OpenAir 400 Reference

Production

This manual was written, edited, and produced by:

Connect, Inc.
1701 Quincy Avenue
Suites 5 & 6
Naperville, IL 60540
www.connectrf.com

Printed in the U.S.A.

Please let us know about any errors in this document at:

http://207.241.78.223/isoxpert/calltrak.nsf/WebTracking?OpenForm

Table of Contents

Chapter 1 • Getting Started

Product Overview

PowerNet products are devoted exclusively to making wireless networks as useful, efficient, and reliable as possible. PowerNet routes data from any portable RF terminal to any host computer quickly, efficiently, and without fail, often by placing the processing load on a server. Many (300+) wireless terminals can operate in a real-time environment without special hardware or customized operating systems.

PowerNet supports:

- All of the most popular 2.4 GHz and 900 MHz wireless systems.
- All major network media, including Ethernet and Serial.
- All major transport protocols, such as TCP/IP.
- All common legacy terminal emulators, including VT100, VT220, 3270, and 5250.
- Multiple dissimilar hosts, using your existing applications and networks.
- Multiple RF LANs, allowing a single controller to support multiple locations, covering such large areas as airport terminals and distribution centers.
- Multiple portable terminal models of American Microsystems Ltd., Casio, Citadel, Compsee, Datalogic, Denso, Fujitsu, Hand Held Products, Intelligent Instrumentation, Intermec/Norand, LXE, Paxar/Monarch, PSC, Psion Teklogix, Symbol, Telxon, and Unitech (with more being added every day).

The PowerNet family of products works with almost all architectures, providing users the ability to change host computers and options as network requirements grow.

- OpenAir Linux - runs on Linux and Windows.
- OpenAir Windows - runs on Windows NT/2000.
- OpenAir 400 - a version of OpenAir Windows that loads directly on a customer's integrated XSeries server AS400 host.
- PowerNet Twin Client - software for virtually all makes and models of wireless data collection terminals. It supports all common terminal emulations and client/server applications.

PowerNet OpenAir provides the user with a fully-integrated hardware Box with either OpenAir Linux or NT/2000 software already integrated and installed, ready to plug in and use.

PowerNet AirLinc provides the user with the ability to develop complete wireless data collection applications that operate with any ODBC compliant database.

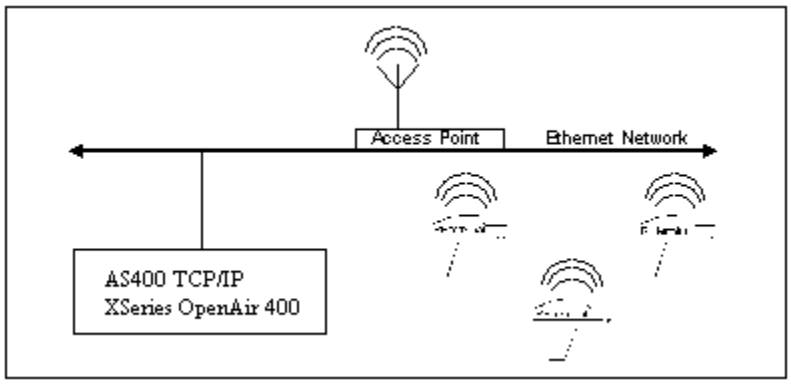
OpenAir 400 Overview

OpenAir 400 is an AS400 resident server product for RF systems. OpenAir 400 can provide 5250, 3270, VT100, VT220, and HP700/92 emulation directly on the host computer, XSeries Server. It is used in conjunction with Connect's award-winning Twin Client terminal emulation software and/or the ODBC Stored Procedure Application generator.

OpenAir 400 enables large terminal count RF systems to be connected directly to the AS400, eliminating the need to use a separate server between the host and the terminals. Configuration of the client is done via Connect's Twin Client Manager.

This new offering is ideal for AS400 installations, and all features of other OpenAir products are supported, including the ability to hot key to up to four dissimilar host sessions.

The following drawing represents OpenAir 400 linking networks.



OpenAir 400 Features

OpenAir 400 offers many additional benefits in addition to functioning as an RF network controller and host computing environment bridge.

- **Intelligent Software Design** – The OpenAir 400 server software with its modular design takes full advantage of the inherent support of multiple host processes. A newly developed DLL and ActiveX interface allows for quick implementation of customer applications for remote RF devices without the need for development of custom client software.
- **Support for Access Point RF Networks** – Using TCP/IP protocols for supporting access point devices combined with the OpenAir 400 server's data management resources provide unmatched performance for large numbers of users in an RF environment.
- **Client Streaming** – The implementation of client streaming programming methodologies using the API or ActiveX offer efficient use of RF network bandwidth and maximum battery life. Now, you can design applications to use processing capabilities of the remote terminals allowing terminals to transmit data only when necessary to complete a transaction. The OpenAir 400 server offers unmatched options for implementing connectivity to a variety of host computing networks in a client streaming environment. Terminal emulation offers easy set up and wireless implementation using existing legacy systems and applications.

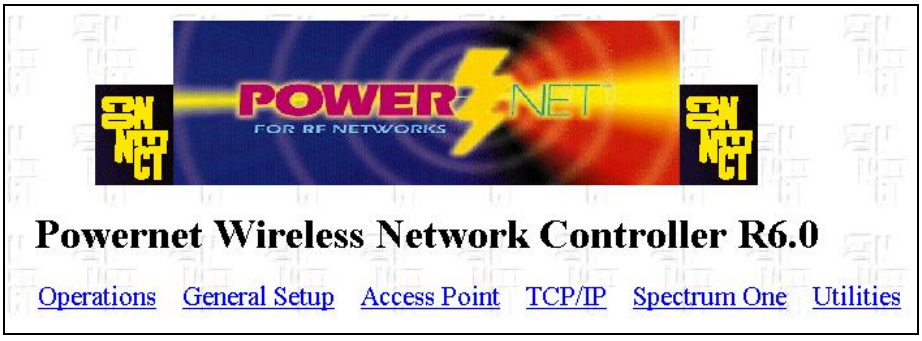
- **Statistical Trend Analysis** – The System Accounting Facility (SAC) and the Performance Facility (PRF) continually collect RF Network and terminal data, which is saved in files and can be viewed as reports on screen. You can use this statistical data to check system-operating trends. You can export data to a spreadsheet or use it with programs designed for analysis.
- **Diagnostics/Logs** – Remote diagnostics are available to the users' support staff so that immediate diagnosis can begin upon receiving a problem report. You can set logging to report different levels and types of information and to report at variable time intervals. Logging generates reports as the system runs.
- **Ease of Use** – The Internet browser-driven program shields the installation and support personnel from the operating system. When certain files require modification, the program calls and opens the proper file. It uses forms to provide information inserted into the file.
- **Fault Resilience** – Since the end-user chooses which points to cover if there is a component or software failure, the system may be termed *fault resilient*. Further, multiple OpenAir 400 servers can be offered with multiple levels of fault resiliency. In addition, analysis by systems personnel helps to define, design, and implement resilient RF systems. These systems include not only the OpenAir 400 server, but RF networks and connections to host(s) network environments as well.
- **Centralized control of terminal functionality** – OpenAir 400 allows system administrators full control and support of remote RF devices from one system. Administrators can have full control of system tuning, functionality, and presentation parameters used in supporting remote RF devices.
- **Sample applications** – The *Samples* directory contains application code using the API/DLL (application program interface.)

PowerNet User Interface Structure

The OpenAir 400 Networks program uses three screen formats – menus, forms, and logs. At program startup, an opening screen containing a menu of hot, clickable items appears.

Opening Screen

From Windows, with your mouse, select **Start**, **Program**, and **PowerNet**. The **Main Menu** screen appears.



The program title, PowerNet, centered at the top, is followed by such menus as Operations, Access Point, and TCP/IP. (Your screen may vary according your installed packages and options.)

Menus

Options appear on the opening screen depending upon which software modules make up the system. The opening screen components discussed in this manual, **Operations**, **General Setup**, **Access Point**, **TCP/IP**, **Spectrum One**, and **Utilities**, are part of *all* OpenAir 400 systems. These functions are used to set up and maintain the RF LAN(s) and provide system management. Subsequent chapters of this manual explain these components and their related options. Related connectivity manuals discuss selections unique to specific connectivity packages.

Forms

Many Windows screens require entry into forms, which are useful for collecting information. Fields appear on forms in which you must select from various fixed choices or enter other information from the keyboard. Conventional Windows selections indicate whether a field is available from a pull-down, a selection box, a radio button, or requires specific keyboard input. This input may be a new field name or the address of a specific network component for which statistics are displayed.

Logs

System logs are dynamic and use their own format. The information, which is constantly being written to a log file, is displayed on the screen when that file is selected. The latest information appears on the last line of the log. To exit from the log display, use the mouse to point and click on the browser window.

Keys and Buttons

Certain keys move the cursor on a menu or form as indicated below. The left mouse button is used for point and click functions, and the right mouse button gives submenus to certain selections such as the **Menu**, **User**, **Screen**, and **Procedures** selections from the **Project Manager Screen**.

Key/Button	Function
<↑>	The **Up arrow** moves the cursor backward one field position in a form.
<↓>	The **Down arrow** moves the cursor forward one field position in a form.
Bksp	When the **Bksp** key is used in fields within a form, it moves the cursor backward one character space, deleting the character.
Del	When the **Del** key is used in fields within a form, it moves the cursor forward one character space, deleting the character.
Tab	Use the **Tab** key to move between fields on a form screen.

In This Manual

The OpenAir 400 manual consists of eleven chapters. Each chapter describes in detail various aspects of the software. The manual also contains sample applications provided for your reference.

- Chapter 1 provides an overview of the OpenAir 400 software and describes the documentation conventions used within this manual.
- Chapter 2 covers the requirements and procedures for installing the system software on the Internet server and explains the standard menus and submenus.
- Chapter 3 explains how to navigate throughout OpenAir 400.
- Chapter 4 covers the requirements and procedures for installing Access Point LAN software and explains the Access Point menu in detail.
- Chapter 5 covers the requirements and procedures for installing Spectrum One LAN software and explains the Spectrum One menu, diagnostics, and activity report in detail.
- Chapter 6 discusses administration and maintenance of Wireless View and Vterm Handler.
- Chapter 7 covers RF networks in terms of cabling, multiLANs, WANs, and additional pertinent technology.
- Chapter 8 discusses Windows NT/2000 ActiveX control, and provides routines for output, extensions, and Windows NT/2000-specific action.
- Chapter 9 lists Windows NT/2000 DLL interface, and provides general, output, extended, and Windows NT/2000-specific routines.
- Chapter 10 covers Ethereal, an open source packet sniffer available from the WEB.
- Chapter 11 discusses TightVNC, a client/software package allowing remote network access to graphical desktops.

This page is intentionally blank.

Chapter 2 • System Installation

This chapter describes the necessary steps for installing OpenAir 400 server software, and summarizes the program menu structure. Included with the OpenAir 400 server are utilities that verify system software, authorize installed software, and capture diagnostic data among others. Later sections of this document discuss these special utilities.

Minimum Installation Requirements

Hardware Requirements:

- Models 170-180x
- Minimum 384 MB Ram
- 40 MB DASD
- 2838 Ethernet Port or 2742 Token-Ring Port
- 4810 PCI INTEG XSeries Server (cost $1,900)

Note: Most AS400's are configured with this option.

Software Requirements:

- OS/400 V4R5, V5r1 or V5R2 with the latest cumulative PTF fix package applied
- 5722WSV Integration for Windows Server
- Windows 95, 98, NT, 2000 or XP

Network Requirements:

- Wireless Access Point system
- TCP/IP access to the AS400

Checking for Microsoft Internet Info Server

To check to see if the Microsoft Internet Information Server programs are installed, use the following steps.

1. Click on **Start**, **Settings**, and **Control Panel**, and then select the **Network** icon on the Control Panel.
2. Click on the **Services** tab to see if the Internet Server programs are installed.

a. If you see the Internet server in this list, proceed to the PowerNet program installation section below. (On an NT/2000 workstation, the Internet Server appears as Microsoft Peer Web Server.)

b. If the Internet server does not appear, install Internet server programs as follows:

1) Insert the Windows NT/2000 server CD in the CD ROM drive.

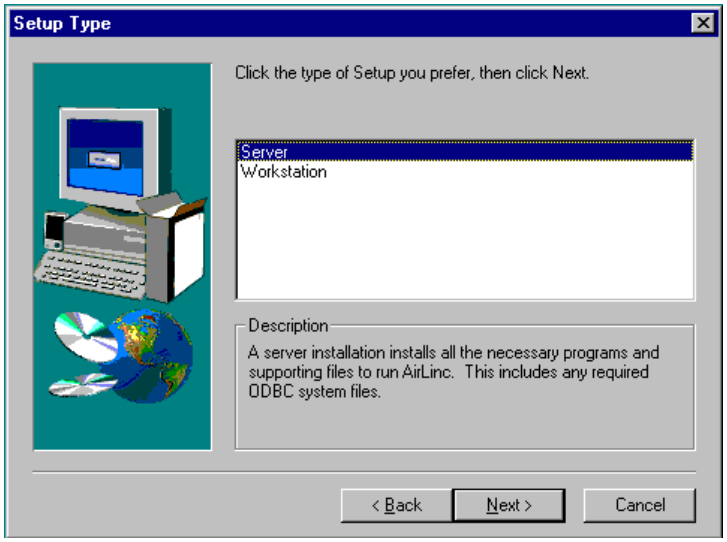
2) While on the Services tab of the Network Control Panel applet, click on **Add**. The system displays a list of available services.

3) Select the Internet Server service to add. Consult the Windows NT/2000 Server Information manual for more information regarding the setup parameters for the Internet Server program.

Note: It is critical that the World Wide Web service be installed. FTP and Gopher are not required.

Installing OpenAir 400 Server for Windows NT/2000

1. Insert the OpenAir 400 CD into the CD-ROM drive, or disk 1 of the installation disk set into the appropriate drive of the PC or AS400.
2. Click on **Start** and **Run** and type **x:\setup** at the prompt. ("**x**" is the drive letter for the drive containing the installation media.)
3. Click on **Run** to begin the installation process. The installation program runs an installation wizard and then presents the Setup screen.



Note: Throughout the installation, click on the **Next** button to continue, or click on the **Cancel** button to exit the installation process.

4. Enter your Installation Key (identification number) when asked for it.
5. You are prompted for a target folder (default is **/PowerNet**). If you choose to change the default folder, click on **Browse**.

6. You are prompted for the name of a program group (folder) into which to install the PowerNet software. The default group is **PowerNet**. You may change this default either by selecting an existing program group or by typing in a new name at the prompt.
7. The PowerNet installation program begins copying the files to the target folders. Insert additional disks as prompted.
8. When the installation program is complete, you must reboot the system to initialize the PowerNet server software. (The system must be restarted in order for the PowerNet software to function correctly.)
    a. To reboot the system immediately, select **Finish**.
    b. To reboot the PowerNet server software later, select the **Restart Later** option, and click on **Finish**.

These steps complete the PowerNet Server software installation process.

This page is intentionally blank.

Chapter 3 • Navigating OpenAir 400

This section explains the OpenAir 400 menu structure and the relation of submenus to the **Main Menu**. A summary of all menus and options appears on the following pages.

Note: The **Main Menu** is common to all systems, although some options shown here may not appear on your system based on the host connectivity and system options installed on your OpenAir 400 server controller.

OpenAir 400 Standard System Map

The system menu map, below, shows the relationship of the submenus to the **Main Menu**. Note that some options shown here may not appear, depending on the options you selected during installation.

Main Menu

Operations Menu
- Access Point Startup
- Access Point Shutdown
- Spectrum One Startup
- Spectrum One Shutdown

Access Point Menu
- Server
- MAC to IP Listing
- AP Discover
- AP Config
- Diagnostics
- Performance Report

Spectrum One
- Network Setup
- Terminals
- Transceivers
- Activity Report
- Performance Report (tabular)
- Performance Report (graphical)
- Frequency Scan Report

General Setup
- Host List
- Terminal Control
- Ports

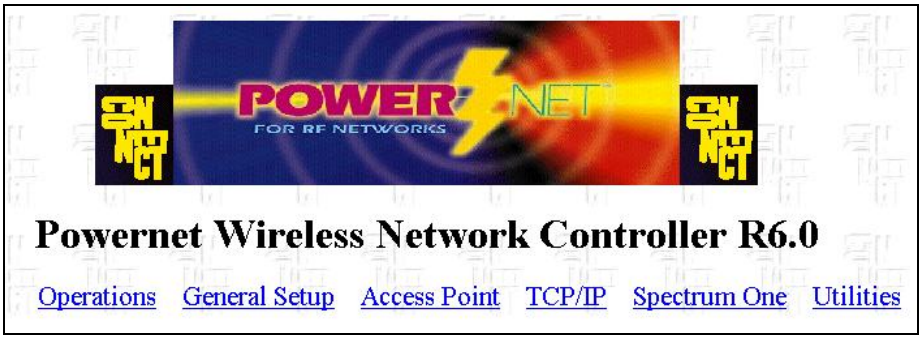
VTerm Serial Setup

TCP/IP Menu
- Addresses
- Network Status
- Ping Test

Utilities Menu
- Verify
- Authorize
- Snapshot
- Cleanup

Using the Menu System

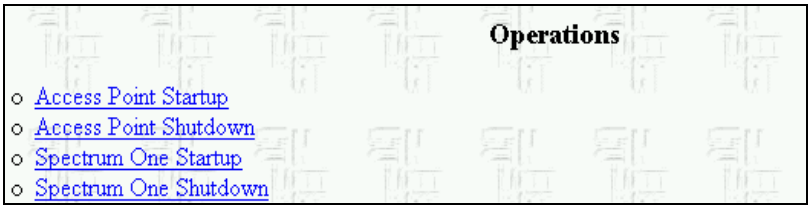
The **Main Menu** accesses the major functions within OpenAir 400. The program title, PowerNet Wireless Network Controller, appears in the top center, followed by the active project name.



Hot, clickable items (hyperlinks) appear below the title, for example, Operations or Access Point. The menu choices from the **Main Menu** are used for creating, opening, closing, copying, deleting, exporting, and importing. The following menus/options are available from the **Main Menu**.

Starting-up and Shutting-down PowerNet

The **Operations Menu** provides access to menu choices used for starting-up and shutting-down the OpenAir 400 server RF network(s).



Access Point Startup — Activates RF Server connection activity to the OpenAir 400 server to communicate with access points. This is for Access Point startup only.

Opens up the New Project screen for the creation of a new project. The New Project becomes the active project shown next to the AirLinc title. Entry of the Project Name, Author, RF Terminal Screen Size, and any Remarks to be associated with the new project are done on this screen.

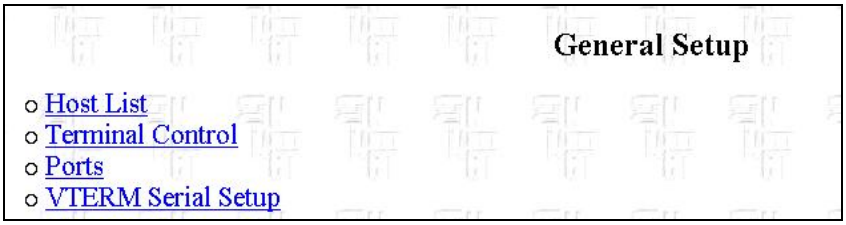
Access Point Shutdown — Stops all activity to the OpenAir 400 from the access points.

Spectrum One Startup — Activates RF Server connection activity to the OpenAir 400 server to communicate with Spectrum One.

Spectrum One Shutdown — Stops all activity to the OpenAir 400 server in the Spectrum One connectivity environment.

Configuring and Enabling

The **General Setup** menu includes Host List, Ports, and VTerm Serial Setup.



Host List Allows system administrators to configure the hosts, applications, and functions that are available to the RF terminals.

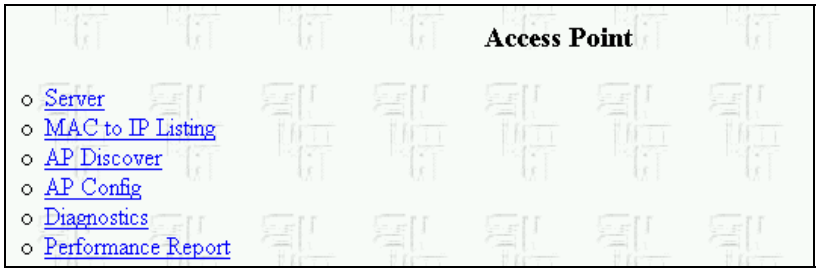
Terminal Control Allows host list entries to be enabled/disabled based on the type of terminal being used.

Ports Designates the serial ports available for use by the OpenAir 400 server.

VTerm Serial Setup Allows the user to set port number, terminal name, speed, and DPS using drop down windows.

Identifying and Connecting Access Points

The **Access Point** menu includes RF Access Point network utilities used for setting and managing terminals and access points, and generating reports.



Server Configures Access Point server used to set log levels, IP addresses, etc.

MAC to IP Listing Displays a list of known terminal and access point MAC addresses and their assigned IP addresses that were reported by the OpenAir 400 server as attaching to the Ethernet network.

AP Discover Runs the discovery program, which identifies RF access points to the system for configuration and topology viewing.

AP Config Establishes a telnet session with active Access Point. This connection provides access to all control and setup functions available on the AP.

Diagnostics Pings the Access Point networks, signaling possible problems.

Performance Report Creates system reports, including transaction data, in tabular form.

Communicating with TCP/IP

The **TCP/IP** menu provides for the selection of addresses and network status.

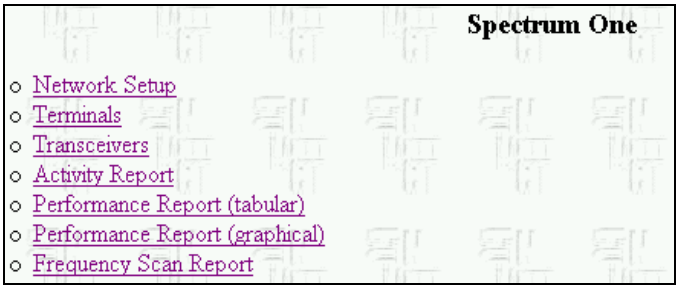
Addresses Lists names and addresses of host nodes, with which the OpenAir 400 terminals establish sessions. This setup form is for communication information.

Network Status Shows information collected from the TCP/IP drivers at the software level for each network interface card.

Ping Test Shows hosts available for a ping test and an entry box for submitting either a host name or IP address.

Setting-up and Reporting with Spectrum One

The **Spectrum One** menu includes Spectrum One network utilities used for setting and managing terminals and transceivers, and generating reports.



Network Setup Provides a form used for initial setup. Chip seeds, channels, ports, etc., are defined for each Spectrum One RF LAN here.

Terminals Displays a form for terminal addresses in use and those available.

Transceivers Displays a form for the transceiver addresses in use and those available for each LAN.

