2. Set the Frame Relay parameters for the physical serial port to be used for Frame Relay, entering a command in the format:

   `set framerelay range=<port> enabled=on BECN=<option> lmi=<scheme> lmiR1fc=<on/off> MTU=<number> nT1=<time> nN1=<cycles> nN2=<error_threshold> nN3=<count>`

   - `<port>` is the port or range of ports that you are configuring.
   - `enabled=on` starts Frame Relay running immediately. You can also set this to `off` to disable Frame Relay on this port during this set up procedure.
   - `BECN`. If <option> is `on`, configures the port to run at the fallback speed if the BECN bit is set in a received frame. FECN is ignored, even if you set BECN `on`. If you set `BECN=off`, PortServer II ignores the BECN bit.
   - `lmi=<scheme>` configures the Local Management scheme to use on the link, `Rev1`, `AnnexD`, or `AnnexA`. Alternatively, enter `none` if there is no LMI on this link.
   - If `lmiR1fc=` is set `on` and `lmi=lmiRev1`, the original LMI flow control method is used.
   - `MTU` is the size of the largest frame that can be sent from the port in bits.
   - `nT1` is the time between LMI status requests. The default value is 10 seconds.
   - `nN1` is the number of polling cycles between full status enquiries.
   - `nN2` is the error threshold and `nN3` is the monitored events count. If `nN2` errors occur in `nN3` tries, the link is flagged as "down." If `nN2` status enquiry transactions occur without error, the link is restored to operational state.

Note: The names used for the LMI variables by Annex D differ from the other schemes, as follows:

PortServer II name	Equivalent Annex D name
nN1	N391
nN2	N392
nN3	N393
nT1	T391

The following example resets port 1 to its default values with Frame Relay enabled:

```
set framerelay range=1 frame=on BECN=on lmi=AnnexD MTU=1600
nT1=10 nN1=6 nN2=3 nN3=4 LMr1fc=off
```

Note: For detailed information on the `set framerelay` options, refer to the *Command Reference Guide.*

3. Set the configuration of each virtual connection (DLCI) that is associated with this Frame Relay link, by entering a command in the format:

```
set FrDlci port=<number> dlci=<number> enabled=<on/off>
fallback=<seconds> ProtoEncap=<on/off> BcMax=<bps>
BcMin=<bps> Be=<bps> CIR=<bps>
```

Note: Each DLCI is unique to a specific port and is referred to as a port/DLCI pair number (for example, 1/128). DLCI numbers must be between 16 and 991, and are assigned by your service provider.

- `enabled=on` allows traffic on this connection, off disables the connection.
- `fallback=` sets the period that PortServer II will use the fallback speed on this DLCI after receiving a BECN bit in a packet. BECN must be set to `on` in step 2 for this setting to be effective.
- If `ProtoEncap=on`, PortServer II uses protocol encapsulation (RFC1490).
- `BcMax` sets the guaranteed maximum transfer rate on the link in bps, within the allowable range.
- `BcMin` sets the fallback transfer rate in bps. The default value is half the BcMax setting.
- `Be` sets the permitted burst transfer rate in bps, within the allowable range.
- `CIR` sets the committed burst rate in bps, within the allowable range.

Note: BcMax, BcMin, Be, and CIR must be set to the values required by your service provider and must reflect the capacity of the Frame Relay circuit. If you do not enter values, BCMax defaults to the value of CIR, BcMin to CIR/2, and Be to zero.

For example:

```
set FrDlci port=1 dlci-128 enabled=on fallback=10
ProtoEncap=on BcMax=56000 BcMin=28000 Be=0 CIR=56000
```

Note: For detailed information on the `set FrDlci port` options, refer to the *Command Reference Guide*.

4. Configure a user for each WAN connection, using the `set user` command. See *Chapter 4, Basic Configuration* and *Chapter 8, Configuring WAN Connections* for more information. Ensure that you set the user's protocol to `FrRelay`. The user must also be configured to use IP addresses, by setting `IPAddr` to the remote IP address of the Frame Relay service. The local IP address should be set to 0, allowing the PortServer II's Ethernet address to be used.

5. When configuration is complete, start the Frame Relay link by entering:

   `set user <name> dialout=on`

Chapter 15

Configuring RADIUS

When should I read this Chapter?

Read this Chapter if you want to configure PortServer II to use the RADIUS authentication protocol on its dial-up connections. You must have RADIUS available on your server.

In this Chapter

This Chapter describes how to configure PortServer II to use RADIUS on all its dial-up connections.

The Chapter includes the following topics:

How does RADIUS work?

RADIUS (Remote Authentication Dial In User Service) is a method of remotely maintaining a database of profiles for dial-in users. RADIUS requires two components, an authentication host server and client protocols. Client protocol software is included with PortServer II's software. PortServer II sends authentication requests to the server and acts on its responses. The RADIUS server accepts and processes authentication requests, and informs PortServer II of the results. For example, in a UNIX environment, the RADIUS server authenticates users against a UNIX password file, Network Information Services (NIS), and a separately-maintained RADIUS database.

When a user logs into a PortServer II that is configured to use RADIUS, PortServer II collects login information such as username and password. It then looks in its local database of users for the username; if it finds the username, the user is locally authenticated. If the local authentication fails, PortServer II creates an **Authentication Request** including attributes such as the user's name, the user's password, and the port through which the user dialled in. For protection against eavesdropping, it hides any password present using an encryption algorithm.

PortServer II then submits the Authentication Request to the RADIUS server via the LAN or WAN. The time it waits for a response and the number of retries are configurable at the RADIUS server. If it receives no response, it may route the request to an alternate RADIUS server, depending on how the network is configured.

The RADIUS server validates the Authentication Request, and decrypts the password. It passes validated information to all compatible security systems maintained on the system.

If any validation condition is not met, the RADIUS server returns an **Access Reject** message to PortServer II. This indicates that the user request is invalid and PortServer II denies the user access.

If all validation conditions are met, the RADIUS server returns an **Access Acknowledgment** message. This message may include additional information, such as the protocol to use, or filtering information to restrict the user to specific resources. PortServer II then provides the user with the service indicated by the Access Acknowledgment message.

To ensure that requests are not responded to by unauthorized intruders on the network, the RADIUS server sends an authentication key or password in each transaction, identifying itself to PortServer II.

Configuring RADIUS

To configure PortServer II to obtain user name and password authentication from a RADIUS server for users on all modem ports:

1. Configure a modem for each dial-in connection, as described in *Chapter 9, Configuring Modems*.

2. Enable RADIUS authentication on all dial-up ports by entering a command in the format:

   `set radius run=on primary=<ipaddress> secret=<password>`

   <`ipaddress`> is the IP address of the primary RADIUS server. <`password`> is the authentication key passed between PortServer II and the RADIUS server, and must be the same as the authentication key configured on the RADIUS server. For example:

   `set radius run=on primary=192.83.159.1 secret=x3y67pft973`

   Note: If you have a secondary RADIUS server to use when the primary server is down or cannot be contacted, configure it with a command similar to the following example:

   `set radius secondary=192.83.159.2 secret=nnxzf25308`

3. If any RADIUS authenticated user will be assigned an IP address from a pool, set up the IP Address Pool, as described in *Chapter 4, Basic Configuration*.

4. Configure RADIUS on your server, as described in the next section.

Configuring RADIUS on a Server

To use RADIUS with PortServer II, you must have at least one RADIUS server on your network. You can download the RADIUS server binary code from Digi International's FTP server, if you do not already have it installed.

Note: The Digi International FTP server includes the following RADIUS server binaries: Linux, SCO Unix, Solaris, Unixware SVR4, Windows NT, BSD, and Free BSD.

To download the FTP server binary:

1. Connect to the Digi International FTP server **ftp.dgii.com**.
2. Change directory to **drivers/portserv/radius**.
3. Identify the **/bin** directory containing the binary for your server (for example, **/bin/sco** contains the binary for SCO Unix server) and download the file.
4. Install the binary following the instructions contained in the associated README file.

Chapter 16

Remote Configuration

When should I read this Chapter?

Read this Chapter if you want to configure PortServer II from a remote host. You can update both the PortServer II software and your own configuration file in this way.

In this Chapter

This Chapter describes how to transfer files to and from PortServer II by TFTP, and how to edit the configuration file.

The Chapter includes the following topics:

When should I use Remote Configuration?

You would normally configure PortServer II from a remote host, rather than a terminal connected to one of the serial ports, in two situations:

1. The PortServer II software has been upgraded and you want to use the new version on your PortServer IIs. You can obtain a copy of the new software and install it on your own TFTP server; you can then update each of your PortServer IIs from your own server.

   Note: Do not TFTP boot directly from Digi's FTP server.

2. You want to configure all your PortServer II's from a central location, for example because the PortServer IIs are geographically remote from each other.

   Note: In situation 1, you are changing the functionality provided by Digi International's software. In situation 2, you are changing the configuration required to make PortServer II operate in your network.

Upgrading PortServer II Software

PortServer II's software is stored in Flash ROM that can be upgraded without physically changing the ROM or other hardware. To upgrade the software:

1. Obtain a copy of the new version of software from Digi International's FTP server and place it on your TFTP server. The address of the Digi FTP server is given in *Chapter 18, Digi Support Services*.

2. Configure PortServer II for booting from a boot file, by entering a command in the following format:

   `set config boothost=<hostip> bootfile=<filename> tftp=yes`

   `<hostip>` is the IP address of the server on which the new version is located. `<filename>` is the name of the new version; depending on your host, you may have to enter the full path name.

   

   Caution!

   Always set `tftp=yes` when you first reboot PortServer II with the new version. This allows you to check correct functionality of the new version, while the previous version remains in Flash ROM, and can be used if the image is corrupted during transfer. You can still run PortServer II via TFTP, if the flash ROM gets

corrupted during downloading. Alternatively, set `tftp=smart`, so that if PortServer II cannot find the file, it will not repeat the attempt.