Activity Report Constructs data traffic and active node reports from available data.

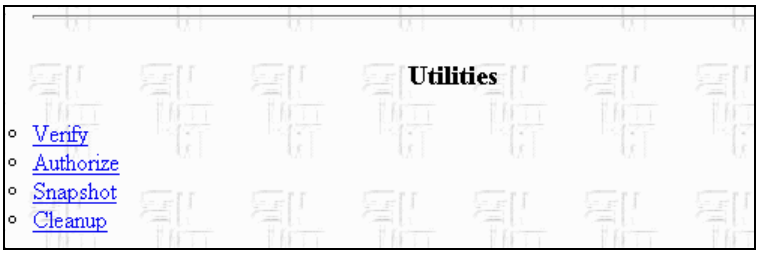
Performance Report (tabular) Creates performance reports for the system, including transaction data, in tabular form.

Performance Report (graphical) Creates graphical performance reports for the system, includes transaction data.

Frequency Scan Report Shows the bandwidth and noise in the 902 - 928 MHz spectrum.

Using Utilities for Checking and Cleanup

OpenAir 400 **Utilities** allow checks of installed software, software authorization, image of current status, and removal of unwanted files. You may use OpenAir 400 utilities at almost any time to verify system software, authorize installed software, capture diagnostic data, prepare files for transfer, and cleanup disk space. The following sections explain the options on the **Utilities** menu.



Verify Checks all installed software. It compares each installed package component against the software release database and reports all correct software, authorized software patches, and any deviations.

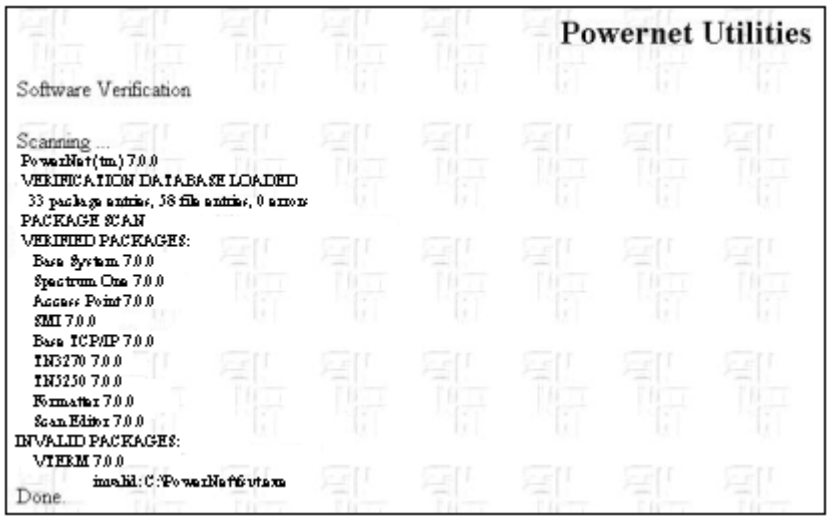
Authorize Checks which software this OpenAir 400 server is authorized to use. It shows which packages are installed and authorized along with the number of terminals authorized for use.

Snapshot Creates a complete image or backup of the current status of the OpenAir 400 server. This option creates a complete archive of all system logs and other relevant information. This utility helps support personnel diagnose possible issues with the environment.

Clean up Removes the system accounting files and clears RF processing log files.

Checking Installed Software

Each system at time of shipment contains the most recent software release database. **Verify** may take up to 30 seconds to analyze the system before displaying information on the screen.



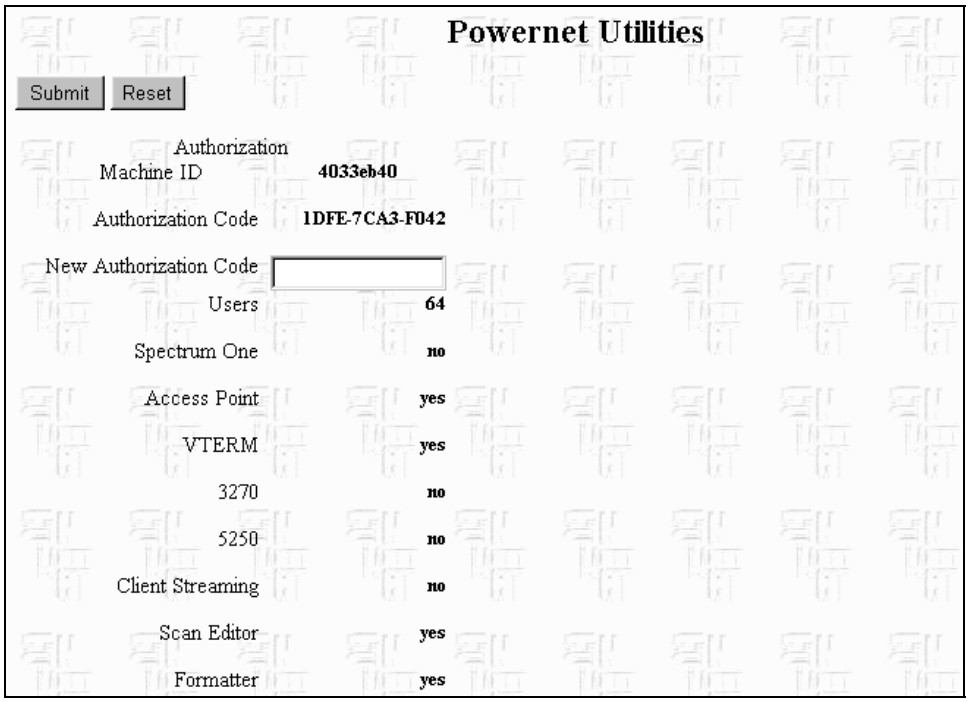
- Select **Verify** from the **Utilities** menu to check all installed OpenAir 400 software. Each installed package component is compared against a software release database that contains a 32-bit CRC of all files. Deviations, including any authorized software patches, are reported.

Entering or Examining Authorizations

Select **Authorize** to enter or update the authorized number of users and installed software packages. This menu option also permits resubmission of the authorization code when adding or removing software and/or hardware from the system.

An unauthorized system operates for one hour, allowing time for completion of the authorization procedure described in the Authorization form supplied with the system. After the one-hour grace period, the system disables OpenAir 400 access until the authorization procedure is completed.

Your authorization code can be obtained from a password-protected page on the Connect, Inc. web site.



1. Turn on the power to the OpenAir 400 server and go to the **Utilities** menu.
2. Select **Authorize** from the **Utilities** menu. The PowerNet Utilities screen appears.
3. Move the cursor to **New Authorization Code**.
4. Enter onto your screen the New Authorization Code **exactly** as shown on our web site (all letters capitalized, and dashes entered). Click on **Submit**.
5. When the code is accepted, the authorization code you entered now appears on the line labeled Authorization Code and the cursor returns to a blank New Authorization Code field.

When these steps are completed, you have successfully entered the New Authorization Code into the system.

If you entered an invalid authorization code, Invalid code appears on the screen. Press the Back button, and the cursor returns to the **New Authorization Code** field. Change the code previously entered. Be sure it matches the code exactly as shown on Connect's web site. Then click on **Submit** to resubmit, and follow the prompts.

Note: If problems continue, call us at (630) 717-7200, and our technical support help desk will provide assistance. They may ask you to *a)* attach a telephone line to the internal modem port labeled **LINE** for us to provide dial-in assistance or *b)* attach the external monitor and keyboard (optional equipment) to enable you to enter diagnostic commands.

Below is the Authorization Fax Form that you may use to authorize your software.

Authorization FAX Form

Please fill out the following information on this form.

Fax this page to Connect, Inc. (630) 717-7243.

Machine ID from the on-screen Authorization menu:

[___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___]

Your Company Name (please print): _____

Location (City, State, Province, Country): _____

_____

_____

_____

Your first and last name (please print): _____

Your fax number with country code / area code: ( ) ___ ___ ___ - ___ ___ ___ ___

Your phone number with country code / area code: ( ) ___ ___ ___ - ___ ___ ___ ___

New authorization code to be supplied by Connect on a return fax.

[___ ___ ___ ___ - ___ ___ ___ ___ - ___ ___ ___ ___]

If you cannot enter the new authorization code into your system, call (630) 717-7200.

Connect, Incorporated, Customer Satisfaction Survey

To provide feedback to our improvement process, please answer the questions below.

Please write down the date the software was downloaded at your facility. _____

Were you able to download successfully? (circle one): Yes No

(If no, please describe problem.) _____

_____

Are you satisfied with the performance of this product? (circle one): Yes No

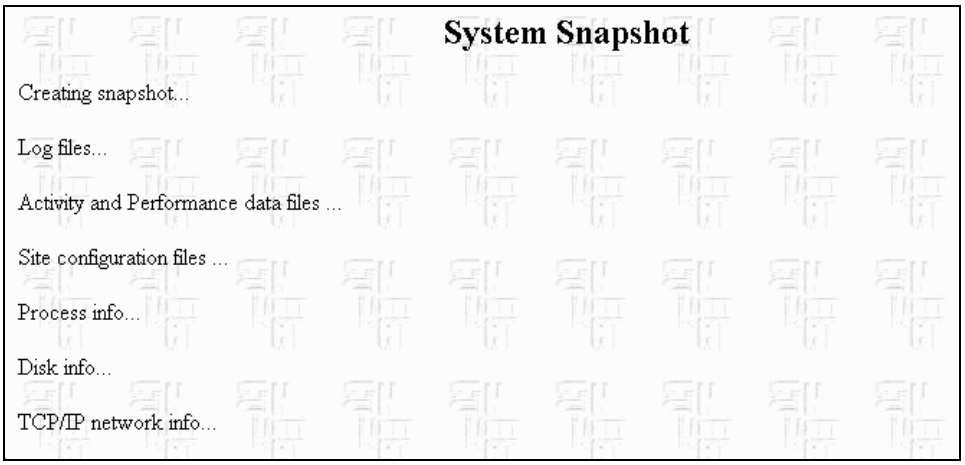
(If no, please describe.) _____

_____

Would you like us to phone you about any problems with this product? (circle one): Yes No

Archiving System Data

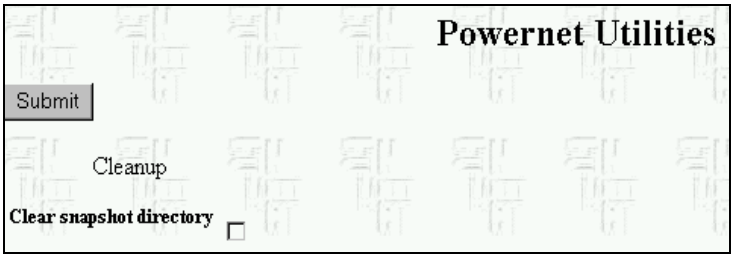
Snapshot copies all logs, system information, disk usage, network status, and configuration to the Snaps subdirectory. **Snapshot** may take 30 to 45 seconds to complete and for all information to appear on the screen.



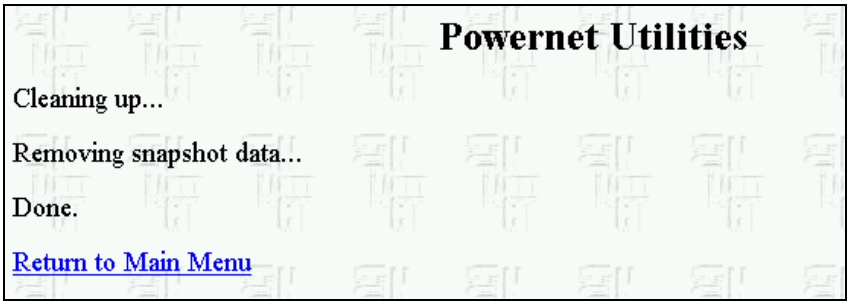
- Select **Snapshot** from the **Utilities** menu to create a complete image or backup of the current status of the system. All snapped files are then saved in the **powernet\snaps** directory.

Removing Unneeded Files

Select **Cleanup** from the **Utilities** menu to remove the OpenAir 400 server activity and performance database files and to clear wireless terminal handler logs.



1. Click to select the box if you want to clear the snapshot directory.
2. Click on **Submit**. The cleanup process completes the processes and displays Done. At this point, you can return to the **Main Menu**.



This page is intentionally blank.

Chapter 4 • Access Point Setup

This chapter describes the software setup procedures for the PowerNet Access Point LAN network. You may find frequent reviews of this section helpful during installation and support of the OpenAir 400 server. Initial configuration requires access to the OpenAir 400 server menu system.

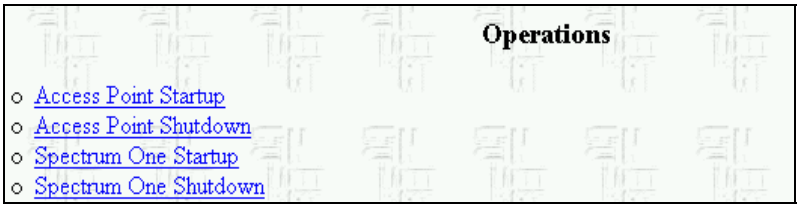
A summary of the setup steps follows:

1. Shut down the Access Point software.
2. Set up the Access Point Network.
3. Restart the Access Point software.
4. Configure and install the new Access Point software.
5. Download the software to RF terminals.
6. Configure terminals and check address assignment.

Shutting Down the Access Point Network

Before configuring an OpenAir 400 server for the first time, shut down the RF processes on the OpenAir 400 server. This shutdown allows configuration parameters to take effect after they have been entered.

1. Select **Operations** from the **Main Menu**. The Operation menu appears.



2. Double-click on **Access Point Shutdown**. A system message asks if you are sure that you want to shut down the network.
3. Click on **OK**, and follow the steps on the screen to shut down the Access Point network.
4. Return to the **Main Menu** by clicking on the browser's **Back** button.

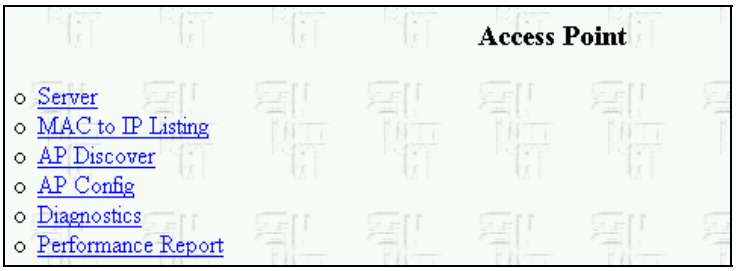
Setting up the Network

Your system administrator must plan the network and make decisions and assignments before work begins. The system administrator:

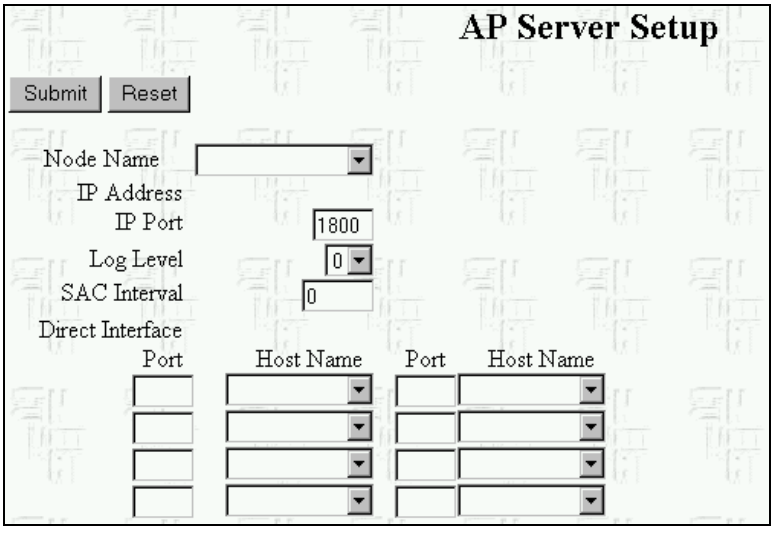
- Decides if a handler is available to the remote terminal devices.
- Assigns a unique IP network address to each Access Point within the RF LAN.
- Assigns the IP Port used as the port address for communicating to the remote devices on the Access Point network.

Setting up the Access Point Server

1. Select **Access Point Menu** from the **Main Menu**. The **Access Point** menu appears.



2. Select **Server**. The **AP Server Setup** screen appears. It consists of one page with form fields for entry or selection.



Node Name This field determines the IP address of the hardware device to which the access point network becomes attached. The values for this field come from the TCP/IP hosts file.
Note: You must establish TCP/IP addresses before selecting Node Names.

IP Address The node-specific IP Address assigned in the TCP/IP setup.

IP Port Your system administrator must enter the IP Port assignment. This assignment is used as the port address for communicating to the remote devices on the Access Point network.

Log Level Logging may provide information at ten different levels. The lowest, zero (0), gives system activity information. It may be set to a higher value, with 9 providing the most information. These level selections may be useful for fine-tuning the system at setup time. Level 1 is the default setting. Set production systems with high transaction rates at zero (0) to minimize the impact of disk logging on system performance.

SAC Interval The SAC interval option controls the RF Network System Accounting facility that generates network load and volume data that can be analyzed with a spreadsheet program. The accounting interval default setting is 150 seconds (2.5 minutes). A setting of zero disables accounting.

The **Direct Interface** fields below allow definition of an IP port number and direct association with a Host list entry. In this manner, a terminal can attach to the PowerNet server and its appropriate host list entry by way of opening a TCP/IP socket connection to the correct port number. This process allows CCP-based terminals to proceed directly to the specific host entry without having to display the entire host list first.

Port The system administrator must enter the Port assignment. The assignment is used as the port address for communication to the Remote devices on the access point network. The value of this entry must match the remote IP port assignment set on the terminals accessing the system by way of this method.

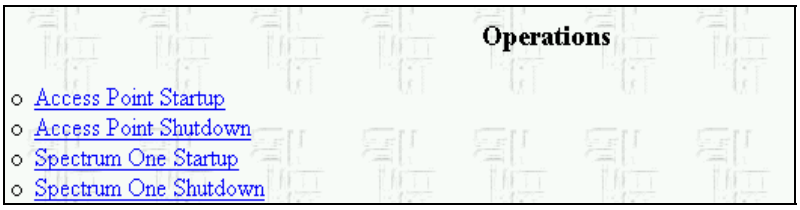
Host Name The Host Name field determines the handler used by the access point terminal to communicate to a host when accessed through the IP port. The values for this field come from the Host List.

1. Select the name of the node from the **Node** list.
2. The IP address appears.
3. Type the correct IP Port number.
4. Choose the correct log level from the **Log Level** field.
5. Type the correct **SAC interval** default setting.
6. Type the remote **Port** assignment number.
7. Select the Host Name from the list.
8. Click on **Submit** to set up this server list.
9. Return to the **Main Menu** by clicking on the browser's **Back** button.

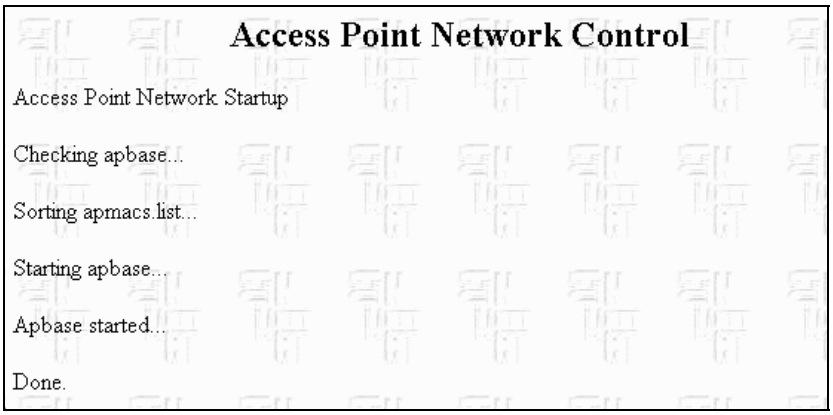
Restarting Access Point

After setting up the Access Point Network, restart Access Point and check your setups.

1. Select **Operations** from the **Main Menu**. The **Operations** menu appears.



2. Double-click on **Access Point Startup**. The **Access Point Network Control** screen appears.

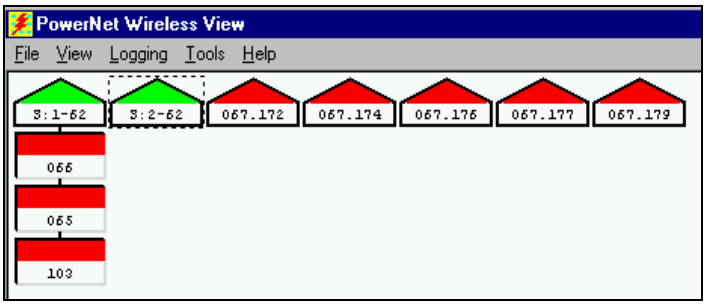


Displaying All Known Access Points

If an Access Point is new, you need to set its IP address at this time. Consult the Access Point vendor documentation for information needed to configure Access Points.

Every Access Point within the RF LAN must have a unique IP network address. Your system administrator assigns this address before installing the Access Points on the network.

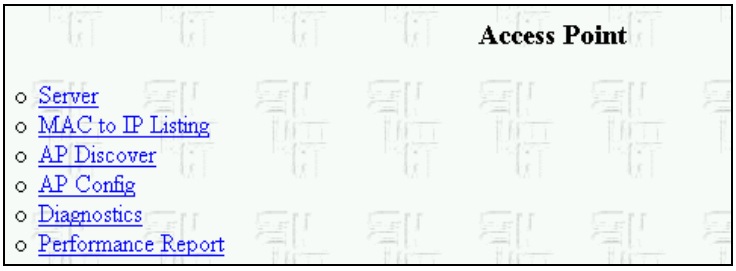
The Wireless View program (discussed in detail later in this document) provides valuable feedback to the installer once this configuration process has taken place. It displays all known transceivers and their status information on the screen.



Updating or Replacing Access Points

Caution: If an Access Point needs to be repaired or replaced, the IP address must be set in accordance to the manufacturer's instructions for the device before it is installed on the network.

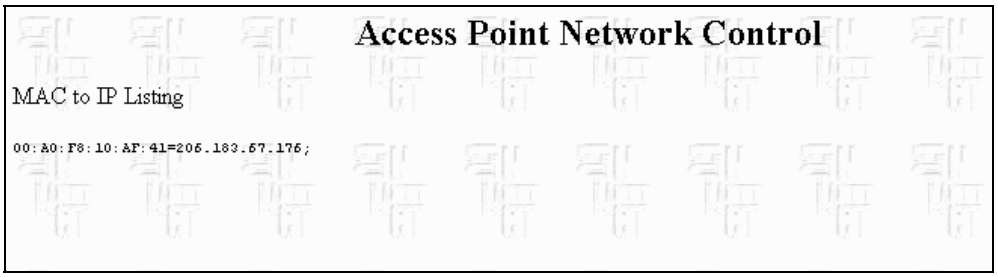
- Select Access Point Menu from the Main Menu. The Access Point menu appears.



Using the MAC to IP Listing

The MAC to IP Listing function allows the system administrator to view the system's known Media Access Control (MAC) addresses, each with its associated IP address. Since some Access Point network devices may have their IP addresses changed or lost (for example, a device sent for repair), you may assign a known MAC address with an IP address that has been assigned to that particular MAC address.

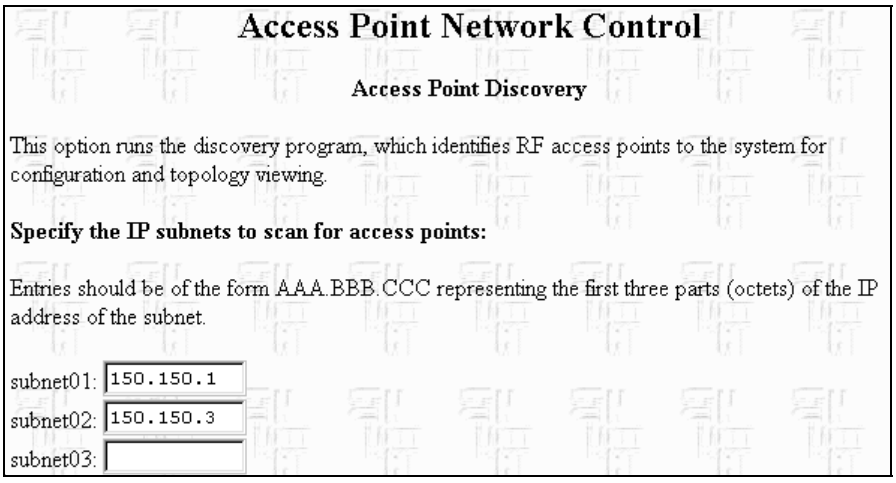
1. Select **MAC to IP Listing** from the **Access Point** menu. A list of known devices on the network appears.
2. Look for the MAC address (usually on a label on the back of the device) in this table and you will find the associated IP address.



Examining IP Addresses

Run **AP Discover** when you first install the PowerNet software, and any time you change (add, remove, or modify) the Access Points (APs), in order to determine the IP addresses and manufacturers of all the APs connected to your network. Several PowerNet functions use this information, including the topology viewer.

1. Select **AP Discover** from the **Access Point** menu. The Access Point Network Control screen appears.



2. Enter in the boxes the three high-order IP octets of each network segment you wish to examine.
3. Click on the **Run Discover** button located at the bottom of the screen (not shown here). AP Discover steps through each possible address on each of the segments identified, looking for any APs. When an AP is found, AP Discover stores its IP address and type code in a file (ipfile.cf) for use by other programs. When AP Discover finishes, the topology viewer then displays information on the APs.

As an example, assume your company uses a class B IP address (for example, 150.150) and the APs are located on the 150.150.1 and 150.150.3 segments. Enter 150.150.1 and 150.150.3 in the boxes, and run Access Point Discovery. It searches from 150.150.1.1 through 150.150.1.254, and 150.150.3.1 through 150.150.3.254. Any APs encountered (for example, 150.150.1.101) are written to the ipfile.cf file.

Configuring AP Parameters

The AP Config function allows system administrators to view and configure Access Point parameters via the AP network backbone.

1. Select **AP Config** from the **Access Point** menu. The **Access Point Selection** screen appears.
2. Enter the IP address of the Access Point to be selected. Once a device is selected, a connection, essentially a Telnet connection, is established with the selected device.

Configuration parameters specific to the device can be accessed via this method. Refer to the device manufacturer's reference manuals regarding these parameters for access point configuration.

Note: Devices may appear on the **Access Point Setup** screen that may have been removed from the active network. If one of these devices is selected, the connection times out after 75 seconds or so.

Chapter 5 • Spectrum One Setup

This chapter describes the software setup procedures for the PowerNet Spectrum One network setup. You may find frequent reviews of this section helpful during installation and support of the OpenAir 400 server. Initial configuration requires access to the OpenAir 400 server menu system.

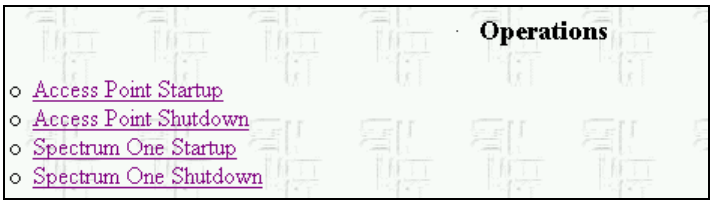
A summary of the setup steps follows:

1. Shut down the Spectrum One software.
2. Complete the Spectrum One Network setup.
3. Restart the Spectrum One software and apply power to the cradle.
4. Apply power to transceivers, one at a time, while checking Wireless View.
5. Download the software to RF terminals.
6. Configure terminals and check address assignment.

Shutting Down the Spectrum One Network

Before configuring a PowerNet server for the first time, shut down the RF processes on the OpenAir 400 server. This shutdown allows configuration parameters to take effect after they have been entered.

1. Select **Operations** from the Main Menu. The Operations Menu appears.



2. Double-click on **Spectrum One Shutdown**. A system message asks if you are sure that you want to shut down the network.
3. Click on **OK**, and follow the steps on the screen to shut down the Spectrum One network.
4. Return to the **Main Menu** by clicking on the browser's **Back** button.

Setting up the Network

Your system administrator must plan the network and make decisions and assignments before work begins. The system administrator:

1. Determines the six channels, chipping sequence (chipseed), speed, and port used for each LAN.
2. Selects the type of port connection.
3. Specifies the device name of the port connection.

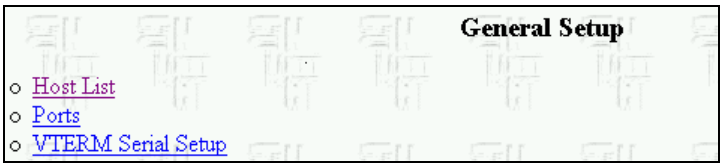
4. Decides if a handler is available to the remote terminal devices.
5. Assigns a unique IP network address to each Spectrum One within the RF LAN.
6. Assigns the IP port used as the port address for communicating to the remote devices on the Spectrum One network.

Setting up Ports

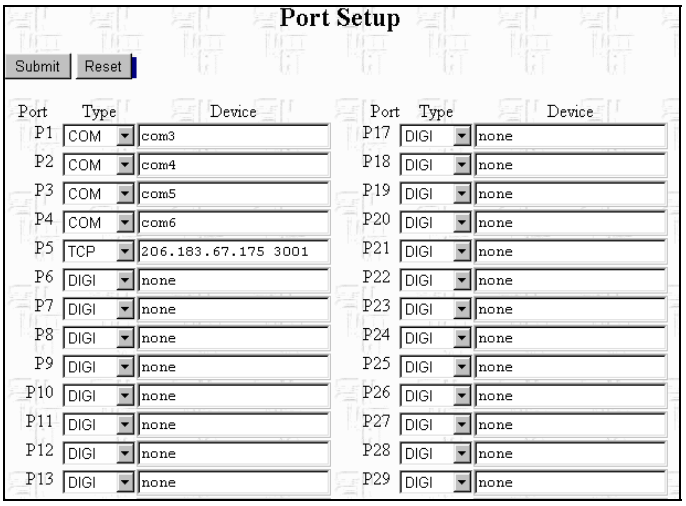
Use the **Port Setup** screen to designate the serial ports available for use by OpenAir 400. The use of multi-port adapters can add 2, 4, 8, 16, or 32 serial ports to an OpenAir 400 server for future, serial-based RF LANs. You can configure these serial ports for transceiver and cradle LANs, all of which are managed by the OpenAir 400 server.

Note: Serial-based RF LANs are not supported with this release of the OpenAir 400 software.

1. Select **General Setup** from the **Main Menu**. The General Setup menu appears.



2. Select **Ports** to use the Port Setup data entry form for selecting network parameters.



Port Lists the port by number.

Type Allows the following selections in this factory-set field:
DIGI for Digi board with multiple connections,
COM for a standard serial port used with your NT/2000 server,
OTHR, CDTA for Central Data terminal server,
TCP for the IP address of Lantronix IP-to-Serial adapter.

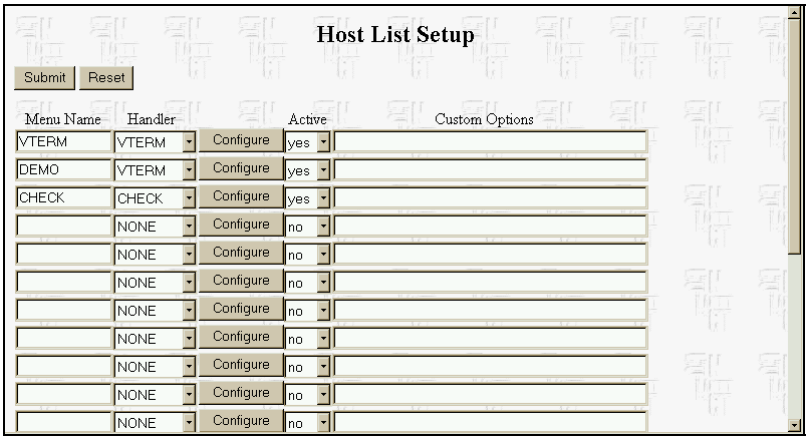
Device Specifies the device name of the port connection. Device names should be obtained from your system administrator.

1. Select the type for each port by using the pull-down menus.
2. Type the device name in the correct space.
3. When the ports to be used have their types and names entered, click on **Submit**.
4. Return to the Main Menu by clicking on the browser's **Back** button.

Setting up Host List

The **Host List** specifies hosts and host applications that are available to RF terminal users. This list appears on RF terminals after they are logged in and configured.

The default contents of the **Host List** depend upon the connectivity option(s) purchased with your system that are explained in the specific reference manuals. In addition, there are *pseudo-host* applications resident on the PowerNet server that you may use for testing.



Menu Name This field allows for entry of up to 12 characters that are displayed on the remote terminal's screen as a host selection.

Handler The handler field specifies the terminal's operating characteristics. Choices depend upon connectivity packages installed. Handler refers not only to the type of connectivity that the application/host works under, but also to the software that process (handle) that application/host in the PowerNet server.

Configure This button brings up the appropriate handler for configuration.

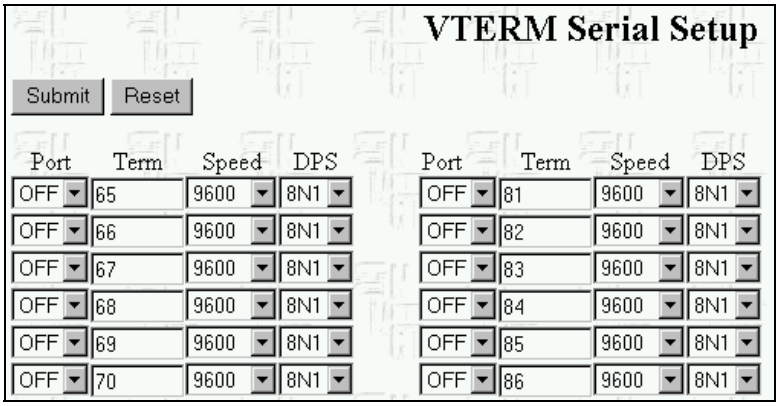
Active This field specifies if a handler is available to the remote terminal devices. When yes, the menu name of the handler is selectable at the remote device. When no, the handler selection is not available to the user.
Active also allows system administrators to set up special 'test' handlers, making them available only when required.

Custom Options This field is used only for options specified by Connect support and should not be required under normal circumstances.

1. Type the name of the program or menu to appear on the remote screen.
2. Choose the **Handler** type and click on **Configure**.
3. Choose **Active** if this menu is available to RF terminal users.
4. Click on **Submit** to set up this host list.

Mapping Terminals to Ports

The General Setup menu provides for the mapping of RF devices to installed serial ports. VTERM Serial Setup, using drop-down windows, allows you to set port number, terminal name, speed, and DPS.



Port Displays the port number.

Term Contains the terminal name/number in this field.

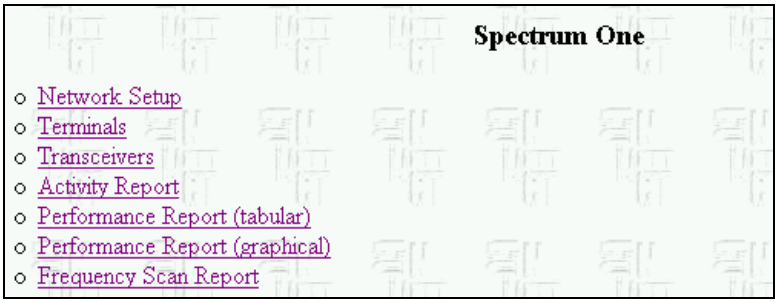
Speed Holds the connection speed.

DPS Sets the data bit, parity, and stop bit.

1. Select the port number from the **Port** list.
2. Type the terminal's name or number in the **Term** field.
3. Choose the correct connection **Speed** for the terminal.
4. Choose the correct bit layout from the **DPS** field.
5. Click on **Submit** to set up this mapping list.
6. Return to the Main Menu by clicking on the browser's **Back** button.

Setting up the Spectrum One Network

1. Select **Spectrum One Menu** from the Main Menu. The Spectrum One menu appears.



2. Select **Network Setup**. The Spectrum One Network Setup screen appears.

Selecting Network Parameters

This screen consists of four pages used to select network parameters. (To view other pages of the network setup form, use the page down button.)

Log Level This sets the level of detail information to be captured in the .log file. Level zero is off, level 1 provides information for most performance analysis, and levels above 5 may affect overall system performance.

Log Lines The file size of all logs is limited with this field. The default size is 5000 lines. When the limit is reached, the log is deleted. A new log is then started. It may be set to a larger value when the Log Level is set above zero to accommodate the additional information that appears in the log. Production systems should be left at the 5000-line setting to minimize disk space usage.

SAC Interval The SAC interval option controls the RF Network System Accounting facility that generates network load and volume data that can be analyzed with a spreadsheet program. The accounting interval default setting is 150 seconds (2.5 minutes). A setting of zero disables accounting.

Total Addresses The number appearing in this field is the number of available network addresses for assignment to terminals.

Channel Lists One channel is assigned to each LAN. If unacceptable levels of interference are detected, the channel can be changed. In addition, this feature field makes it possible to design and implement seamless and segmented RF topologies. Select the first channel using the cursor arrow. Valid entries are 02–50. (Except in Australia, where entries are limited to 26–50.)

Speed The choices for speed in Kbaud are 57.6, 38.4, 19.2, and 9.6. Use either 57.6, 38.4, 19.2, or 9.6 for transceiver LANs and 38.4 for cradle LANs.

Chip Seeds A range of chipping sequences has been predetermined. Any number between 10,000 and 1,000,000,000 may be entered.

Port Port assignments must be the same as those used when assembling the hardware, and are indicated as P1, P2… The selections in this field match the factory-configured ports shown in the Port Setup. OFF should be selected when LANs are not implemented.

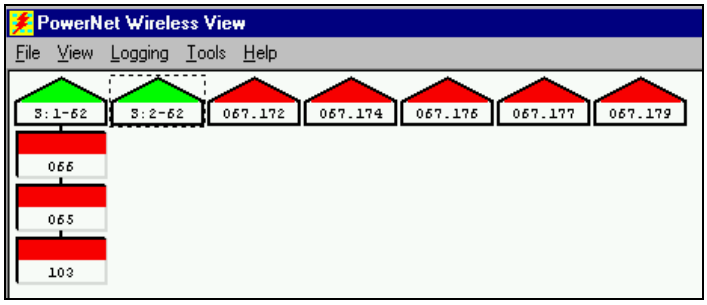
1. Select the level of detail you want to keep in the log from the **Log Level**.
2. Type the number of lines of information allocated for the log in **Log Lines**.
3. Type the correct **SAC interval** default setting.
4. The number appearing in **Total Addresses** is the number of available network addresses for assignment to terminals.
5. Use the cursor arrow keys to select the first channel and assign it to a LAN.
6. Choose a LAN speed.
7. Enter a number for chipping sequences.
8. Assign a port to match those set up on the VTERM Serial Setup window.
9. Click on **Submit** to file this network setup.
10. Return to the Spectrum One screen by clicking on the browser's **Back** button.

Configuring Transceivers

Every transceiver within each RF LAN must have a unique network address. This address is assigned by the PowerNet server when power is first applied to the transceiver.

Note: When setting up the RF LAN for the first time, it is important to power-on each transceiver and let each one configure automatically before continuing on to the next one.

1. From Windows, with your mouse, select **Start**, **Programs**, **PowerNet**, and **Wireless View**. A display of current network topology (access points and transceivers) is presented.
2. Select **Transceivers**. The PowerNet Wireless View screen appears with no transceivers or terminals present. (In the following figure, Spectrum Ones are also displayed, since they can coexist in a PowerNet server environment.)



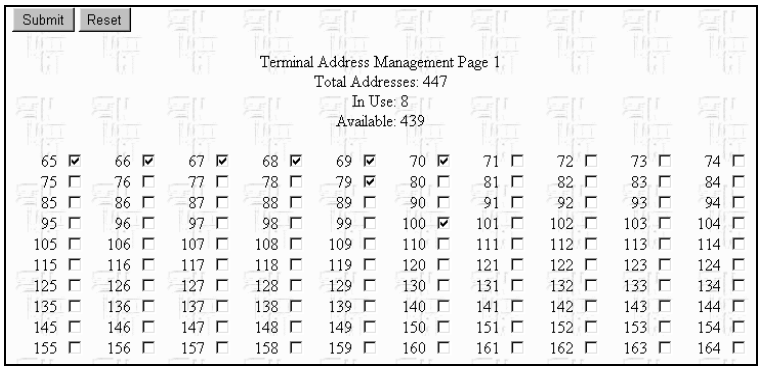
The Wireless View utility provides valuable feedback to the installer while this configuration process is taking place by displaying the transceiver and its status on the screen.

3. Apply power to the first (serial) transceiver of the LAN and it appears on the Wireless View screen.

Assigning RF Terminal Network Addresses

This task provides management of RF terminal network addresses. The addresses begin at 65 and are automatically assigned consecutively when terminals are configured.

1. Select **Terminals** from the Spectrum One menu. The Terminal Address Management screen appears.



- If a terminal address is assigned on the form before configuration, remove the assignment by clicking on the selected box.
- If a terminal is replaced with a new unit, it can be given the original terminal's address, if all numbers up to the reassigned number are in use (or if they are marked with a check just before reassigning the number).

2. Remove the check mark next to the number that is to be assigned.
3. Click on the **Submit** button. The terminal is assigned the first available address, which should be the one that was just cleared.
4. Place the terminal in the cradle for configuration.

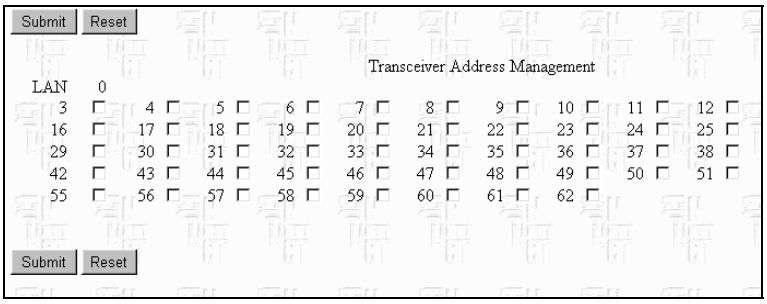
Note: If you do not use the above procedure, the next unused terminal address is automatically assigned to the new terminal.

Eventually, all available addresses are used and additional terminals cannot be added to the system.

Assigning Transceiver Addresses

This task assigns up to 60 transceivers on each of the LANs. Use it after Spectrum One networks are active and transceiver configuration is complete.

1. Select **Transceivers** from the Spectrum One menu. The Transceiver Address Management screen appears.



Note: The default configuration of this form has no assignments. When the first transceiver has power applied to it, an X appears.

- Addresses cannot be assigned manually; only the automatic assignment of the PowerNet server can be used.
- The lowest address (03) is the first address used when transceivers are powered on, and numbers are assigned in order.

Note: If a transceiver address is assigned on this form before configuration, remove the assignment. Click on the X to deselect it.

2. When the form is complete, click on **Submit**.

Deleting and Replacing Transceivers

To delete a transceiver

1. Shut down the Spectrum One system. (If you cannot shut down the active network, the removed transceiver unit continues to appear in the Wireless View Topology screen until the system is shut down and restarted.)
2. Remove the transceiver and display the **Transceiver Address Management** screen.
3. Deselect the X next to the address of the transceiver that was removed. This address is now free for reassignment.

To replace a transceiver

1. Remove the transceiver.
2. Leave the X next to the transceiver's address on the **Transceiver Address Management** screen. Although there is no unit at that position on the backbone, the address is saved.
3. When the transceiver is returned, remove the X.

4. Apply power to the transceiver, and use that same number, if no other lower numbers are free. If they are, they can be temporarily blocked with an X.

Handling Transceiver Address Conflicts

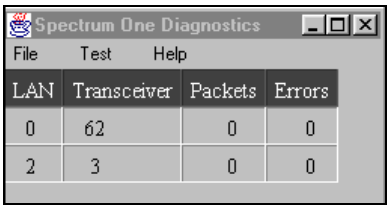
If two transceivers are assigned the same address, or if a transceiver is being moved to another LAN where a transceiver already has that same address, the NVM must be cleared to remove the address from memory.

In the case of conflicts on one LAN, both transceivers involved should be cleared. Consult the manufacturer's guidelines for resetting transceiver addresses.

Using the Diagnostics Facility

This facility provides a means of testing transceivers by sending request packets from the OpenAir 400 server to transceivers. The transceiver(s) then return(s) reply packets with data from various internal registers, giving overall error rates.

1. From Windows, with your mouse, select **Start**, **Programs**, **PowerNet**, and **Spectrum One Diagnostics**.



Downloading RF Applications to the Terminals

The Spectrum One network must be prepared to recognize the RF terminals by downloading software to the terminals, then configuring them with the chipping sequence and channels used to allow them to connect to the RF network. All of this occurs with terminals in cradles.

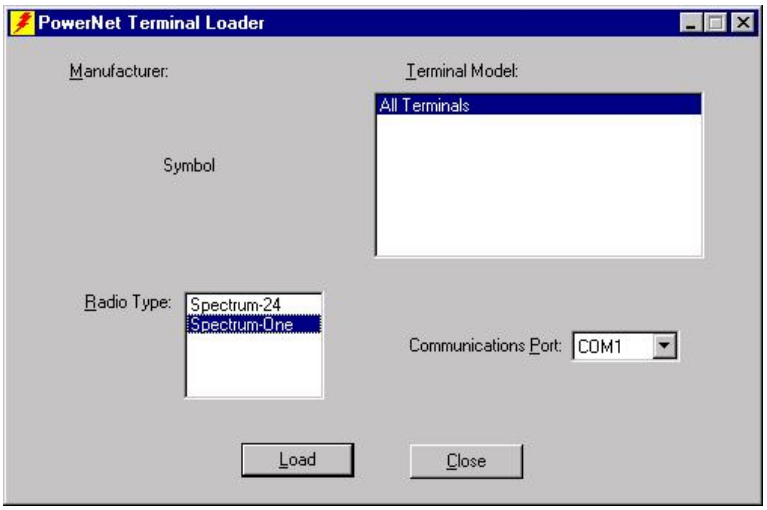
Preparing the Terminals

Note: Vehicle-mounted terminals cannot be inserted in cradles and are connected to the PowerNet server via a null modem cable. See the VRC terminal documentation for more information.