3. Reboot PortServer II by switching power on and off, or by entering `boot action=reset`. When rebooting is complete, PortServer II will operate from the new version of software.

4. Check that PortServer II operates correctly with the new version of software when configured for your application.

5. If PortServer II operates correctly, load the new version of software into Flash ROM, by entering:

   `boot load=<host:filename>`

   `<host:filename>` is the full path and file name of the new version of software on the TFTP server.

   If no errors occur, PortServer II displays the message "*The image now in flash memory appears valid*" on the administrative terminal.

6. Set PortServer II to using the new software in Flash ROM by entering:

   `set config tftpboot=no`

7. If you wish, reboot from Flash ROM by switching power on and off, or by entering `boot action=reset`

8. Repeat steps 2 through 6 for other PortServer IIs on your network.

Editing PortServer II's Configuration from a Remote Host

PortServer II includes a `cpconf` (copy configuration) command that allows you to copy the PortServer configuration file to a specified host for editing. After editing, you can write the modified configuration file back into PortServer II. You can also copy the file into another PortServer II, allowing you to copy similar configurations from one PortServer II to another. The host must be a TFTP server.

Note: The `cpconf` command includes an option (`cpconf term`) that allows you to capture the configuration file to the terminal from which you issued the command. You can then change the configuration file with a text editor before reloading it by sending the file to PortServer II's command line. Because writing the entire configuration file back into PortServer II's Flash ROM is a relatively slow process, we recommend that you use the single parameter `set` commands (`set user`, `set wan`, `set link` etc.) described in previous chapters to make configuration changes from a local terminal. If you use a local terminal to edit the configuration file, ensure the flow control is set correctly between PortServer II and the terminal before you begin.

If you have several PortServer IIs to configure with similar configurations, you may want to maintain a master configuration file on the host and create copy configuration files that you can download into each PortServer II.

Similarly, if you have many users with similar profiles, you may want to consider "cutting and pasting" the relevant lines from the master configuration file to the copy files. You can then edit the copy before downloading it into PortServer II.

Copying a PortServer II configuration file to a host

To copy a configuration file to the host:

1. Create a file with appropriate write permissions on the host by entering a command similar to the following example at a UNIX server:

   `touch /tftpboot/psconfig; chmod 666 /tftpboot/psconfig`

   `psconfig` is the file name for the configuration file.

   Note: It may not be necessary to create a file with write permissions on certain servers; check your host documentation for further information.

2. Copy the configuration file to the host by entering the following command at PortServer II's command line:

   cpconf tohost <host> <filename>

   <host> may be the name or IP address of the host. <filename> is the name of the file created in step 1 (psconfig in the example); you may or may not have to enter the full path name, depending on your TFTP server.

Editing the configuration file

When the configuration file is installed on the host, open it with any text editor and review the content. Each line of the file contains a set statement in the same format as described in previous chapters, for example:

```
set line range=1-4 baud=2400 parity=e csize=7 stopb=1
set config ip=192.83.159.1 domain=digii.com
set port termtype=wy60 range=2-16
set line baud=38400 range=2-16
set flow ixon=on range=2-16
set flow ixoff=off range=1-16
```

The number of lines in the file and their content depends on the previous configuration of the PortServer II. You can edit the file, according to the following guidelines:

- If you add a line to the file, ensure it is in the same format as other lines for the same set command.
- Only delete a line from the file if you are sure the parameters it specifies are no longer required.
- Only change parameters located between an "=" sign and the next space.
- Never delete a required parameter from a line. If you are unsure whether a parameter is required or optional, refer to the *Command Reference Guide*.
- Begin comment lines with the "#" character. All such lines are ignored by the PortServer II command line interpreter.

Important: If you edit a configuration file incorrectly and download the incorrect file, PortServer II may not function correctly.

Restoring a PortServer II configuration file from a host

When you have completed editing of the configuration file:

1. Save it to the same location, overwriting the file you uploaded from PortServer II.
2. Copy the configuration file back to the PortServer II by entering the following command at PortServer II's command line:

   `cpconf fromhost <host> <filename>`

   `<host>` may be the name or IP address of the host. `<filename>` is the name of the edited file (`psconfig` in the previous example); you may have to enter the full path name.

   Note: It will take longer to update the configuration file than to download it, because it takes longer to write to Flash ROM than to read from it.

3. If you want to keep the configuration file on the host, remove write permissions to prevent unauthorized users from changing the file. For example, on a UNIX system, enter:

   `UNIX: chmod 444 savefile`

TFTP Error Messages on PortServer II

If a TFTP error occurs while booting, the host returns an error message, which PortServer II displays on the alphanumeric display on the front panel. This message will be in the format **F***n*, where **F***n* can be one of the following:

F0	Error not defined, see error message (if any)
F1	File not found
F2	Access Violation
F3	Disk Full or allocation exceeded
F4	Illegal TFTP operation
F5	Unknown transfer ID
F6	File already exists
F7	No such user

If a TFTP error occurs:

1. Check that the remote host system supports TFTP.
2. Repeat the TFTP command.
3. If the problem reoccurs:
    a. Press the arrow keys (< and >) on PortServer II's front panel to reboot from internal Flash ROM
    b. Repeat the configuration command `set host` to ensure that the host's IP address, and the path and file name of the configuration file are correct. The syntax for `set host` is given in the *Command Reference Guide*.

If the problem continues, you may have a system or network incompatibility that prevents the use of TFTP.

Chapter 17

Troubleshooting

When should I read this Chapter?

Read this Chapter if you have just installed PortServer II and want to check it is functioning correctly, or if you are having difficulty with a previously installed PortServer II.

In This Chapter

This Chapter contains information on basic diagnostic and fault tracing features.

The Chapter includes the following topics:

Power On Self Test

The PortServer II Power On Self Test (POST) sequence is initiated after PortServer II is switched on, reset, or an internal watchdog timer expires. It comprises a series of basic tests that ensure the main components (such as the microprocessor, memory, and network interface controller) are functioning properly.

One of the ten LED indicators on the front panel lights on successful completion of each of the ten tests. If an error occurs, the corresponding LED is not lit. All ten LEDs then light for one second on completion of the POST sequence. The following table lists the tests performed during the POST, and the LED that corresponds to each test.

POST Test		**LED**
0	CPU test	TD
1	Watchdog test	RD
2	ROM checksum, flash ROM test	RTS
3	RAM test x1000 to top of 64K memory	CTS
4	Timer and real-time clock test	DSR
5	Ethernet and Ethernet address test	DCD
6	On-Board UARTs test	DTR
7	External UARTs test	RI
8	Off when warm-booted	OFC
9	Bad code in Flash ROM	IFC

Interpreting the alphanumeric display

The front panel alphanumeric display shows codes that assist troubleshooting:

- **b** indicates that PortServer II is booting from a remote bootp server.
- **F** indicates that PortServer II is booting from a remote TFTP server.
- **CL** indicates PortServer II is clearing previous configuration information.

Note: You can clear stored configuration information by holding down the < and > pushbuttons while switching on power, and holding them down until **CL** is displayed. (See *Resetting PortServer II to Factory Defaults* later in this Chapter).

- Error codes **E0** through **E9** indicate a failure of one of the POST checks. The number corresponds with the number of the test listed in the table above. For example, **E4** indicates a real-time clock error.
- **PU** indicates the LED display is showing Processor Utilization. If all ten LEDs are lit, PortServer II's CPU is 100% utilized. Under normal operation between one and five LEDs should be lit, depending on the traffic.
- **AC** indicates PortServer II has passed the tests and is now in normal operating mode.

User Diagnostics

You can run diagnostics either from a terminal (or PC with terminal emulation software) connected to Port 1 or from PortServer II's front panel. Diagnostics can be selected any time before the POST test described above finish. To select diagnostics:

- **From the terminal:** Enter either "V" or "v" at a terminal connected to Port 1.
- **From the front panel:** Press either of the "arrow" pushbutton (< or >).

Important: When you run diagnostics from a terminal or PC that is connected to a serial port, the DTR and RTS lines on the serial port will be low (inactive). Ensure that your terminal or PC will send the "v" character when these lines are low. When in doubt, or you encounter problems, use a 3-wire connection to the terminal or PC.

Terminal diagnostics

1. If you have not already done so, set the terminal connected to Port 1 to 9600 baud, 8 data bits, 1 stop bit, and no parity before you begin diagnostics.

2. Press V to enter diagnostic mode and check that the start-up screen shown below appears. The screen displays test information, and waits for user input.

```
Digi International Inc. PortServer II
Ethernet address....12:34:56:78:9A:BC
 ROM revision: 91-398 Rev D
 ROM startup (cold boot)
Instruction cache size: 0x00001000
    Data cache size: 0x00000800
       RAM size: 0x00200000
CPU test............passed
Watchdog test.......passed
ROM checksum test...passed
Flash RAM test......passed
Simple
RAM test............0xA00010000xA01F0010 passed
Complex
RAM test............0xA00010000xA0002000 passed
Timer test..........passed
RT clock+RAM test...passed
Ethernet internal...passed
Ethernet external...passed
Test EBI 0 UART08..15 devices passed
EBI 1...............none
EBI 2...............none
```

```
EBI 3..............none
Press "?" for diagnostics menu
or carriage return to continue booting
```

3. When you press ?, the diagnostics menu appears, as shown below. This lists the eight tests that you can select. These tests perform diagnostic checks on hardware within PortServer II.

```
DIAGNOSTICS MENU---

A = All tests (except 9)
1 = Front panel light test
2 = RAM test
3 = Timer and real-time clock test
4 = EBI internal loopback test
5 = EBI external loopback test
6 = Ethernet internal loopback test
7 = Ethernet external loopback test
8 = Flash RAM test
9 = Watchdog test

C = Configure boot
T = Set date & time
Ethernet address: 12:34:56:78:9A:BC
B = Reboot
```

4. To start any test, press the appropriate test number. See *Basic Test Descriptions* later in this chapter for information on the individual tests.

Front panel display diagnostics

1. Press either the left or right pushbutton on the front panel during the POST tests to enter diagnostic mode. Check that all of the LEDs go OFF, the left alphanumeric display is blank, and the right alphanumeric display shows **1**. The right decimal point blinks at one second intervals. At this point, **Test 1 - Panel Light Test** is selected.

2. To select a test, press the right pushbutton (>). The right alphanumeric display will cycle through the possible test numbers. Once you have selected the desired test, press the left pushbutton (<) to start the test. The left decimal point lights to indicate the test has started and the appropriate test results are displayed on the alphanumeric display.

   The selected test will continue to cycle until an error is detected or until you stop the test by pressing the (<) pushbutton. Failures are detected and the cumulative total is shown on the alphanumeric display.

3. To stop the test, press and hold the < button, and wait for the left decimal point to go OFF. Then release the (<) pushbutton and the test will stop.

The left alphanumeric display will display either a **P** or an **F** for pass or fail respectively. The right display will show the test number that was just executed. For example, **P3** indicates that test 3 has passed, while **F3** indicates that the same test failed.

Basic Test Descriptions

The following descriptions explains the operation of each diagnostic test. Each test is the same whether it is run from the terminal or the front panel.

Test 1 - Panel Light Test

This test cycles the ten LEDs on and off, and in an alternating on/off pattern. It also cycles both alphanumeric displays in one second intervals. You must make a visual check of the LEDs and displays to ensure that they all cycle correctly.

Test 2 - Memory Test

The memory test takes about two minutes for each pass.

Test 3 - Timer Test

This tests the timer to ensure that it counts and interrupts. It tests the real-time clock, checking the set time for legal values. It also tests the battery-backed RAM in real-time clock.

Test 4 - Built-in UART and External EBI Internal Loopback Test

This tests the UARTS within PortServer II, and also the external EBI expansion modules with an internal loop-back test. It tests that the UARTs interrupt and that data loops back.

Test 5 - Built-In UART and External EBI External Loopback Test

Same as Test 4, but uses an external loopback. From the front panel, press the right pushbutton (>) to select the port to test. The display will cycle from **01** to **16** (or higher, if you have expansion modules connected) plus **All**. With the number of the port to test displayed, press the left pushbutton (<) to start the test.

Test 6 - Test Ethernet Internal Loopback

This tests the 10BaseT and 10Base2 circuits within PortServer II.

Test 7 - Test Ethernet External Loopback

This verifies correct operation of the Ethernet interfaces to the external network. You must have a dummy network (for example, a “T” connector with termination resistors) attached to each Ethernet connector for this test to pass.

Test 8 - Test Flash ROM

(Read Only) This verifies that the Flash ROM can be read.

Test 9 - Watchdog Timer Test

This test checks out the watchdog timer that is used to check system reliability.

Resetting PortServer II to Factory Defaults

You can reset PortServer II to its factory default settings under a number of situations, for example:

- You want to use a previously configured PortServer II in a different installation that will require a new configuration.
- You are experiencing unpredictable behavior or PortServer II is “locking up.”

Caution!

Resetting PortServer II to the factory defaults will cause you to lose *all* your configuration settings. Use the `cpconf` command if you want to save your configuration. The syntax for `cpconf` is given in the *Command Reference Guide*.

To reset PortServer II to the factory defaults:

1. Turn off the PortServer II’s power switch.
2. Press *and hold* both “arrow” pushbuttons (< and >) at the same time, and turn the power switch back on.
3. Hold the pushbuttons until the display shows **CL** (CLear program).

This resets PortServer II to boot from internal flash ROM. The default Root password reverts to `dbps`.

Statistics

Viewing statistics

You can use the `info` command to view PortServer II's network statistics tables or to clear their contents (see the *Command Reference Guide* for more details). The information in the tables includes the statistics gathered since the tables were last cleared.

Note: All users can view statistics. You require root privileges to clear tables.

You can view any of the following tables by entering `info` *`tablename`*, where *`tablename`* may be any of the following

- framerelay
- ip
- icmp
- tcp
- udp

Clearing statistics

You can enter the following command to clear all of the tables:

```
info clear
```

Interpretation of statistics

The following screen is an example of a statistics screen. These are IP statistics obtained from a root level session:

```
#> info ip

Network statistics collected over last 34 minute(s).

ipInHdrErrors :     0      ipInAddrErrors  :     0
ipOutNoRoutes :     0      ipForwDatagram  :     0
ipOutRequests :   298      ipInReceives    :   657
ipFragCreates :     0      ipFragOKs       :     0
ipReasmFails  :     0      ipReasmOKs      :     0
ipInDiscards  :     0      ipInUnknownProtos :   0
```

Details of how to interpret the information displayed in response to the `info` command is given below.

IP Statistics

The following statistics are displayed if you enter `info ip`.

ipInHdrErrors

The number of incoming datagrams that were discarded due to errors in their IP headers. Errors that cause a datagram to be discarded include bad checksums, version number mismatch, other format errors, time-to-live exceeded, and errors discovered in processing the IP options. In a correctly configured network, this number should be zero or near to zero.

ipInAddrErrors

The number of incoming datagrams that were discarded because the IP address in the IP header's destination field was not a valid address to be received at PortServer II's network. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). In a correctly configured network, this number should be zero or near to zero.

ipOutNoRoutes

The number of outgoing datagrams that were discarded because no route could be found to transmit them to their destination. The count includes:

- Any packets counted in **ipForwDatagrams** (see below) which also meet this "no-route" criterion.
- Any datagrams that a host cannot route because all of its default gateways are down.

In a correctly configured network, this number should be zero or near to zero.

ipForwDatagram

The number of incoming datagrams for which PortServer II's network was not the final IP destination. PortServer II's attempts to find a route to forward these datagrams to their final destination(s) were not successful.

ipOutRequests

The total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission.

Note: This counter does not include any datagrams counted in **ipForwDatagrams**.

ipInReceives

The total number of incoming datagrams received by IP, including any received in error.

ipFragCreates

The number of IP datagram fragments that have been generated as a result of fragmentation at PortServer II.

ipFragOKs

The number of IP datagrams that have been successfully fragmented at PortServer II.

ipReasmFails

The number of failures detected by the IP re-assembly algorithm. These failures may be for reasons such as time-outs and errors.

Note: This count is not necessarily a count of all discarded IP fragments because some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

ipReasmOKs

The number of IP datagrams that have been successfully re-assembled.

ipInDiscards

The number of incoming IP datagrams for which no problems were encountered to prevent processing, but which were discarded (e.g., for lack of buffer space).

Note: This counter does not include any datagrams that were discarded while awaiting re-assembly.

ipOutDiscards

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space).

Note: This counter include datagrams counted in **ipForwDatagrams** if any such packets met this (discretionary) discard criterion.

ipInUnknownProtos

The number of datagrams that were addressed to PortServer II's network and received successfully but were discarded because of an unknown or unsupported protocol.

ICPM Statistics

The following statistics are displayed if you enter `info icpm`.

icmpInMsgs

The total number of ICMP messages received by PortServer II. This counter includes all messages counted by icmpInErrors.

icmpInErrors

The number of ICMP messages received by PortServer II, but determined as having ICMP-specific errors (for example, bad ICMP checksums or bad length).

icmpInEchos

The number of ICMP Echo (request) messages received.

icmpInAdrMsk

The number of ICMP Address Mask Request messages received.

icmpInAdrMskRp

The number of ICMP Address Mask Reply messages received.

icmpInEchoRp

The number of ICMP Echo Reply messages received.

icmpInRedirect

The number of ICMP Redirect messages received.

icmpInDstUnrec

The number of ICMP Destination Unreachable messages received.

icmpInSrcQuenc

The number of ICMP Source Quench messages received.

icmpInTimeExcd

The number of ICMP Time Exceeded messages received.

icmpInParmProb

The number of ICMP Parameter Problem messages received.

icmpInTimest

The number of ICMP Timestamp (request) messages received.

icmpInTimestRp

The number of ICMP Timestamp Reply messages received.

icmpOutMsgs

The total number of ICMP messages which PortServer II attempted to send.

Note: This counter includes all those messages counted by **icmpOutErrors**.

icmpOutEchos

The number of ICMP Echo (request) messages sent.

icmpOutAdrMsk

The number of ICMP Address Mask Request messages sent.

icmpOutAdrMskR

The number of ICMP Address Mask Reply messages sent.

icmpOutEchoRp

The number of ICMP Echo Reply messages sent.

icmpOutRedirec

The number of ICMP Redirect messages sent.

icmpOutDstUnre

The number of ICMP Destination Unreachable messages sent.

icmpOutSrcQuen

The number of ICMP Source Quench messages sent.

icmpOutTimeExc

The number of ICMP Time Exceeded messages sent.

icmpOutParmPro

The number of ICMP Parameter Problem messages sent.

icmpOutTimest

The number of ICMP Timestamp (request) messages sent.

icmpOutTimestR

The number of ICMP Timestamp Reply messages sent.

TCP Statistics

The following statistics are displayed if you enter `info tcp`.

tcpOutSegs

The total number of segments sent. This includes those on current connections, but excludes those containing only retransmitted octets.

tcpOutRsts

The number of TCP segments sent containing the RST flag.

tcpPassiveOpen

The number of times that TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

tcpAttemptFail

The number of times that TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

tcpInSegs

The total number of segments received, including those received in error. This count includes only segments received on currently established connections.

tcpInErrs

The total number of segments received in error (for example, bad TCP checksums).
tcbEstabResets

The number of times that TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

tcpRetransSegs

The total number of segments retransmitted - that is, the number of TCP segments transmitted containing one or more previously transmitted octets.

tcpActiveOpens

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

UDP Statistics

The following statistics are displayed if you enter `info udp`.

udpInDatagrams

The total number of UDP datagrams that have been delivered to UDP users.

udpInErrors

The number of received UDP datagrams that could not be delivered for any reason other than the lack of an application at the destination port.

udpOutDatagrams

The total number of UDP datagrams that have been sent from PortServer II.

udpNoPorts

The total number of received UDP datagrams for which there was no application at the destination port.