1. Connect the serial load cable from a serial port on the PC to the communications port of the terminal cradle.
2. Connect your PC to your terminal cradle with the correct cables.
3. Check that the terminal is powered off.
4. From Windows, with your mouse, select **Start**, **Programs**, **PowerNet**, and **Terminal Loader** to download the appropriate software to the terminals through the cradle. This software must be loaded on every terminal before assigning addresses.

To select the correct terminal

1. Select **Terminal**, and **Options**. The PowerNet Terminal Loader screen appears.



2. Select from the lists, the correct Manufacturer, Terminal Model, and Radio Type.
3. Select the COM port your PC uses for the communications cable. Click on **OK**.

To download program files

1. Select **Terminal**, and **Send Program Files to Terminal** from the Menu Bar. After selecting the proper terminal options, click on **Load** to begin the transfer. The program asks if you want to include configuration files.
2. Click on **Yes**. Instructions for preparing various terminal models are then displayed; they are a concise version of those shown below**.** If there is more than one **.HEX** file available to download to the terminal, you will be given a list from which to choose. Here *filename* is the name and *nnn* is the version number of the **.HEX** file, which was loaded on the PowerNet server and is listed on the screen.
3. Type ***filenamennn*.HEX** at the prompt, and then click on **Submit**.

To boot in command mode

1. To set up terminals for program downloading, place the terminal in Command Mode using the key combinations shown below. Press and hold down the two letter keys listed, and then press and release the power key, then release the letter keys.

38x0	35-key LRT:	Power + Bksp + Shift + Power
38x0	46-key LRT:	Power + F + I + Power
38x0	35-key PRC:	Power + Bksp + Shift + Power
38x0	56-key PRC:	Power + A + D + Power
38x0	54-key VRC:	Power + A + D + Power
38x0	35-key:	Power + Bksp + Shift + Power
38x0	46-key:	Power + F + I + Power

2. Place the terminal in the cradle.
3. Use the down arrow key on the terminal to select **Program Load**.
4. Press **Enter** on the terminal to erase the EEPROM.
5. Use the down arrow key to select a baud rate of 19200. Press **Enter**.
6. Use the down arrow key to select 8 bits. Press **Enter**.
7. Press **Enter** to select NONE for flow control.
8. Press **Enter** to start program load.
9. Choose **Load** from the PowerNet Terminal Loader Screen and then **OK** to continue with loading the hex file. The screen displays the status of the file load process.
10. Once the **.HEX** file is successfully loaded on the terminal, the terminal screen displays the line: Status: 0000 Download Successful
11. To quit this procedure after it has been completed, press **Del** at the PowerNet server.

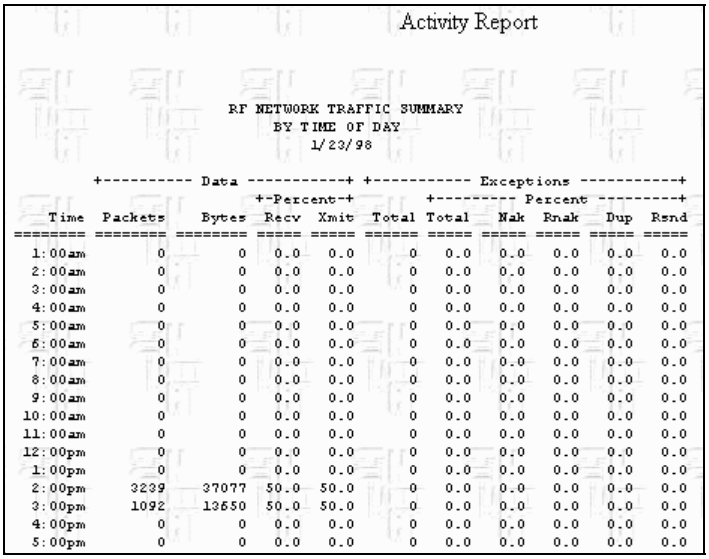
Obtaining and Using Spectrum One Reports

Activity Reporting

The system accounting facility (SAC) generates network load and volume statistics for all nodes on each RF LAN. The data is displayed in activity reports that include traffic summaries with packet, data, and error count totals, LAN and transceiver load distributions, throughput, and error rates.

Displaying Activity Reports

1. Select **Spectrum One** from the **Main Menu**. The Spectrum One menu appears.
2. Select **Activity Report**. The Activity Report screen appears.

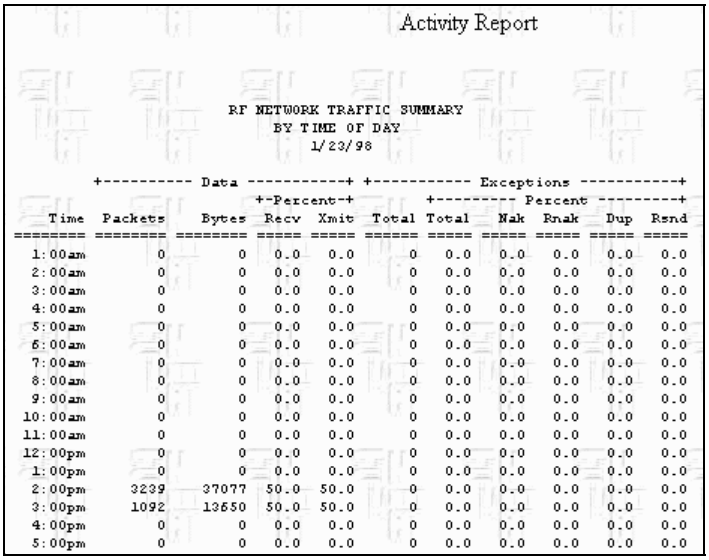


Activity Report

RF NETWORK TRAFFIC SUMMARY
BY TIME OF DAY
1/23/98

	Data				Exceptions					
			Percent			Percent				
Time	Packets	Bytes	Recv	Xmit	Total	Total	Nak	Rnak	Dup	Rsnd
1:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
2:00pm	3239	37077	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0
3:00pm	1092	13650	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0

By Time of Day: Information is shown on an hourly basis and is organized as data packets and exceptions, which are errors (NAKs, RNAKs) duplicate packets, and resends. Columns are totaled at the end of the 24-hour report. These reports give an indication about the total activity at the site.

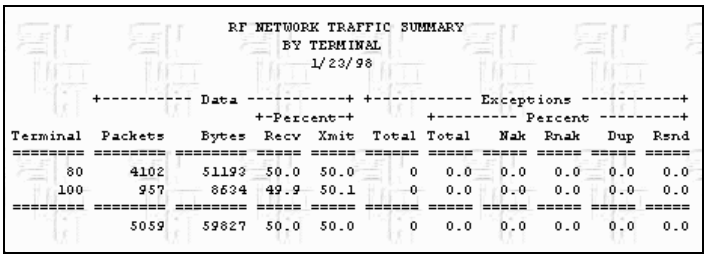


Activity Report

RF NETWORK TRAFFIC SUMMARY
BY TIME OF DAY
1/23/98

Time	Packets	Bytes	Data Percent Recv	Data Percent Xmit	Exceptions Total	Exceptions Percent Total	Percent Nak	Percent Rnak	Percent Dup	Percent Rsnd
1:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
2:00pm	3239	37077	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0
3:00pm	1092	13550	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0

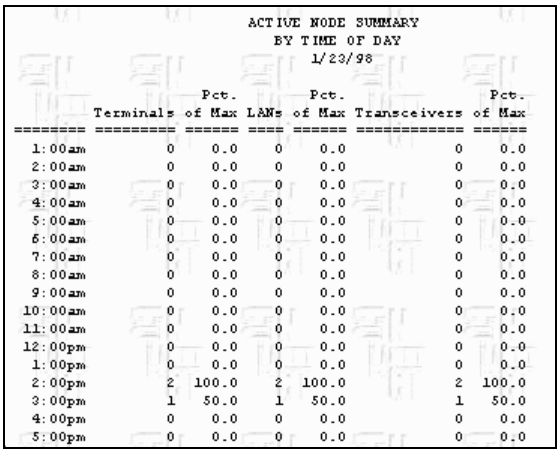
By Terminal: After all of the traffic reports appear, the next set of reports, RF Network Traffic Summary by Terminal, are displayed for the same dates by scrolling down the window. The terminal reports can show trends such as an RF terminal that is operating marginally.



RF NETWORK TRAFFIC SUMMARY
BY TERMINAL
1/23/98

Terminal	Packets	Bytes	Data Percent Recv	Data Percent Xmit	Exceptions Total	Exceptions Percent Total	Percent Nak	Percent Rnak	Percent Dup	Percent Rsnd
80	4102	51193	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0
100	957	8634	49.9	50.1	0	0.0	0.0	0.0	0.0	0.0
	5059	59827	50.0	50.0	0	0.0	0.0	0.0	0.0	0.0

By Time of Day: The third report is the Active Node Summary by time of day. It displays the number of terminals and transceivers active during each one-hour interval. It can indicate trends such as few terminals were powered on at certain hours, or only certain LANs were being used.



ACTIVE NODE SUMMARY
BY TIME OF DAY
1/23/98

	Terminals	Pct. of Max	LANs	Pct. of Max	Transceivers	Pct. of Max
1:00am	0	0.0	0	0.0	0	0.0
2:00am	0	0.0	0	0.0	0	0.0
3:00am	0	0.0	0	0.0	0	0.0
4:00am	0	0.0	0	0.0	0	0.0
5:00am	0	0.0	0	0.0	0	0.0
6:00am	0	0.0	0	0.0	0	0.0
7:00am	0	0.0	0	0.0	0	0.0
8:00am	0	0.0	0	0.0	0	0.0
9:00am	0	0.0	0	0.0	0	0.0
10:00am	0	0.0	0	0.0	0	0.0
11:00am	0	0.0	0	0.0	0	0.0
12:00pm	0	0.0	0	0.0	0	0.0
1:00pm	0	0.0	0	0.0	0	0.0
2:00pm	2	100.0	2	100.0	2	100.0
3:00pm	1	50.0	1	50.0	1	50.0
4:00pm	0	0.0	0	0.0	0	0.0
5:00pm	0	0.0	0	0.0	0	0.0

When all of the reports have been displayed, the PowerNet saves the reports to a file, **sac.rep**.

Using Activity Reports

The system accounting facility (SAC) creates an accounting file, **s1sac.dat**, that does not wrap and grows to a maximum of 2 megabytes. The accounting interval is controlled through the **RF Network Setup** screen and is normally set to 600.

Note: The file can be removed at any time and is recreated at the next interval. By default, the system automatically generates a summary report text file every Monday morning at 1:00 a.m. and then removes the accounting file.

The **s1sac.dat** file, from which the reports are derived, can be exported to a Windows-based program, **RFstats**, for graphical analysis. Alternately, the file can be exported to any user-defined program.

The field definitions for each line entry of **s1sac.dat** are as follows:

	Field	Definition
1.	Sequence	Interval number
2.	Interval	Interval size in seconds
3.	LAN	Network number
4.	Base	Transceiver number
5.	Terminal	Terminal number
6.	LAN load	Total number of nodes attached (***** Year)
7.	Base load	Total number of terminals attached (***** Month)
8.	TX packets	Packets transmitted (***** Day)
9.	RX packets	Packets received (***** Hour)
10.	TX data	Data bytes transmitted (***** Minute)
11.	RX data	Data bytes received (***** Second)
12.	NAK errors	NAKs received from terminal
13.	RNAK errors	NAKs sent to terminal
14.	DUP errors	Duplicates received from terminal
15.	Timeout errors	Timeouts awaiting terminal ACKs

***** Field content when unit (network, transceiver, and terminal) number is zero.

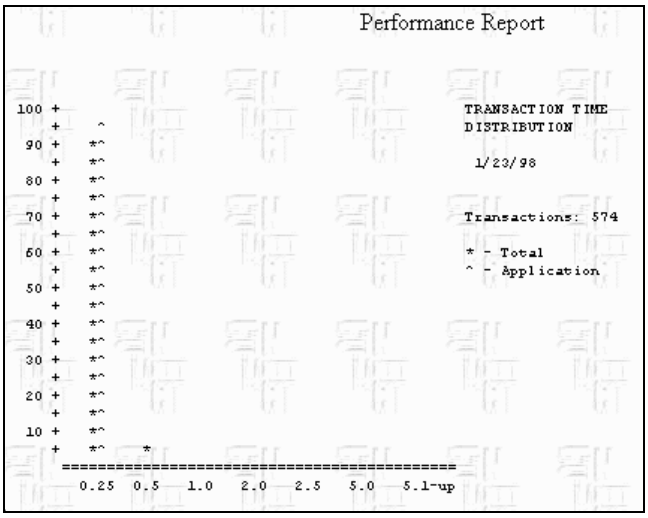
Performance Reporting

Performance Reports help gauge the real-time and application response times of your system. The information can be displayed in graphical and tabular formats.

Displaying Performance Reports

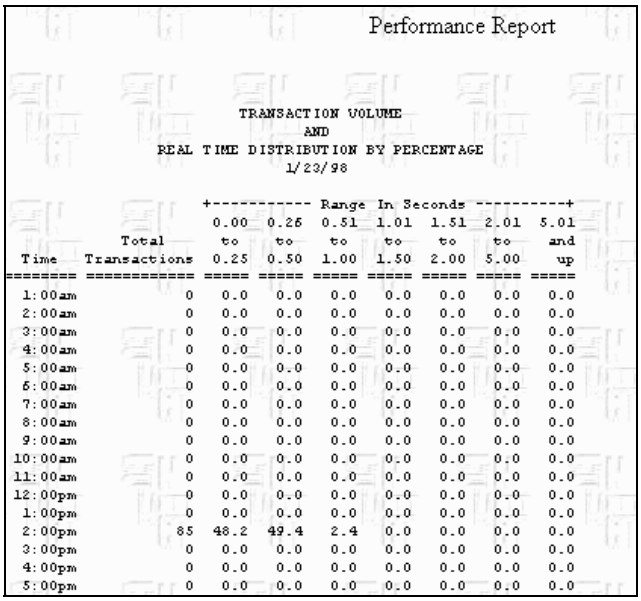
From the Spectrum One menu, select Performance Report (tabular) or Performance Report (graphical), depending upon how you want to view the information. The following illustrations of performance reports show both formats.

By Transaction Time Distribution Information is shown on an hourly basis and is organized as data. Following is a graphical display of transaction time.



The unit for the Y-axis is transactions in percent and, for the X-axis, time in seconds.

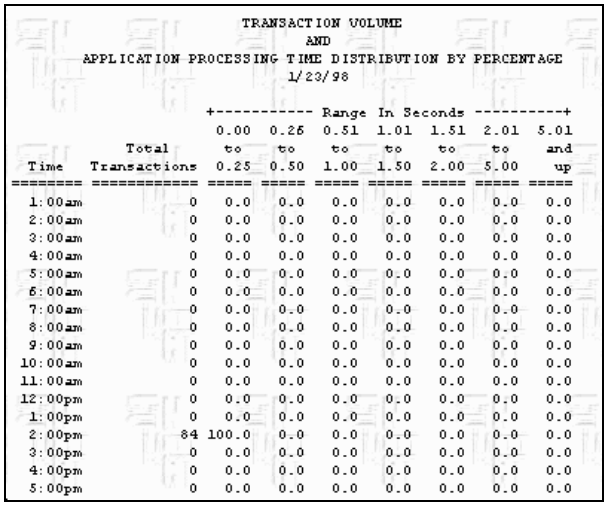
Transaction Time Distribution is shown in tabular format in the next two screen illustrations.



Performance Report

TRANSACTION VOLUME
AND
REAL TIME DISTRIBUTION BY PERCENTAGE
1/23/98

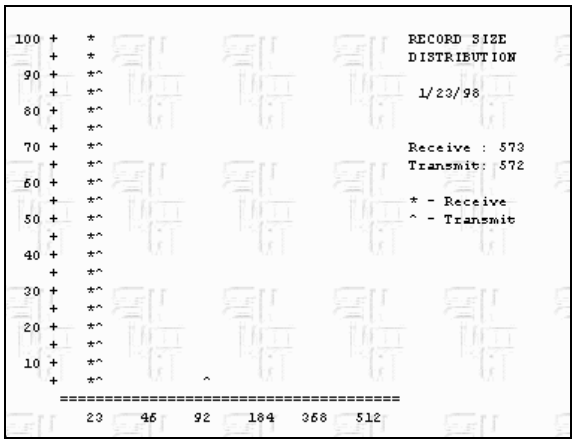
Time	Total Transactions	Range In Seconds 0.00 to 0.25	0.25 to 0.50	0.51 to 1.00	1.01 to 1.50	1.51 to 2.00	2.01 to 5.00	5.01 and up
1:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2:00pm	85	48.2	49.4	2.4	0.0	0.0	0.0	0.0
3:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



TRANSACTION VOLUME
AND
APPLICATION PROCESSING TIME DISTRIBUTION BY PERCENTAGE
1/23/98

Time	Total Transactions	Range In Seconds 0.00 to 0.25	0.25 to 0.50	0.51 to 1.00	1.01 to 1.50	1.51 to 2.00	2.01 to 5.00	5.01 and up
1:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2:00pm	84	100.0	0.0	0.0	0.0	0.0	0.0	0.0
3:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Record Size Distribution information is shown below, organized as a graph, followed by the same data in tabular format.



RECEIVED RECORD SIZE
DISTRIBUTION BY PERCENTAGE
1/23/98

Time	Total	Size Range in Bytes 0-23	24-46	47-92	93-184	185-368	369-512
1:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
2:00pm	84	100.0	0.0	0.0	0.0	0.0	0.0
3:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0

TRANSMITTED RECORD SIZE
DISTRIBUTION BY PERCENTAGE
1/23/98

Time	Total	Size Range in Bytes 0-23	24-46	47-92	93-184	185-368	369-512
1:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
4:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
5:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
6:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
7:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
8:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
9:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
10:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
11:00am	0	0.0	0.0	0.0	0.0	0.0	0.0
12:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
1:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
2:00pm	84	51.2	0.0	47.6	1.2	0.0	0.0
3:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
4:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0
5:00pm	0	0.0	0.0	0.0	0.0	0.0	0.0

When all of the reports have been displayed, PowerNet prompts to save the reports to a file, **prf.rep**.

Using Performance Reports

The **s1prf.dat** file, from which the reports are derived, can be exported to a Windows-based program, **RFstats**, for graphical analysis. (Please see your sales representative for more information.) Alternately, the file can be exported to any user-defined program.

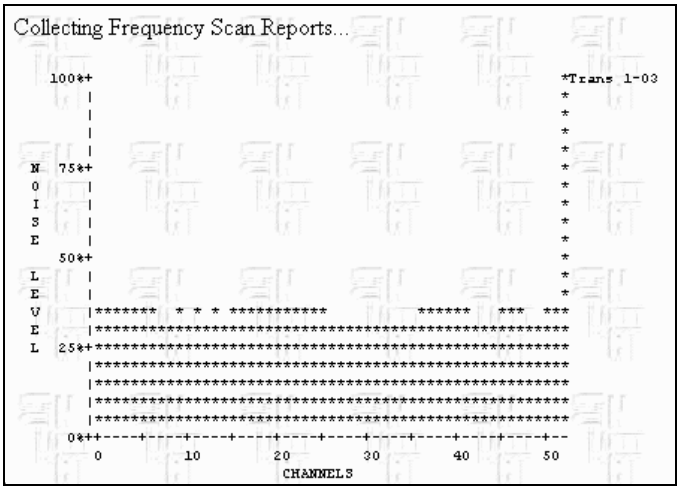
The field definitions for each line entry of **s1prf.dat** are as follows:

Field	Description/Value
1	Sequence Number
2	Elapsed Time in Seconds
3	Unit Number
4	End-End Transaction Times 0ms to 250ms (***** Year)
5	End-End Transaction Times 251ms to 500ms (***** Month)
6	End-End Transaction Times 501ms to 1000ms (***** Day)
7	End-End Transaction Times 1001ms to 1500ms (***** Hour)
8	End-End Transaction Times 1501ms to 2000ms (***** Minute)
9	End-End Transaction Times 2001ms to 5000ms (***** Second)
10	End-End Transaction Times over 5001ms
11	Application Transaction Times 0ms to 250ms
12	Application Transaction Times 251ms to 500ms
13	Application Transaction Times 501ms to 1000ms
14	Application Transaction Times 1001ms to 1500ms
15	Application Transaction Times 1501ms to 2000ms
16	Application Transaction Times 2001ms to 5000ms
17	Application Transaction Times over 5001ms
18	Receive Records Size 0–23 bytes
19	Receive Records Size 24–46 bytes
20	Receive Records Size 47–92 bytes
21	Receive Records Size 93–184 bytes
22	Receive Records Size 185–368 bytes
23	Receive Records Size 369–512 bytes
24	Transmit Records Size 0–23 bytes
25	Transmit Records Size 24–46 bytes
26	Transmit Records Size 47–92 bytes
27	Transmit Records Size 93–184 bytes
28	Transmit Records Size 185–368 bytes
29	Transmit Records Size 369–512 bytes

***** Field content when Unit Number (field 3) is zero.

Frequency Scan Reporting

From the **Spectrum One** menu, select **Frequency Scan Report** to generate Frequency Scan Reports (FSRs). The RF environment surrounding each transceiver is measured and displayed for all Spread Spectrum channels.



The X-axis of this report displays the Spread Spectrum channels 0–52. The Y-axis of the FSR is a relative measure of the noise level (RF activity) as a percent. Although it is not a fixed quantity, it is suggested that a level of 50% or higher indicates that a channel should not be used. If a channel in use has a level of 50% or greater, the channel should be switched to one that is quieter (i.e., with less radio activity).

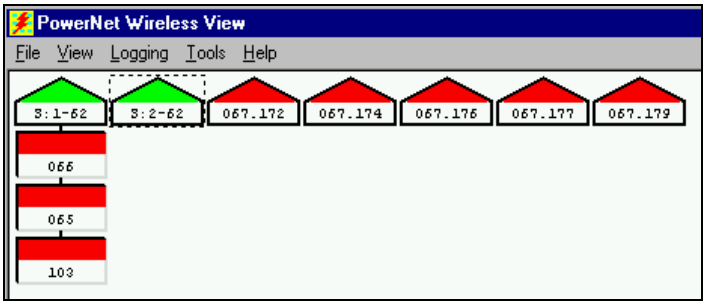
Poor data transmission may result from interference on the channel in use on a particular RF LAN. Examining FSRs allows you to identify potential problem areas. Since interference may only occur in the vicinity of one or two transceivers, all transceivers should be examined when a problem occurs.