Interface Statistics

The following statistics are displayed if you enter `info if`.

ifInOctets

The total number of octets received on PortServer II's interface, including framing characters.

ifInUcastPkts

The number of subnetwork unicast packets that have been delivered to a higher layer protocol.

ifInNUcastPkts

The number of non-unicast (i.e., subnetwork-broadcast or subnetwork multicast) packets delivered to a higher layer protocol.

ifInDiscards

The number of inbound packets that were discarded, even though no error was detected that would prevent delivery to a higher layer protocol. One possible reason for discarding such packet is to free buffer space.

ifInErrors

The number of inbound packets that contained errors preventing them from being delivered to a higher layer protocol.

ifUnknownProtos

The number of packets received at PortServer II's interface that were discarded because of an unknown or unsupported protocol.

ifOutOctets

The total number of octets transmitted from PortServer II's interface, including framing characters.

ifOutUcastPkts

The total number of packets that higher level protocols requested be transmitted to a subnetwork unicast address, including those packets that were discarded or not sent.

ifOutNUcastPkts

The total number of packets that higher level protocols requested be transmitted to a non-unicast (that is, a subnetwork broadcast or subnetwork multicast) address, including those packets that were discarded or not sent.

ifOutDiscards

The number of outbound packets that were discarded even though no error was detected that would prevent transmission. One possible reason for discarding such a packet is to free buffer space.

ifOutErrors

The number of outbound packets that could not be transmitted because of errors.

Frame Statistics

The following statistics are displayed if you enter `info frame`.

In Total

Total number of frames received.

In IP

Number of IP protocol frames received.

In ARP

Number of ARP frames received.

Out Total

Total number of frames sent by PortServer II's interface.

Out IP

Number of IP frames sent.

Out ARP

Number of arp frames sent.

Frame Relay Statistics

The following statistics are displayed if you enter `info framerelay`.

frCircuitReceivedFrames

The number of frames received over this virtual circuit.

frCircuitReceivedOctets

The number of octets received over this virtual circuit.

Received Fragments

The number of fragments received over this virtual circuit.

Reassembled Frames

The number of frames successfully re-assembled from fragments.

Reassemble Failures

The number of failures to re-assemble fragments into a complete frame.

frCircuitReceivedBECNs

The number of frames containing BECN (backward congestion notification) that were received from the network.

frCircuitSentFrames

The number of frames sent from this virtual circuit.

frCircuitSentOctets

The number of octets sent from this virtual circuit.

Sent Fragments

The number of fragments sent from this virtual circuit.

Sent Fragmented

The number of frames that were fragmented before sending.

frCircuitReceivedFECNs

The number of frames received from the network indicating forward congestion.

frErrType

The type of error that was last seen on this interface.

too long

The number of frames received that exceeded the maximum frame size on this circuit.

bad DLCI

The number of frames that were received with an invalid DLCI.

undefined LMI error

The number of received LMI packets that did not match the LMI specifications.

LMI bad sequence

The number of LMI packets received with a bad sequence number.

frErrData

The portion of the frame that caused the error.

frErrFaults

The number of times that traffic was stopped on this circuit due to LMI errors.

frErrTime

The value of sysUpTime at which the error was detected.

undefined error

The number of detected errors that are not defined in the Frame Relay MIB (RFC 1315).

too short

The number of received packets that are smaller than the minimum frame relay packet size.

unknown DLCI

The number of received frames with a DLCI indicating a PVC that is not configured.

LMI unknown IE

The number of Information Elements in received LMI packets that had an unrecognized type code.

LMI unknown report

The number of LMI reports received that had an unrecognized type code.

status requests

The number of LMI status requests sent.

status responses

The number of LMI status responses sent.

rcvd sequence number

The last sequence number received.

updates or async status

The number of updates or async status reports received. The value shown depends on the type of LMI used.

full status requests

The number of full status requests sent.

full status responses

The number of full status requests received.

sent sequence number

The last sequence number sent.

frames/octets sent within CIR

The number of frames/octets sent within the Committed Information Rate.

frames/octets sent beyond CIR

The number of frames/octets sent beyond the Committed Information Rate.

frames/octets buffered

The number of frames/octets held for later transmission because sending them would have caused the Committed Information Rate to be exceeded.

frames/octets dropped

The number of frames/octets discarded because sending them would have caused the Committed Information Rate to be exceeded.

An example of a Frame Relay statistics display is shown below:

```
#> info frame:1-3,5,8,9-10
Link Index 8
frCircuitReceivedFrames:19987 frCircuitSentFrames : 19813
frCircuitReceivedOctets:569596 frCircuitSentOctets : 523582
Received Fragments:          0 Sent Fragments :          0
Reassembled Frames :          0Sent Fragmented :          0
Reassemble Failures :          0

frCircuitReceivedBECNs: 0 frCircuitReceivedFECNs: 0

frErrType : 3         frErrTime :     921300
receive errors : 83 undefined error : 0
too long : 83         too short : 0
bad DLCI : 0         unknown DLCI : 0
undefined LMI error : 0 LMI unknown IE : 0
LMI bad sequence : 0 LMI unknown report : 0
frErrData : 4 1 3 cc 45 0 5 dc 5b 4f 20 0 ff 1 52 21

frErrFaults : 0


LMI statistics
status requests : 19726 full status requests : 3288
status responses : 19726 full status responses : 3288
rcvd sequence number : 177 sent sequence number : 177
updates or async status: 8
```

Note: The info `framerelay` option displays counters and other information about the frame relay circuits (links) or the PVCs (DLCIs) that are bound to any particular circuit.

The first `range` value in the example above specifies the circuits. A single number, a range of numbers, and an "*" for all links may be entered. Even though a mixed range was requested, only Circuit 8 is up in this PortServer II

(the other links requested are not set up for Frame Relay operation).

By adding another index after the link index, it is possible to display the statistics for one, some, or all **active** PVCs for one, some, or all links. For example:

```
#> info fra:*:*
Link Index 8, DLCI 16
frCircuitReceivedFrames: 253 frCircuitSentFrames : 87
frCircuitReceivedOctets: 260432 frCircuitSentOctets : 89610
Received Fragments : 0 Sent Fragments : 0
Reassembled Frames : 0 Sent Fragmented : 0
Reassemble Failures : 0

frCircuitReceivedBECNs : 0 frCircuitReceivedFECNs: 0

frErrType : 3         frErrTime : 921300
receive errors : 83 undefined error : 0
too long : 83         too short : 0
bad DLCI : 0         unknown DLCI : 0
undefined LMI error : 0         LMI unknown IE : 0
LMI bad sequence : 0         LMI unknown report : 0
frErrData : 4 1 3 cc 45 0 5 dc 5b 4f 20 0 ff 1 52 21

frErrFaults : 0


Congestion statistics
frames sent within CIR :       5780
octets                 :    5940896
frames sent beyond CIR :          0
octets                 :          0
frames buffered        :          0
octets                 :          0
frames dropped         :          0
octets                 :          0
```

Note: Only DLCI 16 on Link 8 is active. The command

```
#> info frame:1-3,5,8,9-10:16
```

will produce the same output while the command

```
#> info frame:1-3,5,8,9-10:18
```

will produce no output.

Hardware Error Counts

In Overruns

The number of times a data overrun occurred because PortServer II's Ethernet controller was unable to place a received frame in memory.

In Unaligned

The number of times that misaligned frames arrived.

In No Resource

The number of times that an incoming frame could not be processed due to lack of available buffers.

In Collision

The number of Ethernet collisions that were detected after a destination address was received.

In Short Frame

The number of frames received that were too short.

In Bad CRC

The number of frames received with bad CRC.

Out No Carrier

The number of frames lost when lack of carrier was detected.

Out Lost CTS

The number of frames lost when ClearToSend was reset.

Out DMA Underrun

The number of frames that were lost because transmit buffers were not available.

Out Deferred

The number of times that a transmission was deferred.

Out Collisions

The number of times that an Ethernet collision was detected after starting a transmission.

Trace Messages

When PortServer II encounters an error or unexpected situation, it may print trace messages to help the network administrator or other user identify the problem. You can select trace messages in a specific group if you know the source of the problem, or you can print all trace messages to help isolate a problem that you do not know the cause of.

Trace messages are stored in a circular buffer, which is filled with messages of one of two types, historical or concurrent. In historical mode, the buffer contains all event messages from the group(s) specified by the user. In concurrent mode, the buffer is emptied frequently to make room for new messages. Consequently, the buffer occupies less memory when in concurrent mode.

Depending on the mode selected, the events being traced, and the amount of PortServer II activity, some messages may be dropped. If messages are dropped, the number of dropped messages is noted in a displayed message.

Under normal operation, you should disable trace messages to reduce processor overhead which would otherwise slow PortServer II operation.

Enabling Trace Messages

To enable trace messages, enter a command with the following syntax:

```
set trace [loghost=ipaddress] mode=historical|concurrent
state=on|off|dump syslog=on|off mask=type:severity
```

`loghost` sends trace messages to the host specified by `ipaddress`, which must support the syslog daemon. `syslog=on` enables this selection.

`mode` sets the operation of the trace buffer, either `historical` or `concurrent`.

If `state=on`, all messages in the trace buffer are displayed; when all messages have been displayed, the trace feature remains active to display new messages entering the buffer. If `state=off`, no trace messages are displayed. If `state=dump`, all messages in the buffer are displayed, then the state returns to off.

`mask` specifies the group(s) and severity of message to trace. When troubleshooting, you will normally only be concerned with the CRITICAL and WARNING severity messages.

For example, if you want to display historical trace messages for all groups and all severities, enter:

```
set trace mode=historical trace=on mask=*.*
```

To display all trace messages in the TCP and WAN groups, enter:

```
set trace mask=tcp:* wan:*
```

Similarly, you may want to display all WARNING messages. To do this, enter:

```
set trace mask=*:warning
```

For detailed information on set trace, refer to the *Command Reference Guide*.

Explanation of Trace messages

Trace messages that indicate CRITICAL and WARNING conditions are listed below, grouped into groups by function. Where appropriate, guidance is given on resolving the problem. Remember that you must have Trace messages of the particular type enabled (or **all** Trace messages enabled) to see the messages listed below. The option you must include in the **set trace** message is given for each group.

Many of the messages given below indicate internal errors that should not be encountered under normal circumstances. These message types are marked ***internal error*** and are due to problems that the user probably cannot correct. If one of these messages appears, contact Digi Technical Support for assistance.

Other errors may permit actions to be taken by the user to correct the problem encountered. Digi Technical Support will also be happy to assist in resolving these problems.

Certain messages caused by obscure errors are not listed below. If you see a message that is not listed below, you should contact Digi Technical Support for advice.

Critical Trace Messages

Non-specific trace message:

1) memory allocation failure, pc = <program counter>

A memory allocation was attempted at the address specified by **<program counter>**. The memory allocation failed.

ARP messages:

(Option=ARP)

1) RARP response conflict from <IP address1>: answered <IP address2>

An unexpected or late RARP response was received from **<IP address1>** which conflicted with the IP address of PortServer II.

2) received request with invalid address <IP address>

An ARP request was received with an invalid sender IP address.

3) <IP address> entry conflict

An ARP packet caused an conflict in the ARP table. The new **<IP address>** entry has replaced the old IP address.

4) another unit using our address <IP address>

An ARP packet arrived from a unit using the IP address of the PortServer II.

5) gratuitous ARP found IP address conflict <IP address>

Gratuitous ARP detected another box using the IP address of the PortServer II.

6) closed

Internal error

Serial messages

(Option=Serial)

1) Alloc error : Frames free list empty

Internal error

2) allocb failure, invoking flow control for port <port number>

Allocation of memory for port buffers failed.

3) recovered from allocb failure

Allocation of memory for port buffers had failed, but subsequently recovered.

4) ran out of transmit buffers

Internal error

DNS messages

`(Option=DNS)`

1) could not unlink host <name> by name

Internal error

2) could not unlink alias <name1> for host <name2>

Internal error

3) could not unlink host <name> by address

Internal error

4) could not unlink secondary <IP address> host <name> by address

Internal error

Ethernet messages

`(Option=ether)`

1) config prot error

Internal error

2) bind prot error

Internal error

3) closing

Internal error

Frame Relay messages

`(Option=frame)`

1) LMI in error: data traffic stopped

The Frame Relay LMI code has detected at least N392 errors in the last N393 events and has stopped all data traffic until the link recovers. The errors are:

- The switch (DCE) has not responded to status enquiries, or
- The expected sequence number was not received in the response.

2) LMI status change: deleting PVC <DLCI>

The DCE has informed PortServer II that the **<DLCI>** is no longer valid.

3) IE processing error: invalid buffering

Internal error

4) IE processing error: length mismatch

Internal error

5) PCB <dlci> is not up

While attempting to send data to this **<DLCI>**, PortServer II has detected that it is not in service.

6) LCB is not up

While attempting to send data on this LMI, PortServer II has detected that it is not in service.

IP messages

(`Option=ip`)

1) packet discarded from interface <name>: invalid buffering

Internal error

2) packet discarded from interface <name>: couldn't remove padding

Internal error

3) packet discarded from interface <name>: invalid source address

An IP packet with invalid source address was received.

4) packet discarded from interface <name>: unable to forward to UDP user <port>

UDP user **<port>** was congested and PortServer II dropped a packet.

5) couldn't add to route table (out of memory)

An attempt to add a route to the Route Table failed due to a memory allocation failure.

6) couldn't add to route table (invalid gateway)

An attempt to add a route to the Route Table failed because the gateway is unknown

or invalid.

7) couldn't add to route table (invalid metric)

An attempt to add a route to the Route Table failed because metric was not 1 for a direct host address, greater than 3 for a direct net address, or 0 for an indirect address.

8) couldn't add to route table (invalid address)

An attempt to add a route to the Route Table failed because the address is the same as the Ethernet's IP address.

9) couldn't add to route table (invalid mask)

An attempt to add a route to the Route Table failed because the mask equals 0.

10) couldn't add to route table (invalid interface name)

An attempt to add a route to the Route table failed because the interface (WAN) name is unknown or invalid.

NetCX messages

(`Option=realp`)

1) reset server <IP address> : <cause>

An error of this type is caused by a RealPort Driver violating the RealPort Protocol.

If <cause> is:	The error is caused by:
Port is out of range	A RealPort Driver trying to access a port that does not exist
Access to unopened port	A RealPort Driver attempting to cause an action on a port that it had not opened.
Unimplemented open type	A RealPort Driver sending an unrecognized open request type.
Received Illegal Packet type	A RealPort Driver sending an unrecognized packet type.

2) Server setting socket failed

Internal error

3) tcp_netcx: send EOF error: <error number>

Internal error

NETD messages

(`Option=netd`)

1) Could not set default router: max number of static routes already defined

The default router could not be set.

2) could not open enet

Internal error

3) could not configure enet for IP

Internal error

4) could not configure enet for ARP

Internal error

5) could not configure enet for RARP

Internal error

6) could not push ARP

Internal error

7) could not open IP

Internal error

8) could not open IP for WAN use

Internal error

9) could not link ARP to IP

Internal error

10) could not find ethernet

Internal error

11) could not configure IP

Internal error

12) could not configure altip

Internal error

INETD messages

(`Option=inetd`)

1) could not create direct socket

Internal error

2) could not bind direct socket

Internal error

3) could not set listen on direct socket

Internal error

4) could not set accept on port

Internal error

5) error received on entry %d\n",index

Internal error

6) function error on entry %d\n",index

Internal error

7) could not create proxy socket, entry %d

Internal error

8) could not send accept, entry %d\n", index

Internal error

9) couldn't create entry %-d\n", index

Internal error

10) couldn't bind entry %-d\n", index

Internal error

11) couldn't set listen on entry %-d\n", index

Internal error

12) escaped!!

Internal error

PPP messages

(`Option=ppp`)

1) open(<port>): memory allocation failed.

A memory allocation failed.

2) cid(<port>) Peer Rejected Protocol: <protocol number>

The remote server rejected a protocol necessary for PPP operation.

2) (<port>): memory allocation failed\n

A memory allocation failed.

3) (%d): Link Control protocol rejected by peer

The remote server rejected a protocol necessary for PPP operation.

4) IP Control protocol rejected by peer

The remote server rejected a protocol necessary for PPP operation.

5) Link quality protocol rejected by peer

The remote server rejected a protocol necessary for PPP operation.

6) pap rejected by peer

The remote server rejected a protocol necessary for PPP operation.

7) chap rejected by peer

The remote server rejected a protocol necessary for PPP operation.

Routed Messages

(Option=routed)

1) routed socket unopened (%d)\n", errno

Internal error

2) bind failed on routed socket (%d)\n", errno

Internal error

3) bind failed to set options\n", errno

Internal error

4) recvfrom error (%d)\n", errno

Internal error

5) invalid command <command received>

The command received in the RIP packet was invalid.

6) unable to create route entry

A route defined in a RIP packet could not be created due to a memory allocation failure.

TCP Messages

(Option=tcp)

1) transmit retries exhausted

The retry limit for unacknowledged TCP data has been reached; the TCP session will be aborted.

WAN messages

(Option=wan)

1) not enough memory to start dialing process

PortServer II does not have sufficient memory available to dial out to create a connection.

2) not enough memory to start user process

PortServer II does not have sufficient memory available to create a connection.

3) malloc error dialing on port <port>

A memory allocation failed.

4) link task received M_ERROR

An error occurred in a lower layer (possibly the serial driver) and the link will be aborted.

5) Could not open frame relay driver

Internal error

6) <username>:ioctl failed

Internal error

7) failed to push PPP module

Internal error

8) failed to push VJC module

Internal error

9) couldn't open wan driver

Internal error

10) unable to get minor number

Internal error

11) could not link to wan driver

Internal error

12) Wan Interface table exhausted.

Internal error

13) No memory for WAN Interface.

PortServer II does not have sufficient memory available for the WAN Interface.

14) Could not open wan driver.

Internal error

15) unable to get minor number

Internal error

16) could not link ip to network

Internal error

17) could not configure IP

Internal error

18) could not get_net_by_id\n");

Internal error

19) changestatus ioctl failed

Internal error

20) No memory to start frame relay connection for <name>

PortServer II does not have sufficient memory available for the Frame Relay connection.

21) <name> couldn't send wan close request

Internal error

22) Could not pwd_to_userinfo for <name> (memory problem?)

Internal error

23) connection manager received M_ERROR

Internal error

24) WAN couldn't connect interface to data stream

Internal error

25) WAN couldn't send message, port probably locked up!

A serial port may be stuck in one state. Try rebooting PortServer II to clear the problem. If the condition persists, contact Digi Technical Support.

26) WAN couldn't disconnect interface to data stream

Internal error

27) connect mgr received an illegal msg

Internal error

28) unable to open port <port> for frame relay

PortServer II could not set the port for Frame Relay because the port appeared to be busy.

29) port <port> for frame relay: I_LINK fails

Internal error

30) could not open wan driver

Internal error

31) memory not available for constructing filter

PortServer II does not have sufficient memory available for the required filter.

32) setting wan driver startup failed

Internal error

Dialer messages

(`Option=dialer`)

1) transferred to unknown state: <number>

A script tried to transfer to a stanza that is not defined in the script.

2) command depth overflow, state: <number>

Too many recursive state executions were attempted.

3) error in script: <scriptno> state <stanza> field <string>

Check the script for syntax errors.