Note: Certain combinations of transceiver firmware use a different FSR reporting convention than the one described in this manual. In this case, the channels are normally at a 0% noise level instead of the nominal 25% noise level shown in the illustration.

Chapter 6 • Administration and Maintenance

Using Wireless View

1. From Windows, with your mouse, select **Start**, **Programs**, **PowerNet**, and **Wireless View**. A display of current network topology (access points and transceivers) is presented.



2. Several operator options are available on the PowerNet Wireless View screen as pull down menu items. The following list shows the options available from each pull down menu. The options are explained later in this section.

File Connect – A window appears for you to enter the name of the computer running PowerNet.

Exit – Closes Wireless View.

View Refresh – Updates the topology display.

Logging Access Point Log Level – Sets the level of information detail captured by the apbase.log file.

Spectrum One Log Level – Sets the level of information detail captured by the base.log file.

Terminal Log Level – Sets the level of information detail that is captured in the individual log file.

Tools Statistics – Displays statistics for the selected device.

Network Summary – The total number of all transceivers, terminals (active and inactive), and LANs in the installation is reported.

Tools (cont.)

Locations – Allows the operator to establish a named location for a given Access Point or Transceiver.

Reload Host List – Resets the RF network without terminating current RF sessions.

Clear NVM – Used to clear the selected transceiver's Non-Volatile Memory (Spectrum One only).

Reset LAN – Used to remove the selected transceiver's address assignment (Spectrum One).

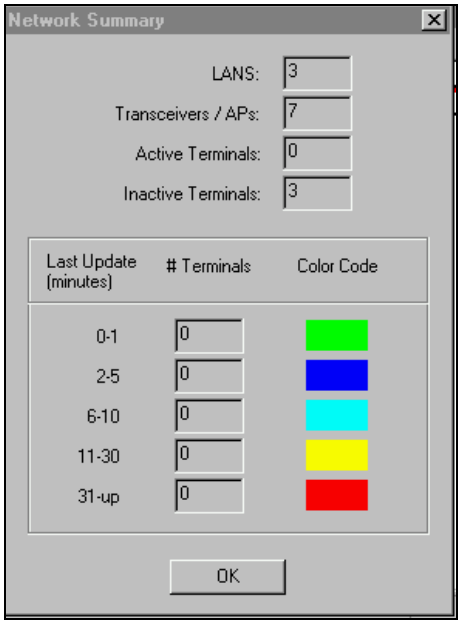
Ping Test – Allows pinging Access Points to confirm that the device is accessible by the PowerNet NT/2000 system.

Refreshing the Screen

This topology view frame normally refreshes every 10 seconds. By selecting **Refresh** from the **View** menu, the operator can manually update the topology display immediately.

Viewing the RF Network Summary

1. Choose **Network Summary** from the **Tools** menu. The Network Summary screen appears. The Network selection displays a summary screen of all RF networks as shown below.



- The total number of all transceivers and terminals (active and inactive) is reported.
- The terminal traffic information is shown.

Describing Access Point and Transceiver Locations

System administrators or installers can enter a written description of the location of a given access point or transceiver. Once a location for a device is set, the location name for that device appears in the **Statistics** screen when that device is selected. Locations cannot be assigned to terminals.

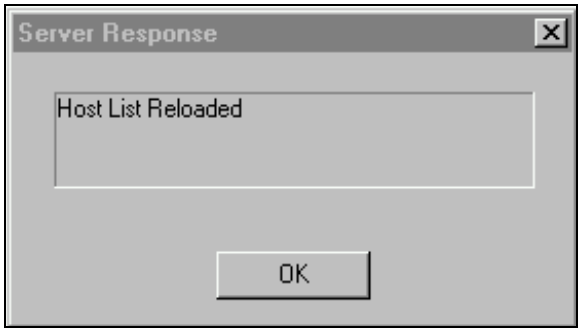
1. Choose **Locations** from the **Tools** menu. The **Location** screen appears.

2. Type a descriptive name for the location of a particular access point or transceiver. Click on **OK**.

Reloading and Resetting the RF Network

This option resets the RF network without terminating current RF sessions. New parameters that have been set, such as log level settings at the handler level, are changed by reading the modified handler file when this option is implemented.

1. Choose **Reload Host List** from the **Tools** menu. The Server Response screen appears.



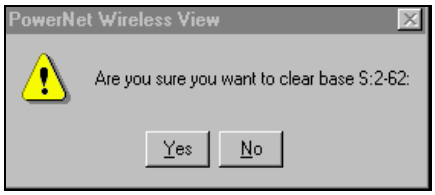
2. Click on **OK**. The host list is reloaded.

Clearing a Transceiver's NVM (Memory)

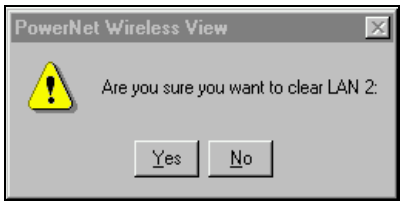
For situations where there are address conflicts, clear the NVM of both transceivers, power both off, clear the address, and then configure one at a time.

1. Select the transceiver to reset by clicking on it from the PowerNet Wireless View screen.

2. Select **Clear NVM** from the **Tools** menu.



3. Click on **Yes** on the verification screen.
4. Manually turn the power off to the transceiver.
5. Select **Reset LAN** from the **Control Menu**.



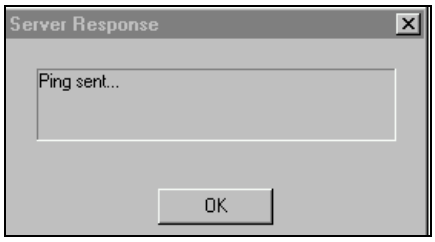
6. Remove the transceiver address assignment from the Transceiver Address Form (delete the X).
7. Manually turn the power on to the transceiver. It is configured using the lowest available address.
8. Ensure that configuration is complete.
9. Check the Wireless View topology to ensure that the process has proceeded successfully.

Using a Ping Test to Check Connections

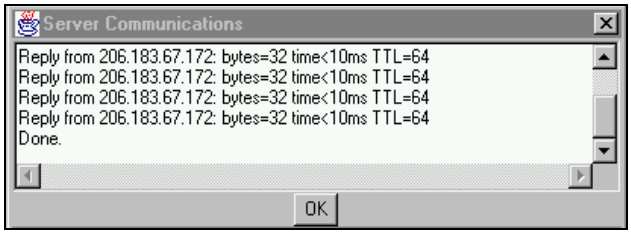
The PowerNet server generates a query packet that is sent to the device chosen. The service then sends a reply packet to the PowerNet server. Should the test fail, be certain that all connectors are properly mated and that there are no breaks in the coaxial cable. If problems with the cabling or connectors are not detected, then further investigation is required.

1. Select a transceiver from the PowerNet Wireless View screen.
2. Select **Ping Test** from the **Tools** menu.
3. Click on **Submit** to begin the test. The server sends the transceiver a single query packet. In pinging an access point, a TCP/IP ping test is used.
4. A Server Response screen appears with the message **Ping sent…**

In pinging a LAN-Trans or Access Point, the result, whether the test was successful or not, is recorded in the lower corner of the screen.



5. Click on **OK**. Results of the ping test are displayed on the screen as the test is being done.



The test stops after five pings.

6. Click on the browser's **Back** button to return to the **Control Menu** and continue.

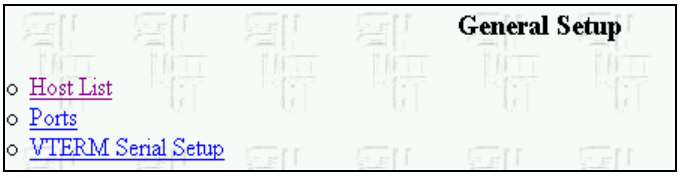
Listing RF Terminal Applications

The Host List specifies hosts and host applications that are available to RF terminal users. This list appears on RF terminals once they are logged in and configured. The Host List is constructed and changed with the Host List Setup Form.

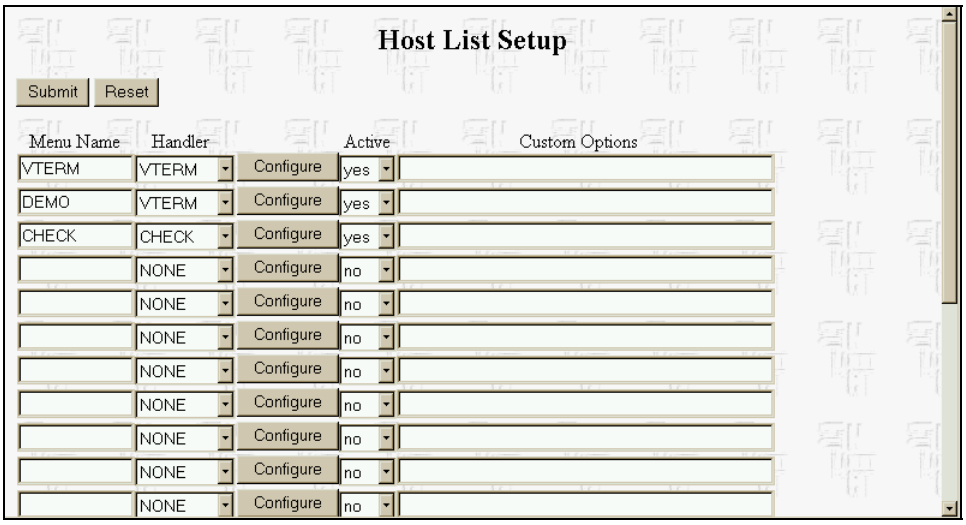
The default contents of the Host List correlate with the connectivity option(s) that is part of each system. These contents are explained in their specific reference manuals. In addition, there are *pseudo-host* applications resident on the PowerNet server that is used for testing.

Creating a New Option on the Host List

1. Select **General Setup** from the **Main Menu**. The **General Setup** menu appears.



2. Choose **Host List** and the Host List Setup form appears.



3. Place the cursor on a blank line in the **Menu Name** column.
4. Enter the name of the selection you want to appear on the RF terminal screen (using upper and lower case where needed).
5. Click on the **Handler** field's down arrow and use the mouse to change the selection. The available choices depend upon the connectivity packages that are installed. The following are currently available: None, TN3270, TN5250, VTerm. The handler refers not only to the type of connectivity that the application/ host works under, but also to the software that processes (handles) that application/host in the PowerNet server.
6. Click on the **Configure** button. This action causes the Host List function to bring up the appropriate handler for configuration.

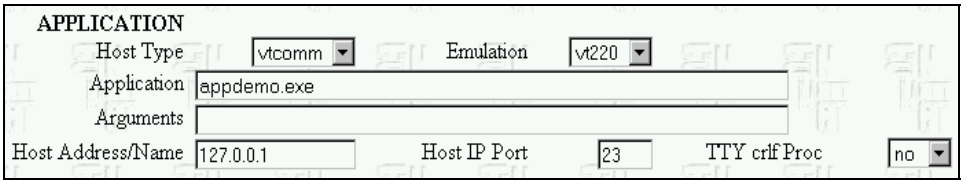
Configuring a Handler

The VTerm handler, accessed through the **Host List** as discussed above, allows application developers to send specific configuration options to the terminal program that resides on the remote device. The device configuration options free the developer from having to set specific configuration options from the application itself. This feature allows the developer to fine-tune the application's operation from within the handler without the need to rebuild the application to suit user requirements.

Clicking on the **Configure** button brings up the **Host List Setup** for the selected configuration. The examples that follow are from a typical VTerm setup.

Note: Many specific terminal options are dependent on the terminal manufacturer's design.

Applications Options



Host Type Establishes session connection to host.
Serial is used for connecting to serial vt100 communications devices.
Ptty allows a Windows NT/2000 console application to be run with the screens read and sent to the terminal.
Telnet is for communicating directly to a vt100/vt220/hp700 host via the telnet protocol.
Vtcomm is for communicating with the rfcurses DLL or ActiveX control. (For DLL details, see Chapter 9, Windows NT/2000 DLL Interface.)

Application This is the application program to be invoked by the handler at session startup if the Host Type is ptty or vtcomm. The default value is c:\winnt\system32\cmd.exe.

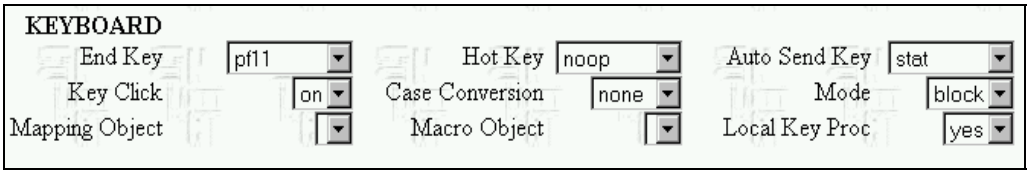
Arguments States the arguments to the application. The default value is blank.

Emulation Sets the emulation mode either to vt100, vt220, or hp700. The default is vt100.

Host Address Allows the operator to enter the IP address of the target host machine, the machine running the DLL in vtcomm mode, or the address of the telnet host in telnet mode.

TTY crlf proc Enables processing of incoming crlf characters. Values for field are yes or no. The default is no. (For ptty hosts only: This connects carriage returns to a carriage return/linefeed pair when used as input to the host application.)

Keyboard Options



End Key Terminates both the host connection and the RF session when transmitted by the terminal. The key choices vary according to the handler type. The default value is pf10 for all handlers.

Hot Key Allows the user to hotkey between sessions. The key choices vary according to handler type. The default value is noop, which disables Hot Key processing.

Note: Each hotkey session requires a terminal license.

Auto Send Key Generated automatically when an input field on the terminal is filled with keyed input (as opposed to scanned input). The default value is noop, which disables key generation. If the Length Check option in the Scanner Options Group is set to on, this function is disabled.

Key Click Enables/disables key clicks on the terminal. The default value is on.

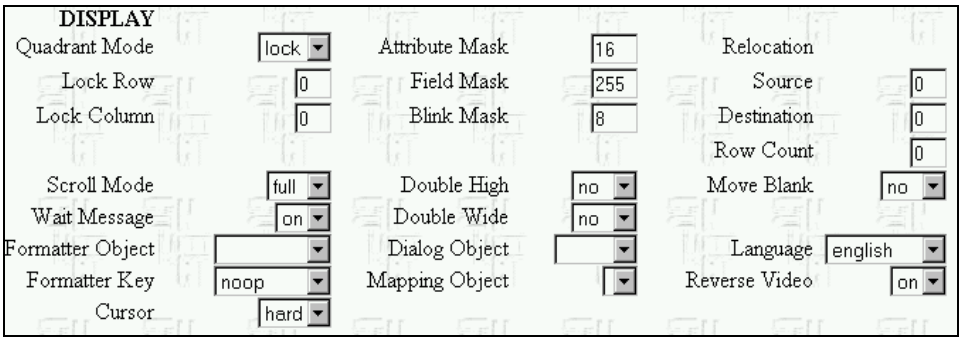
Case Conversion Controls conversion of keyboard input to upper or lower case. Applications may be case-sensitive or case-insensitive. Select the required case, upper or lower, and all alphabetic characters sent from the terminal are in that case. Select none, if the application is case-insensitive. The default value is none, meaning no conversion takes place.

Mapping Object Defines an object used to define keyboard mappings. Refer to the Keyboard Mapping section in the appropriate terminal user's guide for more information on the content and format of this file. The field values for this field are created with files using the Keyboard mapping object editor.

Macro Object Defines an object used to define keyboard macro settings. The field values for this field are created with files using the Keyboard macro object editor.

Local Key Proc Specifies whether all local terminal editing keys, such as backspace, are handled locally on the terminal. The default value is yes. The value should be set to no if Mode is set to char.

Display / Relocation Options



Quadrant Mode Controls automatic quadrant processing by the server handler.
Off disables quadrant processing; the handler attempts to center the current host input field in the terminal display.
On enables quadrant processing. However, input fields that cross quadrant boundaries result in a shift to the left or right. The default value is on.
Soft always positions on a quadrant boundary regardless of input field boundaries. Viewing keys are enabled.
Hard is the same as soft except the viewing keys are disabled.
Lock locks the terminal display to host display row and column coordinates (see Lock Row and Lock Column below). The default value is on.

Lock Row Defines the Y coordinate for display position locking (see Quadrant Mode). Range is zero (0) to maximum number of rows on the host screen. The default value is zero (0). This field is in effect only if Quadrants is set to lock.

Lock Column Defines X coordinate for display position locking (see Quadrant Mode). Range is zero (0) to the maximum number of columns on the host screen. The default value is zero (0).

Scroll Mode Defines scrolling method used when using the viewing keys. A value of half means that the terminal display is moved in half terminal screen increments. A value of full means that the terminal display is moved in full screen increments. When full is selected for an 8 by 20 screen, the screen scrolls up/down 8 rows. If half is selected, it scrolls up/down 4 rows. For left/right, full scrolls 20 columns and half scrolls 10. The default value is full.

Wait Message Enables/disables the display of the "waiting for data" message on the terminal. The default value is on.

Formatter Object Defines the object created with the Screen Formatting utility. The default value is blank. Field values are limited to the available formatter objects created using the screen formatter utility.

Formatter Key This alternate key field is used with the Screen Formatter. When using that package, a key function, other than the Standard Key, can be sent to the host after a bar code is scanned. Therefore, a choice of two keys, which can be used selectively, exists with the Formatter. Use noop when an Alternate Key is not used. The default value is noop.

Cursor Defines how the cursor is displayed on the terminal. The default is hard, which displays a blinking block cursor. A value of hide eliminates the cursor, and soft results in a software generated cursor that displays the current keyboard state (shifted, control pressed, etc.).

Attribute Mask Specifies a decimal value used to mask character attributes that are converted to reverse video. The default is 255. The mask indicates to the VTerm software (in VT100/220 mode) which video attributes should be used to make up an input field. The flag should be set to the decimal equivalent of the appropriate bit pattern. The decimal equivalents for each video attribute are as follows: Bold=1 Dim=2 Underline=4 Blink=8 Reverse=16. For example, to indicate that only underlined or dim fields are to be treated as input fields, set the attribute mask value to 6 (2 (dim) + 4(underline)). This field defaults to 255 indicating that any video attribute encountered is displayed as reverse video.

Field Mask Specifies a decimal value used to mask character attributes for the purpose to defining fields so that block mode operation (See Host Interface Options Group) can be implemented. The default is 255.

Blink Mask Specifies a decimal value used to mask character attributes that result in blinking on the terminal display. The default is zero (0).

Double High Enables/disables double high display of characters on the terminal. The default is no.

Double Wide Enables/disables double wide display of characters on the terminal. The default is no.

Dialog Object Defines the dialog object. The default value is blank for the VTerm handler. Values for this field are created through the dialog object editor.

Mapping Object Display character mapping object file that is created using the display mapping object editor.

Relocation Source Defines the starting row (zero-based) in the host display that is relocated to the Destination row of the host display. The default value is zero (0).

Destination Defines the starting row (zero-based) in the host display where the starting Source row is located. The default is zero (0).

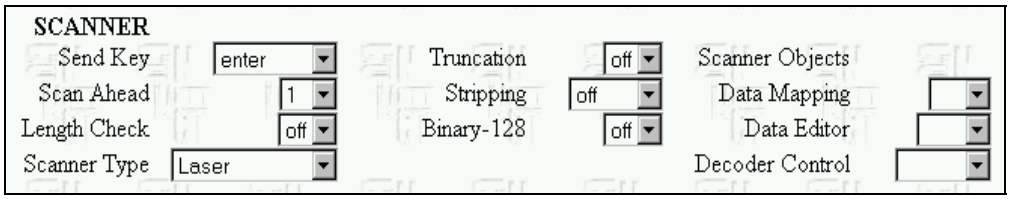
Row Count Specifies the number of host display rows to be relocated, starting with the row defined by the Source option and continuing from top to bottom of the display. A value of zero (0), the default, disables row relocation.

Move Blank Enables the relocation of a row from the host display even if it is blank. (Blank is defined as null or space characters.)

Language Defines the character set used for terminal error message displays. The default value is English. Available selections for this field are: English, French, German, Spanish, and Italian.

Reverse Video Enables/disables displaying the reverse video attribute on the terminal. The default is on.

Scanner / Scanner Objects Options



Send Key Selects the key function that is sent to the host after a barcode is scanned. The key is not pressed, but is sent automatically after the scanned data by CCP (Common Client Program). The default value is enter. This function is disabled by setting the value to noop.

Scan Ahead Sets the number of scans that is buffered in the terminal. Values for this field range from 0 to 24. The default is 1.
Set to 0, scan ahead is disabled and you cannot scan again until the host response has been received. Set to 1, there is no scan ahead limit. Set to 2 and above, and the software will let you scan that many times until a host response is received. Therefore, if you set it for 3, it will allow you to scan only 2 more times before the host responds to the first scan.

Length Check Enables or disables local terminal field length checking. A value of on causes an error message to be generated locally by the terminal if scanned input is greater than the length of the field or if an attempt to enter a non-control key (enter, tab, arrows, function keys, etc.) is made after the field is filled. ”On” also disables Auto Send key operation (See Keyboard Options Group) and the Truncation option (see below). The default value is off.

Scanner Type Informs CCP running on the remote terminal which scanner type to use for input. Values for this field are Laser, Contact/P, Contact/NP, Auto/P, Auto/NP, or WandSim. Contact scanner manufacturers for details about these values and for which type of scanner to use if other than laser.

Truncation Enables/disables the truncating of scanned inputs to the length of the input field. This option is not applicable if the Length Check option is set to on. A value of on also disables field wrapping. The default value is off.

Stripping Strips trailing spaces and/or underscore characters from scanned input. The default is off. The options are off, space, score, and both.

Binary-128 Enables/disables the processing of binary code 128 bar codes on the terminal.

Data Mapping Defines the object that describes how scanned input is mapped. The field values for this field are created with files using the scanner data mapping object editor.

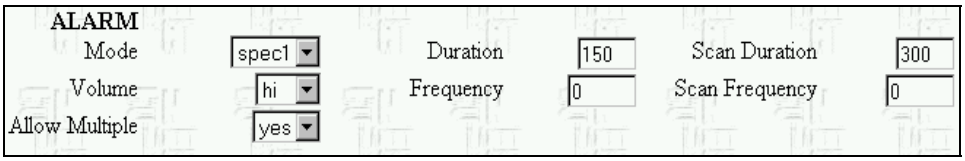
Data Editor Defines the object that describes how scanned input is edited prior to delivery to the host. The Scan Editor utility program creates the object.

Decoder Control Defines the object that describes how the symbology decoder on the scanner functions. The field values for this field are created with files using the scanner decoder control object editor.

Below is a sample scanner data mapping object.

```
#
# Sample scanner data mapping object
#
001253=pf4;
\011=tab;
\015=enter;
ENT=enter;
TB=tab;
```

Alarm Options



Mode Defines the operation of the audible alarm and scanner light on the terminal. The default value is bell.
Off disables the audible alarm.
Bell Enables the audible alarm as a double beep.
Flash Enables the scan indicator light as the alarm with no audible alarm.
Both Enables the double beep and the scan indicator light as the alarm.
Spec1 Enables the audible alarm as a single beep.
Spec2 Enables the single beep and the scan indicator light as the alarm.

Volume Defines the volume of the audible alarm on the terminal. The default value is hi.

Allow Multiple Enables/disables the processing of multiple alarm commands generated by the host application in a single terminal update sequence. If this field is set to yes and the host sends down multiple bells on a screen, the alarm on the terminal beeps as set for each of the bell commands. If this field is set to no, the bells are combined and the terminal only gives one alarm. The default value is yes.

Duration Defines the duration of the audible alarm on the terminal in milliseconds. The alarm mode must be set to spec1 or spec2 for duration to take effect. The default value is 150.

Frequency Defines the frequency of the audible alarm on the terminal in Hertz. The alarm mode must be set to spec1 or spec2 for frequency to take effect. The default value zero (0) represents the factory set default frequency considered optimum for the particular model of terminal.

Scan Duration Defines the duration of the audible alarm generated by a scan operation on the terminal, in milliseconds. The alarm mode must be set to spec1 or spec2 for scan duration to take effect. The default value is 300.

Scan Frequency Defines the frequency of the audible alarm generated by a scan operation on the terminal, in Hertz. The alarm mode must be set to spec1 or spec2 for scan duration to take effect. The default value zero (0) represents the factory set default frequency considered optimum for the particular model of terminal. This parameter, combined with the scan duration parameter, allows system administrators to configure alarm sounds differently for users to differentiate between alarms and scan decode tones.

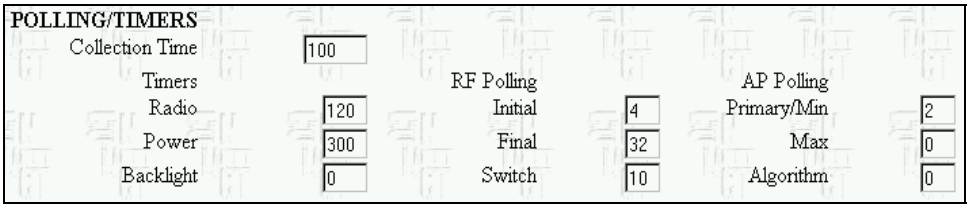
Printer Options

Type Defines the type of printer attached to the terminal. The default value is none.

Init Object Defines an object that contains printer commands to be sent to the terminal immediately after the terminal establishes a session with the handler. If the printer type is set to **none**, the **init** object is not sent to the terminal. The default value is blank. Field values are limited to the available printer init objects created.

Note: The upper limit of print data is 1.5 K.

Polling/Timers Options



Collection Time Specifies the number of milliseconds the handler waits after receiving data from the host prior to processing the data. This setting is used to avoid sending intermediate screens to the terminal, resulting in unnecessary RF traffic and rapidly changing screens on the terminal. For example, many applications display the "background" data for a screen then delay while the actual data is prepared for display. Without a proper collection time setting, the terminal would display the background and be updated again with foreground data when ready. With collection time set, only one screen update is sent to the terminal, with all of the data in it. The default value is 100. Collection time should be adjusted to 30-50ms if Mode is set to char.

Radio Defines the time, in seconds, that the terminal waits for a response from the host application. After the time is exceeded, an error message is displayed on the terminal. The default value is 120.

Power Defines the time, in seconds, in which inactivity results in a power saving shutdown of the terminal power. The default value is 300.

Backlight Defines the time, in seconds, the display backlight remains on after keyboard or scanner input. The default value is zero (0).

Note: All values have a direct affect on battery life. Higher volume and/or longer inactivity times result in shorter battery life.

Initial Defines the initial poll rate (in clock ticks, where 1 tick = 55ms) between polls immediately following an RF transmission from the terminal to the host or when there is network traffic. This operation places overhead on the terminal battery. The Poll Rate becomes a factor in determining battery life. The faster the poll rate, the sooner the battery needs recharging. The radio continues to poll at the initial rate until the switch timer expires (see below). The default is 4 ticks (63 milliseconds).

Final Defines the initial poll rate (in clock ticks, where 1 tick=55ms) between polls after the switch time expires. The default value for this field is 32 ticks (approx. 500ms).

Switch Defines the amount of time (in seconds) that the terminal will poll at the initial rate.

Primary/Min Defines the initial beacon poll rate between polls immediately following an RF transmission from the terminal to the host or when there is network traffic. This operation places overhead on the terminal battery. The Poll Rate becomes a factor in determining battery life. The faster the poll rate, the sooner the battery needs recharging. The default value is 2 (200 milliseconds).

Max Only applies if an algorithm is chosen. The field defines the maximum poll rate, in 100ms increments, between polls a after an RF transmission from the terminal to the host that only applies if a polling algorithm is applied. The default is zero (0).

Algorithm The Algorithm field defines the access point beacon algorithm that the terminal will use. Beacons are broadcast every 100ms, so a value of 1 selects 10 polls per second. When the field is set to 0, it uses the value Primary/Min as the poll rate. 1 provides best performance where 10 provides the best power saving. A value of 11 or above selects a dynamic algorithm that varies from the value set in the Primary/Min field to the Max fields as defined above. When set to 11, the terminal uses the Primary/Min value as the beacon poll rate when there is network traffic for the terminal and Max as the rate when there is none. The default is 0. Value should not exceed the manufacturer's recommended value.

Keyboard Objects Group

Mapping Object Defines an object used to define keyboard mappings. Refer to the Keyboard Mapping section in the appropriate terminal user's guide for more information on the content and format of this file. The field values for this field are created with files using the Keyboard mapping object editor.

Macro Object Defines an object used to define keyboard macro settings. The field values for this field are created with files using the Keyboard macro object editor.

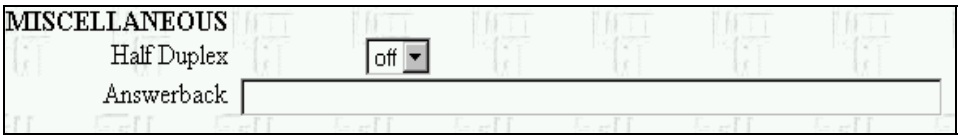
Host Interface Group

Mode Sets the communications mode. The options are block or char. When set to char, the terminal sends every keystroke to the host system. However, scanned data is sent as a string of characters. This mode of operation uses the radio at each keystroke so battery life is greatly affected. When set to block, the terminal only sends data to the host system when enter, tab, pf keys, or other keys as defined in the keyboard mapping object designated as xmit functions are pressed. The default value for this field is block.

Host Type Established session connection to host. Allowable values of this field are Serial and vtcomm. (Serial is not fully implemented, and vtcomm is for communicating with the DLL. For DLL details, see Chapter 9, Windows NT/2000 DLL Interface.)

TTY crlf proc Enables processing of incoming crlf characters. Values for field are yes or no. The default is no. (For ptty hosts only: This connects carriage returns to a carriage return/linefeed pair when used as input to the host application.)

Miscellaneous Options



Half Duplex The default value is off. If set to on, the terminal locally echoes typed characters.

Answerback The message returned to the host application in response to an Answerback request (ctrl-E). Octal escape sequences may be embedded (\nnn). The default value is blank. Inserting the sequence $1 into the answerback string causes VTerm to substitute the 3 digit IP address for Access Point terminals. The following answerback examples send the indicated response back to the host:

answerback = Hello World
Sends Hello World as a response

answerback = $1
Sends three digit ID (i.e., 065, 066, etc.)

answerback = RF$1
Sends RFXXX where XXX is ID

answerback = RF$1\015
Sends RFXXX and carriage return

Log Level Options Group

General Defines the general logging level for the handler. There are 10 log levels, from 0 to 9, with a level of 9 collecting the most information. The default value is 1.

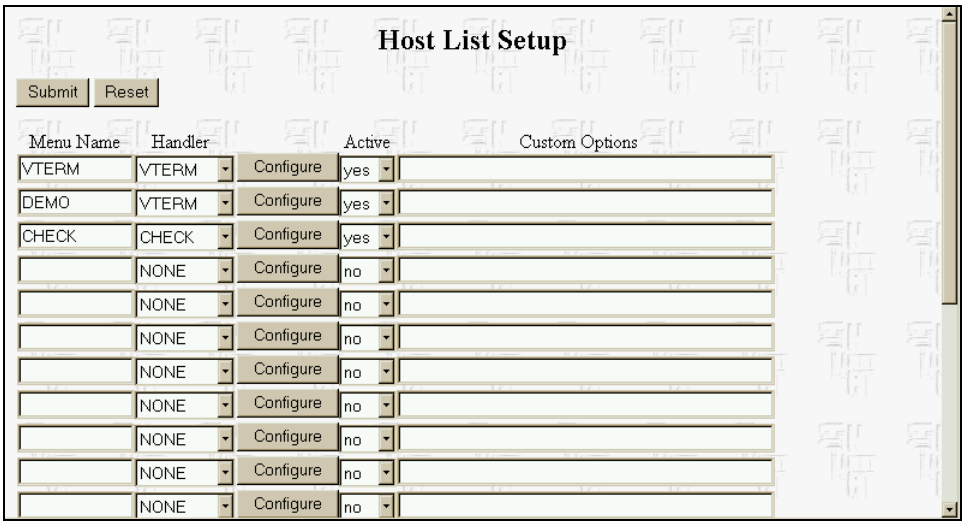
Display Defines the level of logging for host and terminal display logging. A level higher than 7 results in a hex dump of the displays. The default value is zero (0).

Formatter Defines the level of logging for the screen formatting routines. The default is zero (0).

Dialog Defines the level of logging for dialog routines. The default value is 0.

Making the Host List Entries Active

Initially the Host List option is not active. When it is configured as active, it appears in the Host List on the RF terminal.

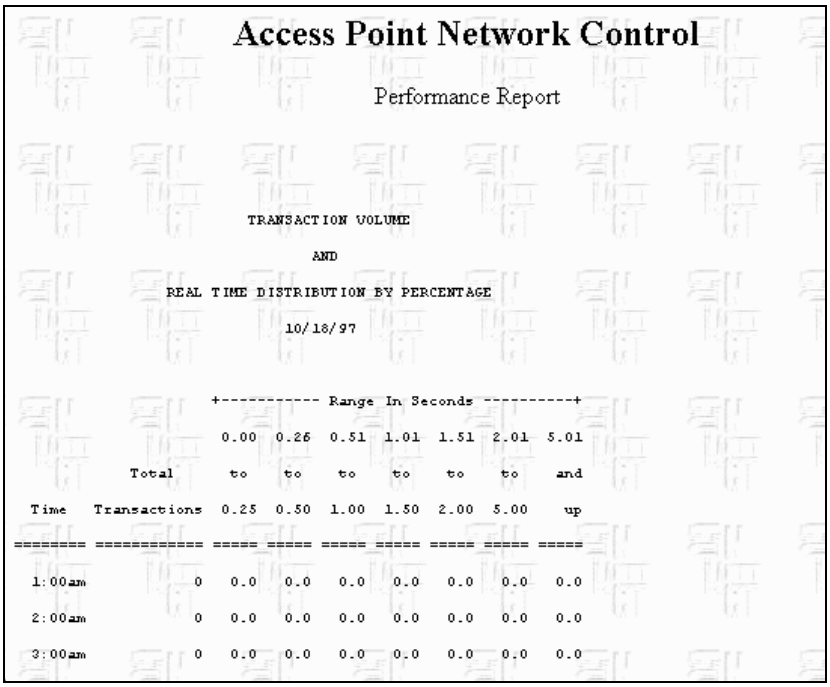


1. Use the mouse to make the option active with the drop down menu next to the Active field. This function is also useful to create special test handlers that can be hidden from normal use. The handlers can be made active when needed for testing or diagnostic purposes.
2. When finished with the handler specific settings and the Host List entries, click on the **Submit** button to save the file.
3. Double-click on the browser's **Back** button to return to the Main Menu.

Performance Reporting

The Performance Report generates network volume statistics for all nodes on the RF network. It creates an accounting file, **apprf.dat**, which does not wrap and grows to a maximum of 2 megabytes. The file can be removed at any time and is recreated at the next interval, which is the SAC interval set in the server's screen.

Reports are displayed on screen, by date. Scroll or use **Pg Dn** to display the next screen.



Access Point Network Control

Performance Report

TRANSACTION VOLUME

AND

REAL TIME DISTRIBUTION BY PERCENTAGE

10/18/97

		+---------- Range In Seconds ----------+						
		0.00	0.26	0.51	1.01	1.51	2.01	5.01
	Total	to	to	to	to	to	to	and
Time	Transactions	0.25	0.50	1.00	1.50	2.00	5.00	up
1:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3:00am	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Information is shown on an hourly basis by day and is organized into three separate sections within the report. Columns are totaled at the end of the 24-hour report. These reports give an indication about the total activity at the site.

Transaction Volume and Real Time Distribution by Percentage displays the end-to-end response time of transactions from the point that the server receives data from a terminal, passes the data to the host application, receives a response back from the host application, and receives acknowledgment from the terminal that the data was received. This report provides a realistic view of application and host response time from the end-user's standpoint.

Transaction Volume and Application Processing Time Distribution by Percentage shows the host response time from the point when the PowerNet server submits data to the host application and then receives a response from the host for that particular transaction. This report provides a true measurement of host application processing time.

Received Record Size Distribution by Percentage shows the size of the records received by the PowerNet NT/2000 server.

Transmitted Record Size Distribution by Percentage shows the size and record distribution of the data sent to the terminals by the server.

The **apprf.dat** file, from which the reports are derived, can be exported to a Windows-based program, Excel or perhaps a Visual Basic-based application, for graphical analysis.

The field definitions for each line entry of **apprf.dat** are as follows:

Field	Description/Value
1	Sequence Number
2	Elapsed Time in Seconds
3	Unit Number
4	End-End Transaction Times 0ms to 250ms (*Year)
5	End-End Transaction Times 251ms to 500ms (*Month)
6	End-End Transaction Times 501ms to 1000ms (*Day)
7	End-End Transaction Times 1001ms to 1500ms (*Hour)
8	End-End Transaction Times 1501ms to 2000ms (*Minute)
9	End-End Transaction Times 2001ms to 5000ms (*Second)
10	End-End Transaction Times over 5001ms
11	Application Transaction Times 0ms to 250ms
12	Application Transaction Times 251ms to 500ms
13	Application Transaction Times 501ms to 1000ms
14	Application Transaction Times 1001ms to 1500ms
15	Application Transaction Times 1501ms to 2000ms
16	Application Transaction Times 2001ms to 5000ms
17	Application Transaction Times over 5001ms
18	Receive Records Size 0-23 bytes
19	Receive Records Size 24-46 bytes
20	Receive Records Size 47-92 bytes
21	Receive Records Size 93-184 bytes
22	Receive Records Size 185-368 bytes
23	Receive Records Size 369-512 bytes
24	Transmit Records Size 0-23 bytes
25	Transmit Records Size 24-46 bytes
26	Transmit Records Size 47-92 bytes
27	Transmit Records Size 93-184 bytes
28	Transmit Records Size 185-368 bytes
29	Transmit Records Size 369-512 bytes

Note: Field content when Unit Number (field 3) is zero.

System Accounting File

The **apsac.dat** file from which the reports can be derived can be exported to any user-defined program. The accounting interval is controlled through the **RF Network Setup** form and is normally set to 600.

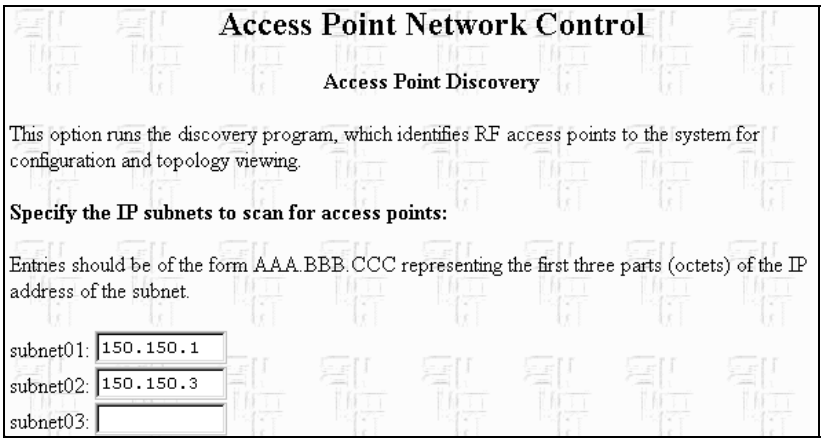
The field definitions for each line entry of **apsac.dat** are as follows:

	Field	Definition
1.	Sequence	Interval number
2.	Interval	Interval size in seconds
3.	LAN	Network number
4.	Base	Transceiver number
5.	Terminal	Terminal number
6.	LAN load	Total number of nodes attached (*Year)
7.	Base load	Total number of terminals attached (*Month)
8.	TX packets	Packets transmitted (*Day)
9.	RX packets	Packets received (*Hour)
10.	TX data	Data bytes transmitted (*Minute)
11.	RX data	Data bytes received (*Second)
12.	NAK errors	NAKs received from terminal
13.	RNAK errors	NAKs sent to terminal
14.	DUP errors	Duplicates received from terminal
15.	Timeout errors	Timeouts awaiting terminal ACKs

* Field content when unit (network, transceiver, and terminal) number is zero.

Using Access Point Discovery

Access Point Discovery is located in the Access Point part of the main menu and is listed as **AP Discover**. It is run to determine the IP addresses and manufacturers of all of the Access Points (APs) connected to your network. Several of the PowerNet functions use this information, including the topology viewer. Run AP Discover when the PowerNet software is first installed, and any time the APs are changed (added/removed/modified).



1. Enter the three high-order IP octets of each network segment to be examined in the boxes.
2. Click on the **Run Discover** button at the bottom of the screen. AP Discover steps through each possible address on each of the segments identified, looking for any APs.
3. When an AP is found, its IP address and type code are stored in a file (**ipfile.cf**) for use by other programs. Once AP Discover completes, the topology viewer can then displays information on the APs.

As an example, assume your company uses a class B IP address (i.e., 150.150) and the APs are located on the 150.150.1 and 150.150.3 segments. Enter 150.150.1 and 150.150.3 in the boxes, and then run Access Point Discovery. It searches from 150.150.1.1 through 150.150.1.254, and 150.150.3.1 through 150.150.3.254. Any APs encountered (i.e., 150.150.1.101) are written to **ipfile.cf**, and then the next address is tried.

Viewing System Logs

All Connect programs, including most of the interactive programs, provide event logging for performance and trouble analysis. In addition, each of the terminal emulation processes maintain separate log levels for specific library routines, such as presentation space management, datastream handling, and reformatting.

System logging in general is a powerful feature that allows system administrators to track data through all RF network points of the PowerNet server. The log files can be accessed through Notepad and are found in the **\powernet\diag** directory.

There are 10 levels of detail, 0 through 9, that can be collected in logs. The lowest is the default setting, zero (0), which gives only system activity information. A level of 1 is sufficient for system administrator analysis purposes.

Log levels may be set up to 9, which provides the most information. Normally, these high log level settings are used strictly for engineering analysis purposes because high levels degrade system performance.

Each level includes the logging of all levels below it. For example, level 6 includes all messages from 0 to 5. In general, log levels conform to the following conventions:

Level 0	Exceptional events only
Level 1	Data transmit and receive size
Level 2-6	Engineering analysis messages
Level 7	Data in hex format
Level 8-9	Engineering analysis messages (useful only to Connect engineering and support staff)

Note: Log files are created on an as-needed basis. Not all files listed below may appear on selection screens while using the log view tool. For example, the base status log file appears as a selection if transceivers with firmware 2.14 or higher are connected to the network.

Log Format

All processes written by Connect use the following log entry format:

MM/DD hh:mm:ss msec [message]

MM	=	Month
DD	=	Day of month
hh	=	Hour
mm	=	Minute
ss	=	Second
msec	=	Milliseconds since previous log event

When level 7 (or higher) is set, data is logged in hexadecimal format as follows:

HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH AAAAAAAAAAAAAAAA

Note: The hexadecimal value is represented above as HH, and the ASCII graphic value is represented as A. The format provides for one line per 16 bytes of data. Non-graphic characters are represented in the ASCII portion as a period.

By default, all processes "wrap" the log file at 1000 lines, which translates into roughly 60,000 bytes.

Note: Reset production systems to zero after a problem has been resolved to minimize the impact of disk logging on system performance.

Access Point Logs (apbase.log)

Access Point logs contain all RF-based transactions passed between the PowerNet server and the access point network in an access point environment. These logs can also include data sent to/from all terminals connected to these devices.

Spectrum One Logs (base.log)

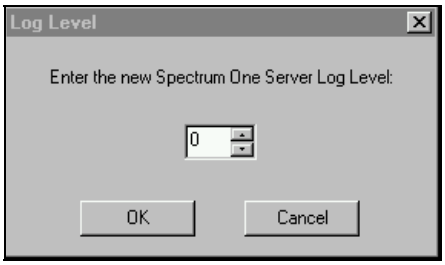
Spectrum One logs contain all RF-based transactions passed between the PowerNet server and the access point network in an access point environment. These logs can also include data sent to/from all terminals connected to these devices.

Terminal Logs (rfnnn.nnn.log)

Terminal logs contain information regarding RF data activity for individual terminals. Where the associated Access Point (base) logs contain information regarding all terminal traffic connected to the network, these logs contain information relating only to a particular device.

Setting Log Level-of-Detail

1. Select Access Point Log Level or Spectrum One Log Level from the **Logging** menu. The **Log Level** selection box appears.



2. Type in the appropriate log level and click on **OK**. This entry overrides the log level set with the Access Point or Spectrum One Setup Form. A message showing the level that was set then appears.

Setting Terminal Log Levels

Every terminal has a separate terminal log file that is kept to record terminal activity. The log level for a specific terminal log can be changed with the Terminal Log Level option.

This log details application level terminal activity, such as when data records are sent and received, etc. If a terminal appears to be malfunctioning, then the level of its log can be adjusted and the data analyzed to determine related issues. More information regarding terminal behavior on the network can be collected. A level of 7 or above provides a hexadecimal-format data dump.

1. Select the specific terminal by clicking on it in the **PowerNet Wireless View** screen.
2. Select **Terminal Log Level** from the **Logging** menu. A **Log Level** selection box similar to the one shown previously appears.
3. Enter the log level (valid entries are 0 - 9) and press **OK**. A confirming message is then displayed.

Note: If the RF terminal is rebooted, the terminal log reverts to its original setting as specified in the handler (host list) setup.