Warning Trace Messages

ARP Messages

(`Option=arp`)

1) response for <IP address> ignored: used multicast address

The remote system's ARP response used a multicast address, so the mapping was ignored.

2) request timed-out for <IP address>

No ARP response was received, and PortServer II timed out.

DNS Messages

(`Option=dns`)

1) entry <name> time-out, removing from list

DNS mapping for **<name>** timed out.

Frame Relay Messages

(`Option=frame`)

1) LMI in error watch: data traffic may be stopped

An LMI fault condition has been detected. If the fault persists, data traffic may be stopped.

2) LMI again OK: data traffic can resume

The LMI fault condition that had been detected has now cleared. Data traffic will not be stopped.

3) LMI fault event: sequence number mismatch

A sequence number mismatch fault has occurred.

4) LMI async change: PVC <dlci> is now <state: active | inactive>

PVC **<dlci>** has changed state.

5) LMI status change: deleting PVC <dlci>

The DCE has informed PortServer II that the unused **<dlci>** is no longer valid.

6) LMI Status change: PVC <dlci> bandwidth is <bps>

An LMI Rev1 flow control message has been received to change the bandwidth to the value **<bps>**.

7) LMI status change: PVC <dlci> added

The DCE has informed PortServer II of a new PVC **<dlci>**.

8) LMI fault event: status response not received

A no status response fault event has occurred.

Telnet Messages

(`Option=telnet`)

1) Fail to turn of escape character recognition

Internal error

2) Fail to set Binary mode

Internal error

IP Messages

(`Option=ip`)

1) packet discarded from interface <name>: header error

An error was detected in the IP header from interface **<name>**. If the problem continues, use the Trace Debug command to identify the cause.

2) packet discarded from interface <name>: not a router

The PortServer II is not configured as a router and a packet to be routed was received.

3) packet discarded from interface <name>: source route failure

The PortServer II is not defined as a router and a packet to be routed using source routing was received.

4) packet discarded from interface <name>: TCP broadcast

A TCP broadcast message was received and discarded.

5) packet discarded from interface <name>: TCP header error

A TCP packet had a header error, probably a TCP checksum error, and was discarded.

6) packet discarded from interface <name>: UDP altip address

A UDP packet was received using an AltIP address and was discarded.

7) packet discarded from interface <name>: UDP header error

A UDP packet had a header error, probably a UDP checksum error, and was discarded.

8) packet discarded from interface <name>: unsupported protocol

A packet was received with an unknown protocol and was discarded.

9) interface <name> lower queue canput failed - again

Persistent congestion is occurring on interface **<name>**.

10) packet discarded to interface <name>: congestion failure

Congestion prevent transmits on interface **<name>**.

11) packet discarded to interface <name>: bad destination

The IP address of the destination is invalid and the packet was discarded.

12) packet discarded to interface <name>: bad source route

There was an error in the source route information and the packet was discarded.

13) packet discarded to interface <name>: fragmentation error

The fragment size defined is too small and the packet was discarded.

14) packet discarded to interface <name>: fragmentation not allowed

The interface requires packet to be fragmented, and the packet will not allow (further) fragmentation. The packet was discarded.

15) packet discarded to interface <name>: fragment broadcast

Packets to broadcast addresses cannot be fragmented, so the packet was discarded.

16) packet discarded to interface <name>: fragmentation data error

The packet to be fragmented was in error, so the packet was discarded.

Forwarder Messages

(`Option=fwdr`)

1) packet discarded on interface <name>: <reason>

The reason(s) for discarding the packet are detailed in associated Trace IP messages.

Routed Messages

(`Option=routed`)

1) adding dependency route for <index> (<IP address>) failed: out of memory

The static route identified by **<index>** could not be added to the Route Table due to a memory allocation failure.

2) adding dependency route for <index> (<IP address>) failed: invalid gateway

The static route identified by **<index>** could not be added to the Route Table because the gateway is unknown or invalid.

3) adding dependency route for <index> (<IP address>) failed: invalid metric

The static route identified by **<index>** could not be added to the Route Table because the metric was not 1 for a direct host address, greater than 3 for a direct net address, or 0 for an indirect address.

4) adding dependency route for <index> (<IP address>) failed: invalid address

The static route identified by **<index>** could not be added to the Route Table because the address is the same as the Ethernet's IP address.

5) adding dependency route for <index> (<IP address>) failed: invalid mask

The static route identified by **<index>** could not be added to the Route Table because the mask equals 0.

6) adding dependency route for <index> (<IP address>) failed: invalid interface name

The static route identified by **<index>** could not be added to the Route Table because the interface (WAN) name is unknown or invalid.

7) invalid route address family: <value>

A RIP packet entry was found with an invalid address family, and the entry was ignored.

3) invalid source address: <IP address>

A RIP packet sender was found with an invalid **<IP address>**, and the packet was ignored.

4) invalid length: <length> bytes

A RIP packet was found with an invalid length, and the packet was ignored.

5) invalid version: <version>

A RIP packet was found with the wrong version, and the packet was ignored.

6) invalid source port: <port>

A RIP packet's source port is wrong, and the packet was ignored.

8) invalid route metric: <metric>

A RIP packet entry was found with an invalid metric, and the entry was ignored.

9) invalid route address: <IP address>

A RIP packet entry was found with an invalid address, and the entry was ignored.

10) zero fields are non-zero

A RIP packet entry was found with garbage in zero fields, and the entry was ignored.

NetCX Messages

(`Option=realp`)

1) allocb failure during transmit

Available memory is low.

2) allocb failure during transmit:server

Memory is low.

3) failed to open server

Internal error

NETD Messages

(`Option=netd`)

1) IP address is unknown - RARPing

PortServer II will try to learn its IP address via RARP.

INETD Messages

(`Option=inetd`)

1) unexpected signal - <number>

Internal error

Serial Messages

(`Option=serial`)

1) (<port>) spurious interrupt

Internal error (possible hardware problem).

2) receive error - mask 0x<number>

Internal error

User Messages

(`Option=user`)

1) auto-connect-port: illegal user id on port <port>

The Autoconnect port UID field indicates a non-existent user.

2) time string error <string>

A user's access time string has a syntax error

3) Verify routine found no user name

Internal error

4) <name>: Autoconnect enabled with no host address.

The user was configured for autoconnection, but no host is defined.

5) <name>: Menu access enabled with no menu selected.

The user was configured for access via a menu but no menu is defined.

6) <name>: Menu number references undefined menu

The user is configured to use a menu that is not defined.

7) <name>: Syntax error in AccessTime string.

The user is configured for access only at certain times or on certain days, and there is a syntax error in the configuration.

8) <name>: Invalid subnet mask for dynamic/negotiated IP address.

For negotiated or dynamic IP addresses, WAN users must have an IP mask of 255.255.255.255.

9) <name>: Negotiated IP address can only be used with PPP.

It is not possible to used address negotiation with the currently configured protocol. Only the PPP protocol supports address negotiation.

10) <name>: Keepup filter ineffective with no idle timeout.

If you do not configure an idle timeout, there is no requirement for a keepup filter (the keepup filter resets the idle timer).

11) <name>: Bringup/Keepup/Passpacket/Logpacket filter not found.

One of the filters configured in the user's profile was not defined by **set filter**.

12) <name>: Must specify a device for outgoing connections

You cannot enable dial out without first defining a device to dial out on.

13) <name>: Device name not found in device table

The user's profile is configured to use a device that is not listed in the Device Table.

14) <name>: No frdlci information for port <port> dlci <dlci>

You must define an entry in the frdlci table to correspond with the port/dlci pair for this user.

15) <name>: Port <port> not configured for frame relay.

Use the **set frame** command to enable Frame Relay for this port.

RADIUS messages

(`Option=radius`)

1) Radius: Callback not supported, access denied.

The user's RADIUS configuration specifies callback, which is not supported by PortServer II.

2) Radius: Auth. only not supported, access denied.

The RADIUS server returned a service type of “Authorization only”, which is not supported by PortServer II.

3) Radius: Service not recognized, access denied.

The RADIUS server indicated a type of service that PortServer II does not understand.

4) Radius: Protocol not recognized, access denied.

A Framed Protocol value other than SLIP or PPP was specified.

5) Radius: Routing value not recognized, access denied.

An undefined Routing value was indicated by the RADIUS server.

6) Radius: Compression value not recognized, access denied.

An undefined Compression value was indicated by the RADIUS server.

7) Radius: Login service type not recognized, access denied.

An undefined Login-Service value was indicated by the RADIUS server.

8) Radius: Attribute type <number> not recognized, ignored.

An attribute has been found that is not defined by the RADIUS specifications. This attribute was ignored because “tolerate unrecognized attributes” has been turned on in **set radius**.

9) Radius: Attribute type <number> not recognized, access denied.

An attribute has been found that is not defined by the RADIUS specifications. Access was denied because “tolerate unrecognized attributes” has been turned off in **set radius**.

10) Radius: Login host missing, access denied.

The RADIUS server told PortServer II to log the user into a host automatically, but did not specify the host.

11) Radius: Need port for raw TCP service.

The RADIUS server told PortServer II to connect the user by a raw TCP connection, but did not specify the host.

12) Radius: Ippool can only be used with 255.255.255.255 netmask

The RADIUS server told PortServer II to give the user an IP address from the IP pool, but gave a netmask value that was not 255.255.255.255.

13) Radius: Must have service type; access denied

The RADIUS server returned an invalid packet. This indicates a problem with the RADIUS server, as the RADIUS specifications require a Service-Type to be present.

14) Radius: filter <name> not found, access denied

A filter specified by the RADIUS server to use is not defined in the Filter Table. Access is denied because this could lead to a security breach.