Terminal Process Log

The terminal process log file is created using the following naming convention: rf<NNN.NNN>.log, where NNN.NNN is the last two octets of a terminal's IP address. Each terminal emulation handler process provides one or more of the following log levels in addition to the standard debug log level:

- Dialog Dialog script
- PS Presentation space
- Formatter Screen formatter

All of these levels force a message output to the terminal process log file.

Viewing RF Device Statistics

By double-clicking on any device in the topology, the operator can view the statistics for that individual RF device. The Unit Statistics appear in a floating window that will stay open, but still allow other actions to be performed.

Unit Statistics Fields

Terminal - Displays the network ID (IP address) of the selected device.

TX Host Packets -The number of IP packets transmitted to the host application.

TX Bytes -The number of bytes transmitted from the RF device.

RX Host Pkts -The number of IP packets received from the host application.

RX Bytes -The number of bytes received by the RF device.

Chapter 7 • RF Networks

OpenAir 400 manages all attached RF LANs and completes the data path between the terminals and host. This chapter reviews Access Point networks along with OpenAir 400 RF management features.

About OpenAir 400

The OpenAir 400 server performs the controlling and maintenance functions for the RF LAN, and provides access to the host. At times, the OpenAir 400 server can also act as a host with the application program(s) installed on its hard disk. The multitasking Windows NT/2000 operating system makes this possible.

Note: Caution must be used in multitasking situations.

There are several different models of OpenAir 400, but all provide the same basic functions that are necessary for proper operation of the wireless LAN. These include:

- Management of all radio functions. Assignment of 6 RF operating channels is made to each transceiver, cradle, and RF terminal. One chipping sequence is assigned to all devices so they all can decipher RF transmissions on a per LAN basis.
- Data packaging of terminal information sent to the host and proper formatting of replies sent to terminals. This is in accordance with specific connectivity protocol(s) being used.
- Address management for LANs, transceivers, terminals, and cradles using Spectrum One and Access Point definitions.
- Serial connections for attaching to the loading cradles and the first transceiver of each Spectrum One RF LAN (sometimes referred to as the serial transceiver or serial base). Up to 32 serial ports for RF LANs can be installed on each OpenAir 400.
- Ethernet connectivity to TCP/IP networks for terminal access via Access Points or Spectrum One transceivers.

Using Access Points for Linking

Access Points (APs) are devices that provide a wireless communications link between remote RF terminals and an Ethernet network. The Ethernet backbone contains the host, an OpenAir 400 server, which provides data management and/or optimization functions for the access points.

Similar to transceivers, access points contain a radio unit and an antenna for communication with terminals. Access points are connected to a coaxial Ethernet backbone (10base2) or to a hub using twisted pair wiring (10baseT). In some cases, access points can also support PPP connections to a TCP/IP network. These access points and the data transferred to and from the remote terminals are managed by the OpenAir 400 server.

RF Terminals for Collecting Data

Various models of RF terminals can be used, some with scanners, some wrist-mounted, others attached to vehicles. All have radios that communicate with transceivers or access points and have screens and keyboards that allow the user to log in to the system.

The main function of a terminal is to send collected data to the host and respond to replies from the host. Communication is accomplished via RF transmissions to and from the transceiver/access point, which communicates with the OpenAir 400 server.

Terminal Software

All terminals must have a program loaded in order to operate on a wireless network. The commands are issued by host applications and/or the OpenAir 400 server communicating with the terminal. The following terminal emulation protocols are supported by OpenAir 400: vt100, vt220, hp700, tn5250, and tn3270. OpenAir 400 also supports interfacing directly to Windows NT/2000-based applications via an application program interface (API).

Terminal Cradles

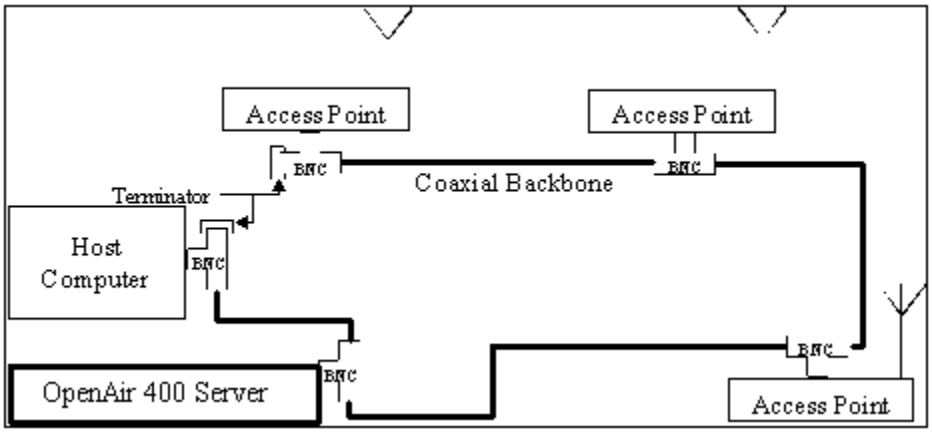
Terminal cradles recharge the batteries for RF terminals and download files and information to the terminals. Software files that provide operating instructions for terminals can be downloaded from the OpenAir 400 server through the cradles to terminals. RF channel selection and chipping sequences are also given to terminals when they are placed in cradles for configuration.

Terminal cradles are connected to the OpenAir 400 server via serial ports. Sometimes referred to as the cradle network, some cradles can be daisy chained together if there is a need for charging many terminals.

Cabling the LAN

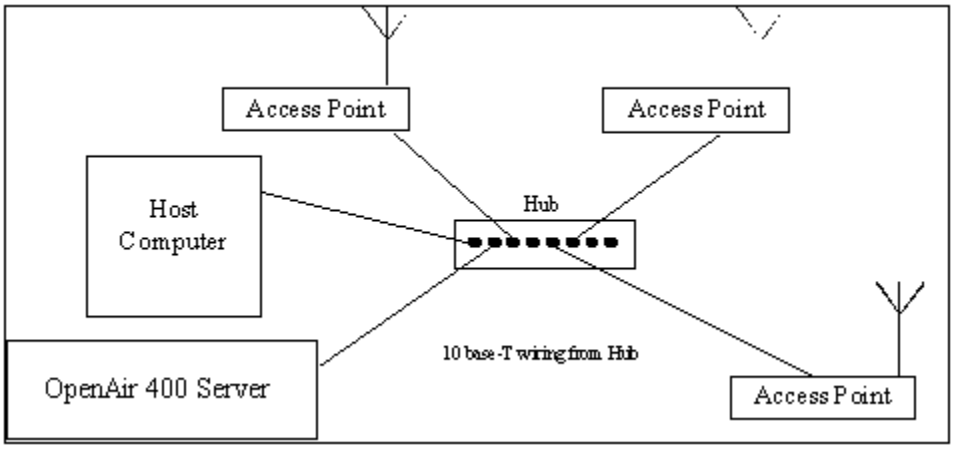
The following diagram shows a simple LAN connected with a coaxial-backbone cable. Access Points and the OpenAir 400 server are attached to the coaxial cable with BNC T-connectors.

Access Points use standard Ethernet cabling and can use conventional Ethernet wiring schemes. The backbone coaxial cable length can extend up to 600 feet using coaxial cable. Distances may be increased with repeaters if needed.



The diagram below shows a simple Hub-LAN connected with 10 base-T Ethernet. Access Points and the OpenAir 400 server are attached to the Hub with 10 base-T Ethernet wiring.

The cable length is limited by the standard Ethernet constraints.



Converter Pinouts

When RS-232 to RS-422 converters are used with Spectrum One networks, the cable pinouts are as follows:

Signal	Converter Pin Number	Transceiver Pin Number
+ data from transceiver	17	11
- data from transceiver	5	13
- data to transceiver	2	23
+ data to transceiver	14	25
signal ground (shield)	shield	shield

MultiLAN Capabilities

With the use of multiport adapters, 2, 4, 8, 16, or 32 serial ports for RF LANs can be added to an OpenAir 400. These serial ports can be configured for transceiver and cradle LANs, which are all managed by the OpenAir 400 server.

WANs

Wide area networks (WANs) can be implemented when the RF LAN(s) must be in a location remote from the OpenAir 400 server, for example, when the host connection must be directly attached to the OpenAir 400 server and the site is miles away.

Leased and T1 lines, as well as satellite and microwave links, have been used to connect serial ports on the OpenAir 400 server to the first transceiver of remote Spectrum One networks. Ensure that there is no latency in the data communications equipment and that it does not require flow control.

Assigning Component IP Addresses

Access Points are assigned IP addresses based on the network to which they are attached. Since the display software uses the last two octets of the IP address to identify equipment, it is suggested (though not required) that all Access Points and Stations be assigned IP addresses that are unique in these two octets (i.e., 206.181.*90.1* and 206.181.*90.2*).

RF Communications

Radio frequency (RF) wireless communications have proven to be one of the fastest and most accurate ways of collecting data in situations that require user mobility. Spread Spectrum and narrowband communications can be used to fill the RF requirement, but Spread Spectrum is proving to be the radio technology of choice.

Spread Spectrum Technology

Originally developed by the military, Spread Spectrum technology is now used for business sites such as warehouses, department stores, airports, shipping companies, or any business where data connection radio technology is beneficial.

Spread Spectrum use does not require an FCC license. It is a redundant signal and therefore very robust for data delivery. It supports multiple users, withstands interference, and is resistant to interception and decoding by outside sources.

The data rate for the RF portion of the data path is 60,000 bps. The Spread Spectrum frequency range is 902 – 928 MHz and, using the spreading technique, a 1 MHz bandwidth is employed. The available frequency range has been divided up into 53 channels (49 usable), to make frequency selection easier. The signal level is low and additional range is added to the system with transceivers.

Direct sequence is used by the Spectrum One Network to distribute an output signal over the 1 MHz bandwidth. This requires a chipping sequence, which is a pseudo-random pattern of ones and zeros, to produce the output signal. Use of this technology prevents others from deciphering the transmitted data/signal.

Conducting Site Surveys

Prior to installing Spectrum One Network or Access Point equipment, a site survey should be conducted. Radio receptions tests help determine the number of transceivers required for the site as well as their optimum placement. This, in turn, will help define cable lengths and installation requirements.

Note: The transaction rate capacity of transceivers must be verified. Although one transceiver may cover a large area, it will not support 50 RF terminals with a high volume transaction rate.

The site survey will provide information about RF coverage of portable terminals from all locations at the site. The quietest channels (those with the least interference) are selected, and the network is established accordingly.

As RF interference characteristics can vary over time (e.g., the store next door may start using RF equipment), the OpenAir 400 server software includes features that allow you to determine RF activity and change channels as needed.

Using RFSurvey

On Spectrum One Systems, the terminal software includes the ability to run another application, RFSurvey, to check terminal-transceiver communication. The software runs a loopback test between the terminal and the OpenAir 400 server. It displays the particular transceiver that is communicating with the terminal for the current test and assigns an “average” grade.

This grade indicates how good coverage is. Tests can be run with different transceivers from varying distances. Dead spots can be identified easily with this option.

RF Topology for More Than One LAN

Two types of RF topology, seamless and segmented, can be implemented when more than one RF LAN is set up. The ability to assign the same, or different, channels and chipping sequences makes this possible.

In large installations, a combination of both of these RF topological strategies can be used. For example, LAN 0, 1, 2, and 3 may be seamless transceiver LANs, while 4, 5, and 7 may be segmented transceiver LANs. Then 8, 9, 10, and 11 may be used for cradle LANs.

Seamless LAN

In the seamless LAN topology, two or more LANs are set to identical channels and chipping sequences, creating redundant transceiver networks in the same location. In addition, physically separated LANs can use this design to provide seamless coverage between networks. Terminals communicating with a LAN in one location can be moved to another, which has a different LAN, without affecting active terminal sessions.

Segmented LAN

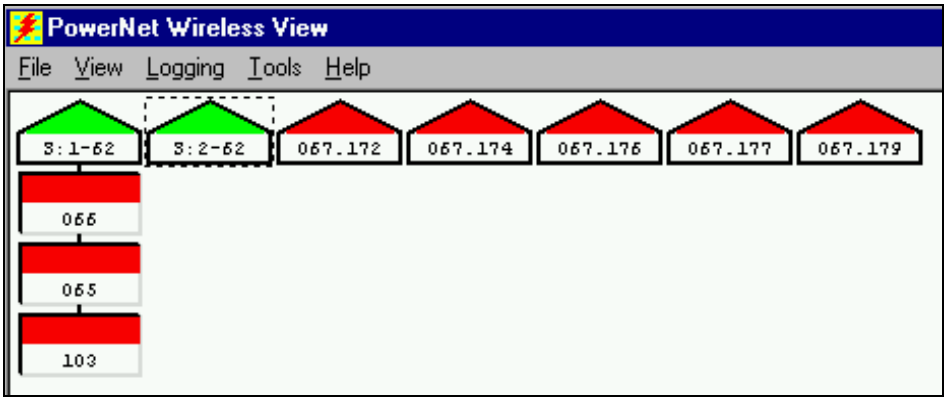
Segmented LAN topology has two or more LANs set to different selections of channels and different chipping sequences. This RF topology option can be used to divide and distribute RF traffic, thus optimizing network bandwidth. Terminals communicating with transceivers on one network, will not be “heard” by the transceiver on other network(s).

To make terminal configuration easier in this case, two (or more) cradle networks can be assigned the different channels and chipping sequences. If only one cradle network is available, the groups of terminals for each LAN must be configured separately to match the channels and chipping sequences in use. Note that segmentation can be full or partial.

Note: When managing segmented LANs, the cradle channel list and chipping sequence can be changed without resetting the RF network. The OpenAir 400 server default channel and chipping sequence settings are acceptable for the majority of single or seamless multiple LAN installations. However, in the event that Frequency Scan Reports indicate high levels of interference on one or more of the selected channels, it may be necessary to modify the channel lists. Access Point systems can achieve the same seamless and segmented configuration by setting the RF parameters as indicated in the Access Point User’s Guide.

Physical LAN Topology

The OpenAir 400 server software provides a visual display, similar to the one shown, of current RF LAN topology with the Wireless View option. The display is useful for diagnosing backbone cabling problems, malfunctioning transceivers, and terminals that are not performing properly.



The display shows transceivers that are a part of, and terminals that are logged into, each LAN. In this figure, S0–03 shown in the shaded area is the serial transceiver on LAN 0. Terminal 066 (positioned below this transceiver) is communicating with transceiver S0–03. Transceiver C0–04 is attached to the coaxial backbone cable of LAN 0, “downstream” from the serial transceiver.

Terminals and transceivers are color coded to indicate the last time communication took place with the RF network. The colors and definitions are listed below.

Symbol	Definition
green	Communicated within the last 1 minute
blue	Communicated within the last 2–5 minutes
cyan	Communicated within the last 6–10 minutes
yellow	Communicated within the last 11–30 minutes
red	No communication within the last 31 minutes

Wireless View is used to monitor transceivers as they are being powered-on, one at a time. It is also used to display terminals as they are configured. If an address conflict occurs, it will be readily apparent and easily resolved.

Statistics can be displayed for components that are selected. In addition, other features such as a network summary and tests are available from Wireless View.

System Testing

Several software testing and diagnostics tools are included in the OpenAir 400 server package. Please see the specific product manual or on-line help for instructions on the use of those products.

The diagnostics facility provides a means of testing transceivers by sending request packets from the OpenAir 400 server to transceivers. The transceiver(s) then return(s) reply packets with data from various internal registers, giving overall error rates.

Controlling the Terminal Scanner

This section documents the methods used to control the terminal scanner both from host program control and from the controller by use of a control file. It describes the parameters used to control the active bar codes when using VTerm.

In all emulation modes (VT-100, VT-220, HP700/92), a configuration file can be created using the object editor, which sets the active bar codes whenever a handheld is brought online. In addition, when using the VT-100/220 emulation, there is an extended command set, which allows issuing escape sequences having most of the functionality of the configuration file. As an example, this capability could be used to turn on UPC-A when prompting the user to scan the UPC code on a case. Then UPC-A could be turned off and I2of5 turned on to input the packing code. This prevents the user from scanning the wrong type of bar code for a given field.

Add the appropriate command lines from the following lists. The only rule on sequencing the lines is that the *state* command (i.e., dc_codabar_state) must be the first line for that code type.

The following example file enables the host TEST to scan code-39 in full ASCII mode and UPC-A with optional 5-digit supplementals.

file test.cfb

```
dc_code39_state=1;
dc_code39_minlen=1;
dc_code39_maxlen=30;
dc_code39_depend=1;
```

```
dc_upca_state=1;
dc_upca_minlen=0;
dc_upca_maxlen=1;
dc_upca_depend=2;
```

Note: All lines must be terminated with a semicolon (;).

Be aware of the following rules when enabling/disabling bar codes:

- When a handheld is brought online, the emulator checks for the presence of a bar code command file. If the file exists, the first action taken is to disable ALL bar codes so you must add command lines to turn back on all of the codes desired. If the file does not exist or does not contain any commands, then ALL bar codes are enabled.
- When turning a code on, **minlen**, **maxlen** and **depend** all default to zero. If the zero state is not wanted, add the modifying line.
- All other commands are NOT reset when the handheld is rebooted. For example, if you turn editing **on** and then decide later that it should be **off**, you must include the disable command in your control file.

CODABAR

dc_codabar_state – Disable/enable codabar scanning.

0 – Disabled
1 – Enabled

dc_codabar_minlen – Minimum valid bar code length.

1–54

dc_codabar_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_codabar_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – Codabar redundancy disabled
1 – Codabar redundancy enabled

dc_notis_edit – Remove the stop and start characters from codabar prior to transmission to the host.

0 – Disabled
1 – Enabled

dc_clsi_edit – Change 14 digit codabar into 17 digits by adding spaces after the first, fifth and tenth digits. Additionally, the start and stop characters are removed.

0 – Disabled
1 – Enabled

VT-100 Extended Command: <ESC>[?0;<state>;<minlen>;<maxlen>;0z

CODE 11

dc_code11_state – Disable/enable code 25

0 – Disabled
1 – Enabled

dc_code11_minlen – Minimum valid bar code length.

4–54

dc_code11_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

4–54

dc_code11_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – Disabled (single scan sufficient)
1 – Enabled (two valid scans required)

dc_code11_depend – Number of check digits.

0, 1 or 27

dc_code11_rptchk – Should check digits be transmitted to the host.

0 – Do not transmit
1 – Transmit check digits

VT-100 Extended Command: <ESC>[?11;<state>;<minlen>;<maxlen>;<depend>z

CODE 128

dc_code128_state – Disable/enable code 128 scanning.

0 – Disabled
1 – Enabled

dc_code128_minlen – Minimum valid bar code length.

1–54

dc_code128_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_code128_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – Code 128 redundancy disabled
1 – Code 128 redundancy enabled

VT-100 Extended Command: <ESC>[?2;<state>;<minlen>;<maxlen>;0z

CODE 39

dc_code39_state – Disable/enable code 39 scanning.

0 – Disabled
1 – Enabled

dc_code39_minlen – Minimum valid bar code length.

1–54

dc_code39_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_code39_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – Disabled
1 – Redundancy enabled

dc_code39_chkb – Code 39 check digit.

0 – Check digit ignored if present and just transmitted to the host as data
1 – Check digit required and must be valid

dc_code39_depend – Disable/enable full ASCII code 39 interpretation.

0 – Disabled
1 – Full ASCII enabled

VT-100 Extended Command: <ESC>[?3;<state>;<minlen>;<maxlen>;<depend>z

CODE 49

dc_code49_state – Disable/enable code 49 scanning.

0 – Disabled
1 – Enabled

VT-100 Extended Command: <ESC>[?4;<state>;0;0;0z

CODE 93

dc_code93_state – Disable/enable code 93 scanning.

0 – Disabled
1 – Enabled

dc_code93_minlen – Minimum valid bar code length.

1–54

dc_code93_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_code93_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – disabled
1 – enabled

VT-100 Extended Command: <ESC>[?5;<state>;<minlen>;<maxlen>;0z

I 2 of 5

dc_coded25_state – Disable/enable code 25.

0 – Disabled
1 – Enabled

dc_coded25_minlen – Minimum valid bar code length.

1–54

dc_coded25_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_coded25_red – Is one valid laser sweep sufficient or does the scanner have to +make two valid sweeps (redundancy).

0 – d2of5 redundancy disabled (single laser scan sufficient for validation)
1 – d2of5 redundancy enabled (two laser scans required for validation)

VT-100 Extended Command: <ESC>[?6;<state>;<minlen>;<maxlen>;0z

INTERLEAVED 2 OF 5

dc_codei25_state – Disable/enable interleaved 2 of 5 scanning.

0 – Disabled
1 – Enabled

dc_codei25_minlen – Minimum valid bar code length.

1–54

dc_codei25_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

1–54

dc_codei25_red – Is one valid laser sweep sufficient or does the scanner have to make two valid sweeps (redundancy).

0 – Interleaved 2of5 redundancy disabled
1 – d2of5 redundancy enabled

VT-100 Extended Command: <ESC>[?7;<state>;<minlen>;<maxlen>;0z

EAN-13

dc_ean13_state – Disable/enable EAN-13 scanning.

0 – Disabled
1 – Enabled

VT-100 Extended Command: <ESC>[?8;<state>;0;0;0z

EAN-8

dc_conv_ean8to13 – Should ean8 bar codes be zero padded to 13 digits.

0 – Do not pad
1 – Add padding zeroes

dc_ean8_state – Disable/enable EAN-8 scanning.

0 – Disabled
1 – Enabled

VT-100 Extended Command: <ESC>[?9;<state>;0;0;0z

PLESSEY/MSI

dc_msi_state – Disable/enable code MSI scanning.

0 – Disabled
1 – Enabled

dc_msi_minlen – Minimum valid bar code length.

4–54

dc_msi_maxlen – Maximum valid bar code length. The maximum length must be greater than or equal to the minimum length.

4–54

dc_msi_red – Is one laser scan sufficient or does the scanner have to make two valid scans (redundancy).

0 – Single scan sufficient
1 – Two valid scans required

dc_msi_depend – Number of check digits.

0, 1 or 2

dc_msi_rptchk – Should check digits be transmitted to the host.

0 – Do not transmit
1 – Transmit check digits

VT-100 Extended Command: <ESC>[?10;<state>;<minlen>;<maxlen>;<depend>z

UPC-A

dc_upca_state – Disable/enable UPC-A scanning.

0 – Disabled
1 – Enabled

dc_upca_chkb – Should the check digit be transmitted to the host.

0 – Do not transmit
1 – Transmit check digit

dc_upca_preamble – Transmit no prefix characters (0), prefix the number system for the code type (1), or prefix a zero followed by the code type number system (2).

0 – No prefix
1 – Prefix the code type number system
2 – Prefix 0 followed by the code type number system

VT-100 Extended Command: <ESC>[?13;<state>;0;0;0z

Note: In order to control supplementals, use the Supplemental Bar Code Handling sequences presented later in this chapter.

UPC-E0

dc_upce0_chkb – Should the check digit be transmitted to the host.