15) Request Denied <number>

The RADIUS server denied access

16) socket failed

Internal error

17) bind failed

Internal error

18) send malloc failed, recv malloc failed

Out of memory

PPP Messages

(`Option=ppp`)

1) (<port>) Bad LCP_CFG packet

Internal error

2) (<port>) Bad IPCP_CFG packet

Internal error

3) (<port>) bad auth configure packet

Internal error

4) cid(<port>): Dropping <n> bytes: packet is too large

Internal error (The higher level protocol tried to send a packet too large for the configured MTU).

5) cid(<port>): Dropping <n> bytes: Network layer is down

PortServer II was attempting to send data before a network connection is established.

6) cid(<port>) discard partial packet

A packet was discarded because it is too small to be a PPP frame. This can happen when PortServer II is shifting into PPP mode at the beginning of a connection.

7) cid(<port>) Dropping <n> byte packet - Network layer down

A data packet was received before PPP negotiations were finished.

VJ (Van Jacobsen) Messages

(`Option=IP`)

1) IP Header Short

A packet was discarded because it is too small to be an IP packet.

2) IP header with no TCP payload

A packet was discarded because it appears to be an IP packet carrying TCP data, yet is not long enough to hold the TCP header.

13) Cid(<port>) unknown message type (<num>) ignored

Internal error

Wan Messages

(Option=wan)

1) IP Pool out of addresses, could not start <name>

More users are trying to use dynamic IP addresses than there are addresses in the IP pool.

2) <name>: No phone numbers

The user is configured to dial out, but there are no phone numbers listed in his or her profile.

3) login to remote failed for <name>

The script that logs the user into the remote system failed.

4) login script <scriptname> not found for <name>

A script that the user is configured to use is not defined.

5) Network connection <name> not started: verify failed.

The user is incorrectly configured. Use the **wan verify** command to find out why.

6) IP Pool out of addresses, could not start <name>

The user require an IP address to start, but there are none left in the IP Pool.

7) <name> attach failed, dropping connection (port <port>)

Internal error (The user could not be connected to the IP forwarding mechanism).

8) <name>: unexpected getmsg failure: <errno>

Internal error

9) <name>: No dlci information for port <port> dlci <dlci>, aborting

There is no information in the **set frdlci** table for this port/dlci pair. An entry in this table is necessary to create a Frame Relay connection.

10) <name>:ioctl failed

Internal error

11) <name> failed verify

The user is incorrectly configured. Use the **wan verify** command to find out why.

12) <name>: bad value for access time

The access time during which the user can start a connection has a syntax error.

13) can't create route for interface=<name> (<reason>)

This message indicates a route cannot be set up for this user (probably an internal error).

14) adding route for <name> (<network>) failed(<reason>)

The route indicated by the user's IP mask could not be created.

15) Connection <name> aborted; port <port> not set up for Frame Relay

Frame Relay is not enabled on the named port (see **set frame**).

16) no available port to start connection <name>

All ports that this connection is configured to use are busy.

17) <name>: verify failed, no outgoing network created

The user is incorrectly configured. Use the **wan verify** command to find out why.

18) <name>: interface already connected, failing new connection

An attempt was made to connect the same user twice.

Dialer messages

(`Option=dialer`)

1) receive failure while executing script <number>

Internal error

2) no phonenumber for script <name> state <num>, field <string>

A script requested a phone number when none has been configured.

3) chat parameters not defined for script <name> state <num>, field <string>

A script tried to use chat parameters where none have been defined.

Tracing the Route to a Host

Many problems that you will encounter when using PortServer II in a network may be due to faults with external systems. This is especially true if you use PortServer II to connect to the Internet, where data may pass through many systems between source and destination. To assist in identifying problems with intermediate Gateways, PortServer II includes a **traceroute** command.

This command attempts to trace the route that an IP packet follows to an Internet host. It identifies intermediate hops by sending probe packets with a small TTL (Time To Live) and listening for an ICMP **Time Exceeded** reply from each Gateway. The first three probe packets that are sent have a TTL of one, then this value is incremented by one for each subsequent group of three packets. When PortServer II receives an ICMP **Port Unreachable** reply, a probe packet reached the host or the maximum of 30 hops was reached.

The address of each responding gateway is logged, together with the time to reply to each probe packet. If there is no response within the five second timeout interval, a "no response line is printed ("* * * ").

Probe packets are in UDP format, with the destination port set to a value that should ensure the Gateway host does not process them.

The syntax for the **traceroute** command is:

```
traceroute <host>
```

where <host> may be the name or the IP address of the target Internet host.

Examples of printout generated by the traceroute command

The following example includes replies from all intermediate Gateways:

```
traceroute nis.nsf.net
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 56 byte
packet
1 helios.ee.lbl.gov (128.3.112.1)  19ms  19ms  0ms
2 lilac-dmc.Berkeley.EDU (128.3.216.1)  39ms  39ms  19ms
3 lilac-dmc.Berkeley.EDU (128.3.216.1)  39ms  39ms  19ms
4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39ms  40ms  39ms
5 ccn-nerif22.Berkeley.EDU (128.32.168.22)  39ms  39ms  39ms
6 128.32.197.4 (128.32.197.4 )  40ms  59ms  59ms
7 131.119.2.5 (131.119.2.5)  59ms  59ms  59ms
8 129.140.70.13 (129.140.70.13)  99ms  99ms  80ms
9 129.140.71.6 (129.140.71.6)  139ms  239ms  319ms
10 129.140.81.7 (129.140.81.7)  220ms  199ms  199ms
11 nic.merit.edu (35.1.1.48)  239ms  239ms  239ms
```

Note: Lines 2 and 3 are the same. This may be due to a fault on the second system, causing it to forward packets with zero TTL.

The following example shows the result if some gateways do not respond to probe packets:

```
traceroute allspice.lcs.mit.edu
traceroute to allspice.lcs.mit.edu (18.26.0.115),30 hops max
1 helios.ee.lbl.gov (128.3.112.1)  0ms  0ms  0ms
2 lilac-dmc.Berkeley.EDU (128.3.216.1)  19ms  19ms  19ms
3 lilac-dmc.Berkeley.EDU (128.3.216.1)  39ms  19ms  19ms
4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 19ms  39ms  39ms
5 ccn-nerif22.Berkeley.EDU (128.32.168.22)  20ms  39ms  39ms
6 128.32.197.4 (128.32.197.4 )  59ms  119ms  39ms
7 131.119.2.5 (131.119.2.5)  59ms  59ms  39ms
8 129.140.70.13 (129.140.70.13)  80ms  79ms  99ms
9 129.140.71.6 (129.140.71.6)  139ms  139ms  159ms
10 129.140.81.7 (129.140.81.7)  199ms  180ms  300ms
11 129.140.72.17 (129.140.72.17)  300ms  239ms  239ms
12 * * *
13 128.121.54.72 (128.121.54.72)  259ms  499ms  279ms
14 * * *
15 * * *
16 * * *
17 * * *
18 ALLSPICE.LCS.MIT.EDU (18.26.0.115)  339ms  279ms  279ms
```

Note: Gateways 12 and 14 through 17 did not respond to the probe packets. Typically, this occurs if the host does not send ICMP **Time Exceeded** messages, or send them with a TTL that is too small to reach PortServer II.

Troubleshooting Frame Relay

Problems with a Frame Relay link generally fall into one of two categories:

- The link does not come up
- The link appears to start, but does not route or pass packets

To identify the source of the problem, we suggest you do the following:

1. Check for critical errors since PortServer II was initialized by reading the Trace Log. Enter the following command to print the Trace Log:

   ```
   set trace state=dump
   ```

   Refer to the *Command Reference Guide* for more information on the **set trace** command.

2. If the Trace Log does not provide any assistance in identifying the problem, check the current state of PortServer II. You can do this by setting the Trace mode to **concurrent** with the following command:

   ```
   set trace mode=concurrent
   ```

   Frequently, the process that is failing to run will retry periodically. If you can identify this process, you may be able to isolate the location of the fault.

3. Display the counters for the Frame Relay links, by entering the following command:

   `info frame :*` (to display the counter for all Frame Relay links), or

   `info frame :*:*` (to display the counter for all PVCs)

   Note: The best option to select depends on the number of Frame Relay links and the extent of the fault. Select the option that gives you sufficient counters for meaningful interpretation.

   In either case, both sent and received frames should be displayed with no errors (Err) and error fault (ErrFaults) counts, or a small number of errors.

4. If you require more information, you can display detailed debug commands by entering a command similar to the following:

```
set trace frame:debug, wan:debug, user:debug
```

The detailed messages displayed should indicate the source of any problems with the Frame Relay link. If you require help in interpreting the debug messages displayed, contact Digi Technical Support.

Chapter 18

Digi Support Services

When should I read this chapter?

Read this chapter if you have a problem with PortServer II and need advice from Digi International support personnel, or if you want to download software or information.

In This Chapter

This chapter describes Digi International's support services. It covers the following topics:

Web Server: Access to Digi Information

Purpose

The Digi Web server provides you with access to product information, manuals, new product announcements, programs, and application stories.

URL

You can access the Web server at **http://www.dgii.com**.

Internet FTP Server: Access to Digi Drivers

Purpose

The Digi anonymous FTP server provides you with access to Digi drivers and related installation information.

Address

You can access the FTP server at ftp.dgii.com.

Tips on Using the FTP Server

When you access the Digi FTP server:

- Log in as `anonymous`.
- Enter your E-mail address when asked for a password.
- Locate drivers and related installation tips in the appropriate operating system/board type directory.
- Enter `bin` before downloading any driver files to ensure proper binary transfer.
- Enter `ASCII` before downloading ASCII text files.
- See the text file *download.doc* for information on uncompressing files after downloading.

Digi BBS: Access to Drivers and Information

Purpose

The Digi electronic bulletin board service (BBS) provides the following:

- Access to general and technical information about Digi's products
- Access to the most recent software driver updates and upgrades
- An opportunity to leave messages for Technical Support regarding Digi products.

Modem Support

The Digi BBS supports the following:

- Line speeds of 1200, 2400, 9600, 14,400, and 28,800 bps
- V.32, HST 14.4, V.42 and V.42bis standards, with full MNP class 15 error correction and data compression.
- Modem settings of eight bits, no parity and one stop bit (8 N 1), though other settings may also work.
- Zmodem, Xmodem, Ymodem, Kermit and other download protocols.

Telephone Numbers

Here are the telephone numbers for access to the BBS:

- In North America: (612) 912-4800
- In Europe: +49 221 920 5211
- In Asia: +65 735 2460

FaxBack Server: Information by fax

Purpose

The FaxBack server provides you with manuals and technical information by fax.

How to Use the FaxBack Server

Call (612) 912-4990 on a touch-tone telephone and listen for instructions.

Customer Service

Purpose

Digi's staff of customer service representatives can help you with the following:

- Software and documentation update requests
- Returned Merchandise Authorizations (RMAs)—if you need to return your Digi product for repair.

How to Reach Customer Service

To reach Customer Service, do any of the following:

- Telephone (612) 912-3456
- Fax (612) 912-4959
- Send e-mail to cust_serv@dgii.com (Please include your telephone and fax numbers.)

Return Procedures

Warranty Information

Digi products have a five-year parts and labor warranty, and Digi assumes responsible for any defective parts, according to the limits specified in the warranty. However, many problems are due to factors other than product defects. To save you time and possibly additional cost, Digi asks that you first try to resolve difficulties by contacting our technical support representatives at (612) 912-3456.

Return Procedure

To return your Digi product for repair:

- Obtain an RMA (Returned Merchandise Authorization) number from a Digi customer service representative.
- Place the RMA number on the shipping carton, on or near the address label.
- Ship authorized returns to Digi International, 10000 West 76th Street, Eden Prairie, MN 55344, USA.

Technical Support

Introduction

If you experience difficulty with a product, Digi has a staff of technical support specialists to assist you.

Support Process

Follow this process to resolve the problem with Digi products:

- First, contact your Digi dealer or distributor. They provide first-level technical support and have the training to help you with any installation questions or difficulties you may have.
- If you still experience difficulties (after contacting first-level support), contact Digi Support Services.

When You Call Technical Support

When you call to Technical Support, please call from a position where you can operate your PortServer II and management terminal.

How to Contact Digi Technical Support

To reach Digi Technical Support:

- In USA: Telephone (612) 912-3456; Fax (612) 912-4958; E-mail support@dgii.com
- In Europe: Telephone +49 221 920-5200; Fax +49 221 920-5210; E-mail support@dgii.com
- In Asia: Telephone +65 732 1318; Fax +65 732 1312; E-mail support@dgii.com

Glossary

10BaseT	IEEE 802.3 specification, using twisted pair wiring and RJ-45 connectors.
10Base2	IEEE 802.3 specification network, using coaxial cables and BNC connectors. Sometimes called “thin Ethernet” or “Thinnet”.
Address	A data structure used to identify a unique entity, such as a particular process or network.
ALTPIN	A software feature that makes the DCD signal available on pin 1 of an eight pin RJ-45 connector, allowing use of eight pin RJ-45 to DB-25 adapters.
Annex D	Annex D of the ANSI T1.617 Frame Relay protocol specification.
ARP	Address Resolution Protocol. ARP maps hardware network addresses to external IP addresses.
Asynchronous transmission (async)	Operation of a network on which events are initiated by a clock signal. In such instances, individual characters are encapsulated in control bits called *start* and *stop* characters.
Baud	Signaling speed equal to the number of signal events per second. *Baud* is the same as *bits per second* if each signal event represents exactly one bit.
Bc	Committed Burst rate. See *Chapter 14, Configuring Frame Relay.*
BCN	Backward Congestion Notification. See *Chapter 14, Configuring Frame Relay.*
BECN	Backward Explicit Congestion Notification. See *Chapter 14, Configuring Frame Relay.*
BootP	Bootstrap Protocol.
Broadcast packets	Packets that are sent to all devices on a network.
BPS	Bits per second. See also *Baud*.

CFM	Cubic Feet per Minute
Channel	A communication path. Multiple channels can be multiplexed over a single cable in certain environments.
CHAP	Challenge Handshake Authentication Protocol. A method of authenticating a connection request on a PPP port. See also PAP.
CIR	Committed Information Rate. See *Chapter 14, Configuring Frame Relay.*
CSLIP	Compressed Serial Line Interface Protocol. This is *SLIP* data with Van Jacobsen compression applied to the TCP packets headers.
CU	A command on Unix systems that allows you to log in to and use another system, while remaining logged into your home system.
Dial-up connection	Communications circuit that is established by a switched circuit connection using the telephone network
DLCI	Data Link Connection Identifier. Used on Frame Relay networks to identify a particular virtual connections. A DLCI identifies a bi-directional circuit, not a destination or source. It may be changed by each Frame Relay switch that handles traffic associated with the connection. See *Chapter 14, Configuring Frame Relay.*
DNS	Domain Name Service. A method of providing network devices with names, rather than numerical addresses.
EBI	External Bus Interface, a Digi International standard for connecting external modules to devices such as PortServer II.
Ethernet	A baseband LAN, operating at 10 Mbps using CSMA/CD to operate over coaxial cable. Ethernet is similar to *IEEE 802.3* standards.
FCC	Federal Communications Commission.
FCN	Forward Congestion Notification. See *Chapter 14, Configuring Frame Relay.*
Flash ROM	A non-volatile storage device that can be electrically erased in-circuit and reprogrammed.

Frame	The smallest unit of data on a network. Each frame contains source and destination addresses, length field, data, and checksum elements.
Frame Relay	A network interface standard derived from narrowband ISDN technology, in which *frame*s of data are relayed on a public frame relay service. See *ANSI Standard T1.606* or *ITU-T (CCITT) Standard I.233* for more information.
FTP	File Transfer Protocol. An IP application protocol for transferring files between network devices.
ICMP	Internet Control Message Protocol. A subset of the Internet Protocol (IP) that handles error monitoring and testing.
Intranet	A network that is based on the Internet architecture, but is restricted to a single site or company.
IP	Internet Protocol. See *RFC791* for more information.
LAN	Local Area Network. A network covering a relatively small geographic area, usually not larger than a small building. LANs usually provide a relatively high data rate and low error rates.
LCP	Link Control Protocol. Used on PPP connections for testing and configuring the link.
Link	A network communications channel consisting of a circuit or transmission path, including all equipment, between a sender and receiver.
LMI	Local Management Interface. A protocol for managing Frame Relay connections, for example, *ANSI T1.617 Annex D* or *ITU (CCITT) Q.933 Annex A*.
MIB	Management Information Base. The database of parameters that an SNMP server can read from an SNMP-compatible device such as PortServer II. See *RFC1213* for more information.
Modem	Modulator-demodulator. A device that converts digital signals for transmission over analog telephone lines, and vice versa.
NIS	Network Information Services.
Packet	A logical grouping of data in a transmission that normally includes a header and user data.

PAP	Password Authentication Protocol. A method of authenticating a connection request on a PPP port. See also *CHAP.*
Ping	A message sent to test the reachability of a network device. Also (as a verb), to send a message to a device and wait for an acknowledgment.
Port	An interface on PortServer II that is associated with a serial connector.
PPP	Point-to-Point Protocol. See *RFC1331* for more information.
PSTN	Public Service Telephone Network.
PVC	Permanent Virtual Circuit. Used in Frame Relay networks to provide a logical connections between two points.
RADIUS	Remote Authentication Dial In User Service. A security authentication protocol developed by Livingston Corp.
RARP	Reverse Address Resolution Protocol. A method by which a network device such as PortServer II can "learn" its IP address from a server.
RealPort	A Digital International protocol that allows a host to control the operation of serial ports on a device such as PortServer II.
RFC	A *Request for Comment* document on a specific networking topic. Refer to the note at the end of this Glossary for details of how to obtain RFCs.
RIP	Routing Information Protocol. See *RFC1058* for more information.
RJ-11, RJ-45	Standard North American pattern telephone and data connectors.
Rlogin	A method of logging on to a remote system via another computer. Generally available on UNIX systems.
RS-232C	A physical interface standard, also referred to as *V24*.
RTTY	Software that allows a pseudo-TTY to act as a serial device, permitting the use of CU and UUCP.
SLIP	Serial Line Internet Protocol. A protocol employed on dial-in or dial-out connections.

SNMP	Simple Network Management Protocol, a method of administering devices that are connected to an IP network. See *RFC1157* for more information.
TCP/IP	Transmission Control Protocol/Internet Protocol. TCP corresponds to Layer 4 of the OSI model, the data transport layer. IP corresponds to Layer 3 of the OSI model, the network layer. See *RFC793* and *RFC791* for more information.
Telnet	A command used to establish a connection between two computers. See *RFC854* for more information.
Terminal emulator	Software that allows (typically) a PC to function as a terminal.
TFTP	Trivial File Transfer Protocol. See *FTP*.
TTY	A Teletype standard compatible terminal.
Twisted pair wiring	Unshielded, twisted pair wiring is typically used in telephone systems. It is also compatible with the 10BaseT Ethernet standard.
UDP	User Datagram Protocol. A method of multiplexing IP data. See *RFC768* for more information.
UL	Underwriters Laboratories
UUCP	Unix-to-Unix copy.
WAN	Wide Area Network, a network spanning a large geographical area.
WWW	World Wide Web. Areas of the Internet that are accessed by the HTTP protocol.

Note: "RFCs" are Request For Comment documents relating to networking subjects. You can download an RFC from any of the following Internet sites:

- **DS.INTERNIC.NET**. Directory `/ftp/rfc`
- **NIS.NSF.NET**. Directory `internet/documents/rfc`
- **NISC.SRI.COM**. Directory `rfc`

RFCs are typically maintained as text files identified by the number of the document, for example, **rfc0768.txt**.

Index

T

U

V

W

WWW: http://www.dgii.com

90030500B