0 – Do not transmit
1 – Transmit check digit

dc_upce0_depend – Expand 6 digit UPC-E0 to a 12 digit UPC-A .

0 – Do not expand
1 – expand to UPC-A

VT-100 Extended Command: <ESC>[?14;<state>;<minlen>;<maxlen>;<depend>z

UPC-E1

dc_upce1_chkb – Should the check digit be transmitted to the host.

0 – Do not transmit
1 – Transmit check digit

dc_upce1_depend – Expand 6 digit UPC-E0 to a 12 digit UPC-A.

0 – Do not expand
1 – Expand to UPC-A

dc_upce1_preamble – Transmit no prefix characters (0), prefix the number system for the code type (1), or prefix a zero followed by the code type number system (2).

0 – No prefix
1 – Prefix the code type number system
2 – Prefix 0 followed by the code type number system

VT-100 Extended Command: <ESC>[?15;<state>;<minlen>;<maxlen>;<depend>z

General UPC Options

dc_linear_upc – Require decoding UPC blocks in a single sweep of the laser.

0 – Multiple sweeps are allowed
1 – UPC must decode in a single sweep

dc_upcean_security – How stringent should the UPC decode algorithm be.

0 – 3

dc_conv_upce1to2a – Expand 6 digit UPC-E1 to a 12 digit UPC-A.

0 – do not expand
1 – expand to UPC-A

dc_upce_preamble – transmit no prefix characters (0), prefix the number system for the code type (1), or prefix a zero followed by the code type number system (2).

0 – No prefix
1 – Prefix the code type number system
2 – Prefix 0 followed by the code type number system

Supplemental Bar Code Handling (UPC/EAN) Options

dc_supps_state – This flag should always be set to one.

dc_supps_minlen – Two character supplementals enabled/disabled (only applies to transmit mode 1).

0 – No codes with two character supplementals decoded
1 – Codes with two character supplementals decoded

dc_supps_maxlen – Five character supplementals enabled/disabled (only applies to transmit mode 1).

0 – No codes with five character supplementals decoded
1 – Codes with five character supplementals decoded

dc_supps_depend – Supplementals transmit mode.

0 – No supplementals transmitted to host, supplemental enable/disable ignored
1 – Codes with the appropriate supplemental (from minlen/maxlen) decoded
2 – All codes sent as is, supplemental enable/disable ignored

dc_supp_max – How many passes should be made to attempt to decode UPC supplementals.

2-10

VT-100 Extended Command: <ESC>[?12;<state>;<minlen>;<maxlen>;<depend>z

General Bar Code Options

dc_xmit_codeid – If non-zero, prefix the scanned bar code with the indicated character as shown below:

A – UPC, UPCE, UPCE1, EAN13, EAN8
B – Code39
C – Codabar
D – Code 128
E – Code 93
F – Interleaved 2 of 5
G – D 2 of 5, IATA
H – Code 11
J – MSI

0 – disabled
1 – enabled

dc_bidir_red – For any bar codes that have redundancy enabled, the two sweeps must be in opposite directions.

0 – Same direction scans sufficient
1 – Opposite direction scans required

The VT-100 extended command has no equivalent in the configuration file. It is used to disable ALL bar codes on the handheld. Typically, this would be issued before turning on an individual bar code type just to make sure of the state of the active bar codes.

VT-100 Extended Command: <ESC>[?z

Chapter 8 • Windows NT/2000 Active X Control

Description

Connect, Inc. supplies an interface to Connect's Windows NT/2000-based Wireless Network Server, OpenAir 400. This document presents details about the RFCurses ActiveX control (RFCursesCtrl.ocx) supplied by Connect, Inc. for interfacing.

The application RFCurses is designed to support an application environment in which one application controls all RF terminals, or in which there is a separate copy of the application program for each terminal.

The functions are almost identical to RFCurses.dll except for the *crf_* at the beginning of the names and the parameter types. Other differences are noted.

Note: Terminal application developers wishing to interface with VTerm should use the Connect supplied RFCursesCtrl.ocx, RFCurses.dll, and vtcomm.dll.

The routines prefixed with mv require y and x coordinates to which to move before performing the appropriate action. The mv() routines imply a call to move() before the call to the other routine. The letter "y" always refers to the row (of the window), and x always refers to the column. The upper left corner is always (0,0), not (1,1).

In addition to drawing characters on the screen, video attributes may be included, which cause the characters to be underlined or shown in reverse video on terminals that support such display enhancements. Line drawing characters may be specified to be output. On input, RFCurses is also able to translate arrow and function keys that transmit escape sequences into single values. All routines return either the short CRF_ERR or the short CRF_OK, unless otherwise noted.

See the Windows NT/2000-specific routines section for additions to the curses standard for handling RF terminals in the NT/2000 environment.

Note: An ActiveX control refers to functions as Methods.

Architecture

If all terminals are controlled by one application, which must be running, and must have called Initialize () before a terminal logs on. The application uses the term parameter with most RFCurses functions to specify which terminal is being affected by that call. The application can use multiple threads to facilitate handling multiple terminals.

If multiple copies of the application are used, each copy must handle only one terminal. Connect's VTerm program spawns a copy of the application when a terminal initially logs on and communicates only that terminal's data to that copy of the application. When the application is started, it should load RFCursesCtrl.ocx and immediately call Initialize (). Then it must immediately be available to accept a connection from VTerm using either CursesRecv() or WaitForConnection(). The term parameter, used in many RFCurses calls, is always the same value, which is that returned by CursesRecv() or WaitForConnection().

General Routines

The following documents the curses calls implemented for RFCursesCtrl.ocx:

short Initialize(boolean UseActivexEvents)

The first routine called should always be Initialize(). This initializes all data structures. This is the same as Initialize in RFCurses except for the new parameter UseActivexEvents. If set to false, the control works like RFCurses when using the WaitHandle routines for detecting the receipt of terminal data or a terminal logon. If set to true, the control notifies the user's application of terminal data or logons by using ActiveX Events. The events are described in the section called Events (see below).

short movecursor(short term, short y, short x)

The cursor associated with the window is moved to line (row) y, column x. The position specified is relative to the upper left corner of the window, which is (0, 0).

short refreshterm(short term)

Causes all current RFCurses commands to be transmitted immediately to the terminal. The VTERM terminal handler will automatically send commands to the terminal after a short inactivity timeout, but this command causes VTERM not to wait for the timeout, but to go ahead and send the commands right away. This function is optional, but can result in slightly better system performance.

short CleanUp()

A program should always call short CleanUp() before exiting.

Events

The control can fire two different events if turned on by the Initialize method.

Connection - This event is fired when a terminal logs in. The application should call CursesRecv to process the logon.

TerminalData - This event is fired when a terminal transmits data to the server. The application should call short CursesRecv to process the data.

Output Routines

The following routines are used to manipulate text in windows.

short addch(short term, short ch)

short mvaddch(short term, short y, short x, short ch)

The character ch is put into the window at the current cursor position of the window and the position of the window cursor is advanced. At the right margin, an automatic newline is performed. At the bottom of the scrolling region the scrolling region is scrolled up one line.

If ch is a tab, newline, or backspace, the cursor is moved appropriately within the window. A newline also does an clrtoeol() before moving. Tabs are considered to be at every fourth column. If ch is another control character, it is drawn in the ^X notation (ctrl-X notation).

Video attributes can be combined with a character by ORing them into the parameter. This results in these attributes also being set. These attributes are combined with any attribute settings based on previous attroff, attron, attrset, standend, and standout calls. See standout(), etc., below.

short addstr(short term, string str)

short mvaddstr(short term, short y, short x, string str)

These routines write all characters of the null-terminated character string str on the given window. This is equivalent to calling short addch() once for each character in the string. At the right margin, an automatic newline is performed. At the bottom of the scrolling region the scrolling region is scrolled up one line.

short attroff(short term, long attrs)

short attron(short term, long attrs)

short attrset(short term, long attrs)

short standend(short term)

short standout(short term)

These routines manipulate the current attributes of the named window. These attributes can be any combination of the constants A_STANDOUT, A_REVERSE, A_BOLD, A_DIM, A_BLINK, and A_UNDERLINE. These attributes can be combined with the logical OR (|) operator.

The current attributes of a window are applied to all characters that are written into the window with addch() or addstr(). Attributes are a property of the character, and move with the character through any scrolling and insert/delete line/character operations. To the extent possible on a particular terminal, they are displayed as the graphic rendition of the characters put on the screen.

attrset() sets the current attributes of the window to attrs. attroff() turns off the named attributes without turning on or off any other attributes. attron() turns on the named attributes without affecting any others. standout(term) is the same as short attron(). short standend() is the same as short attrset(), that is, it turns off all attributes.

short beep(short term)

This routine is used to signal the terminal user. beep() sounds the audible alarm on the terminal.

short erase(short term)

These routines copy the character blank to every position in the window.

short clrtobot(short term)

All lines below the cursor in this window are erased. In addition, the current line to the right of the cursor, inclusive, is erased.

short clrtoeol(short term)

The current line to the right of the cursor, inclusive, is erased.

short delch(short term)

short mvdelch(short term, short y, short x)

The character under the cursor in the window is deleted. All characters to the right on the same line are moved to the left one position and the last character on the line is filled with a blank. The cursor position does not change (after moving to (*y*, *x*), if specified).

short deleteln(short term)

The line the cursor is on in the window is deleted. All lines below the current line are moved up one line. The bottom line of the window is cleared. The cursor position does not change.

short insch(short term, short ch)

short mvinsch(short term, short y, short x, short ch)

The character *ch* is inserted before the character under the cursor. All characters to the right are moved one space to the right, losing the rightmost character of the line. The cursor position does not change (after moving to (*y*, *x*), if specified.

short insertln(short term)

A blank line is inserted above the current line, and the bottom line is lost.

Extended Routines

short printmode(short term, Boolean on)

Turns on or off printing to a printer attached to the terminal. If on is 1, any subsequent calls to addch(),addstr(), or printw() sends the data specified to the printer instead of the terminal display until print_mode is called again for that terminal with on set to 0.

short cursormode(short term, Boolean on)

Controls whether or not the cursor is visible on the terminal. Calling with *on* set to 1 causes the cursor to be displayed. Calling with *on* set to zero causes the cursor to be hidden.

short barcode(short term, long code, long status, short minlen, short maxlen, long depend)

#defines as follows.

Name	Value (decimal)
DC_ALLOFF	-1
DC_CODABAR	0
DC_CODE_11	1
DC_CODE_128	2
DC_CODE_39	3
DC_CODE_49	4
DC_CODE_93	5
DC_CODE_D25	6
DC_CODE_I25	7
DC_EAN_13	8
DC_EAN_8	9
DC_MSI	10
DC_PDF_417	11
DC_SUPPS	12
DC_UPC_A	13
DC_UPC_E0	14
DC_UPC_E1	15
DC_UPCAADDON	16
DC_UPCE0ADDON	17
DC_IATA	18
DC_DISABLE	0
DC_ENABLE	1

short setbeep(short term, long beep_dur, long beep_frq, long sbeep_dur, long sbeep_frq)

Sets the frequency and duration of beeps on the terminal for application generated and scanner generated tones.

beep dur Duration in milliseconds of a standard application beep generated with the beep() function.

beep_frq Frequency in hertz of a standard application beep generated with the beep() function.

sbeep_dur Duration in milliseconds of a scan decode beep.
sbeep_frq Frequency in Hertz of a scan decode beep.

short prompt(short term, short y, short x, short length, long attrib)

Displays an input prompt on the terminal for data entry using the terminal keyboard or bar code scanner. The prompt starts at the coordinates specified by *y,x*, and will have a maximum length of length characters. Functionality of the prompt can be altered by setting prompt attributes using the attrib parameter. The following prompt attributes are defined and multiple attributes can be set by Or'ing them together.

Input Characters

Name	Value (hex)	Description
PROMPT_ALPHA	001	Only alphabetic characters can be entered.
PROMPT_NUMERIC	002	Only numeric characters can be entered.
PROMPT_ALPHANUM	004	Only alphabetic and numeric characters can be entered.
PROMPT_ANYCHAR	008	Any characters can be entered (default).

Input Devices

PROMPT_KEY	010	Data can only be entered using the keyboard.
PROMPT_SCAN	020	Data can only be entered using the scanner.
PROMPT_KEYORSCAN	080	Data initially accepted from either keyboard or scanner. Scanning not allowed on partially keyed field. If operator clears field then scanning is again allowed.
PROMPT_KEYANDSCAN	040	Data initially accepted from either keyboard or scanner. Scanning allowed on partially keyed field. Keyed data discarded and scanned data accepted. (default)

Other

PROMPT_HIDDEN-	100	Data typed not displayed on terminal screen. Typically used for a password prompt.
PROMPT_NOLINE	200	By default underscores displayed to indicate position of prompt. Attribute turns off display of underscores.
PROMPT_AUTOENTER	300	This attribute causes data to be transmitted from the terminal as soon as the prompt is filled, without the user having to press the enter key.

short fontsize(short term, Boolean doublehigh, Boolean doublewide)

Sets the terminal screen to display in double high and/or double wide character mode. Calling with the parameters set to 1, turns on double high and/or double wide. Calling with the

parameters set to 0, turns off double high and/or double wide. An application can call *query()* to determine the number of rows and columns currently available on the terminal.

short query(short term, string termtype, string termver, string termkeys, long rows, long cols)

Receives terminal specific data, including the number of rows and columns on the terminal screen, the number of keys on the terminal keyboard, the type of terminal, and the version of the terminal software.

termtype	Type of terminal in use
termver	Version of the terminal software
termkeys	Number of keys on the terminal keyboard
rows	Number of rows on the terminal display
cols	Number of columns on the terminal display

Windows NT/2000-Specific Routines

short WaitForConnection(short pterm, long timeout)

This call causes the program to wait the specified timeout period for an RF terminal to connect to the control. A connection occurs whenever a terminal reboots and runs the VTERM application. The terminal identifier will be stored in the variable pointed to be *pterm*. This value should be used for all future *RFCurses* calls for this terminal. The return values are as follows.

Name	Description	Return
CRF_OK	A terminal connected to the dll	0
CRF_TIMEOUT	Call timed out	2
CRF_ERR	Error occurred	1
CRF_PARAM	One or more parameters invalid	3

OLE_HANDLE GetConnectWaitHandle()

Returns a standard Windows NT/2000 event handle, which is signaled when a terminal connects to the RFCurses dll. This handle can be used in any Win32 WaitFor… functions. When the event handle becomes signaled, the application should immediately call short WaitForConnection().

short Disconnect(short term)

Terminates the connection with the VTERM application associated with the specified terminal. The return values are as follows.

Name	Description	Return
CRF_OK	Call completed successfully	0
CRF_PARAM	One or more parameters are invalid	3

short CursesRecv(short type, short term, string buf, long terminator, short source, string symbology, long plen, long timeout)

This call causes the program to wait the specified timeout period for an RF terminal to connect to the control, or for data to be received for a terminal as specified by the *type* parameter. *type* can have one of the following values:

Name	Value	Description
RECV_ANY	-1	Any terminal connects or sends data.
RECV_CONNECT	-2	Any terminal connects.
RECV_DATA	-3	Any terminal sends data or type, can specify a particular terminal ID and returns only when data is received from that terminal.

term Receives the value of the terminal which caused the call the return. (i.e., The terminal that connected or sent data).

Len Must initially contain the length of the buffer pointed to by buf and is changed to the number of bytes written to buf.

terminator Receives the value of the key which was pressed to terminate the data entry field. The value is one of those found in the function key reference below. If the last character input has not yet been read by short CursesRecv, the value of terminator is zero.

source Set to either K if the data was keyed on the terminal , or S if the data was scanned by the bar code scanner.

Symbology Receives the symbology of the bar code scanned. (Not yet implemented.)

Return Values:

Name	Description	Return
CRF_OK	Received data from a terminal.	0
CRF_CONNECT	New terminal connected.	4
CRF_TIMEOUT	Call timed out.	2
CRF_ERR	Error occurred.	1
CRF_PARAM	One or more parameters are invalid. Invalid terminal ID perhaps specified for *type.*	3
CRF_CLOSE	The connection to the terminal was closed.	5

OLE_HANDLE GetRecvHandle(short term)

Returns a standard Windows NT/2000 event handle, which is signaled when data is received for the specified RF terminal or the connection to the terminal is closed. This handle may used in any Win32 WaitFor… functions. When the event handle becomes signaled, the application should immediately call CursesRecv(), specifying the same terminal as GetRecvHandle() and with a timeout of 0. The return value is zero if the term specified is invalid.

short GetWaitHandles(variant phandles, long pcount)

Returns an array of standard Windows NT/2000 event handles, which are signaled when a terminal connects to the control, when data is received for the specified RF terminal through

the control, or when a terminal connection is closed. These handles can be used in the Win32 WaitForMultipleObjects function. When an event handle becomes signaled, the application should call CursesRecv with a type of RECV_ANY and with a timeout of 0. The size of the array must be specified in the variable pointed to by pcount, and the actual number of handles will be returned in the variable pointed to by pcount after the call completes. The application should check to see that the original count value is greater than or equal to the returned value, otherwise, not all handles could be fit into the array. The return values are as follows.

Name	Description	Return
CRF_OK	The call completed successfully	0
CRF_PARAM	One or more parameters are invalid	3

Attributes

The following video attributes can be passed to the routines attron(),attroff(), and attrset(), or ORed with the characters passed to addch().

Name	Key Name	Value (hex)
A_REVERSE	Reverse video	0100
A_BLINK	Blinking	0200
A_NORMAL	Bit mask to reset all attributes off	0000
A_STANDOUT	Terminal's best highlighting mode	0100
A_BOLD	Extra bright or bold	0400
A_DIM	Half bright	4000
A_UNDERLINE	Underlining	1000
A_ALTCHARSET	Alternate character set (*short addch()* only)	2000

Function keys

The following function keys can be returned by *CursesRecv()* as the terminator. Note that not all may be supported on a particular terminal if the terminal does not transmit a unique code when the key is pressed.

Name	Value (octal)	Value (decimal)	Key name
KEY_ENTER	0015	13	Enter or send
KEY_BACKSPACE	0010	08	Backspace (unreliable)
KEY_TAB	0011	09	Tab character
KEY_ESCAPE	0033	27	Escape key
KEY_DEL	0177	127	Delete key
KEY_DOWN	0402	258	Down arrow key
KEY_UP	0403	259	Up arrow key
KEY_LEFT	0404	260	Left arrow key
KEY_RIGHT	0405	261	Right arrow key
KEY_HOME	0406	262	Home key (up + left arrow)
KEY_F0	0410	264	Function keys. Space for 20 keys is reserved.
KEY_F(n)	(KEY_F0+(n))		Formula for *fn*

KEY_IC	0513	331	Insert char or enter insert mode.
KEY_BTAB	0541	353	Back tab key
KEY_FIND	0552	362	Find key
KEY_SELECT	0631	409	
KEY_REMOVE	0632	410	

Line Graphics

The following variables may be used to add line-drawing characters to the screen with *addch()*. When defined for the terminal, the variable has the A_ALTCHARSET bit turned on. Otherwise, the default character listed below is stored in the variable. The names were chosen to be consistent with the DEC VT100 nomenclature.

Name	Default	Glyph Description	Hex Values
ACS_ULCORNER	+	upper left corner	206c
ACS_LLCORNER	+	lower left corner	206d
ACS_URCORNER	+	upper right corner	206b
ACS_LRCORNER	+	lower right corner	206a
ACS_RTEE	+	right tee (-|)	2074
ACS_LTEE	+	left tee (|)	2075
ACS_BTEE	+	bottom tee (|)	2077
ACS_TTEE	+	top tee (|)	2076
ACS_HLINE	-	horizontal line	2071
ACS_VLINE	|	vertical line	2078
ACS_PLUS	+	plus	2067
ACS_DIAMOND	+	diamond	2060
ACS_CKBOARD	:	checker board (stipple)	2061
ACS_DEGREE	'	degree symbol	2066
ACS_PLMINUS	#	plus/minus	2067
ACS_BULLET	o	bullet	207e
ACS_BOARD	#	board of squares	207f
ACS_BLOCK	#	solid square block	2061

Chapter 9 • Windows NT/2000 DLL Interface

Description

This document presents details about the Application Program Interface (API) supplied by Connect, Inc. for interfacing to Connect's Windows NT/2000 based Wireless Network Server, OpenAir 400.

The application RFCurses is designed to support an application environment in which one application controls all RF terminals, or in which there is a separate copy of the application program for each terminal.

This particular API is based on the UNIX curses standard that allows applications to control UNIX terminals in a generic fashion. In fact, much of this document is derived from the UNIX manual pages for the curses library. Connect has implemented a subset of the standard curses library for communicating with Connect's VTerm RF application.

Terminal application developers wishing to interface with VTerm will use the Connect-supplied RFCurses.dll and vtcomm.dll. These are standard Win32 dynamic link libraries (dlls), and are accompanied by RFCurses.lib—a C language export library for linking the dlls. Applications must link with RFCurses.lib and source files must #include <RFCurses.h>.

The routines prefixed with mv require y and x coordinates to which to move before performing the appropriate action. The mv() routines imply a call to move() before the call to the other routine.

Note: Y always refers to the row (of the window), and x always refers to the column. The upper left corner is always (0,0), not (1,1).

In addition to drawing characters on the screen, video attributes may be included, which cause the characters to be underlined or shown in reverse video on terminals that support such display enhancements. Line drawing characters may be specified to be output. On input, RFCurses is also able to translate arrow and function keys that transmit escape sequences into single values. The video attributes, line drawing characters, and input values use names, are defined in <RFCurses.h>, such as A_REVERSE and KEY_LEFT.

All routines return either the integer CRF_ERR or the integer CRF_OK, unless otherwise noted.

The chtype structure is defined in RFCurses.h. All timeout parameters are integer quantities, which are expressed in milliseconds. The constant INFINITE can be supplied to "wait forever." The value zero can be supplied to return immediately.

See the Windows NT/2000-specific routines section for additions to the curses standard for handling RF terminals in the NT/2000 environment.

Architecture

If all terminals are controlled by one application, which must be running, and must have called crf_Initialize () before a terminal logs on, the application uses the term parameter with most RFCurses functions to specify which terminal is being affected by that call. The application can use multiple threads to facilitate handling multiple terminals.

If multiple copies of the application are used, each copy must handle only one terminal. Connect's VTerm program spawns a copy of the application when a terminal initially logs on and communicates only that terminal's data to that copy of the application.

When the application is started, it should load RFCurses.dll and immediately call crf_Initialize (). Then is must immediately be available to accept a connection from VTerm using either crf_CursesRecv() or crf_WaitForConnection(). The term parameter, used in many RFCurses calls, is always the same value, which is that returned by crf_CursesRecv() or WaitForConnection().

General Routines

The following documents the curses calls implemented for RFCurses.dll:

int crf_Initialize()

The first routine called should always be crf_Initialize (). This initializes all RFCurses data structures.

int crf_CleanUp()

A program should always call crf_CleanUp() before exiting.

int echo(int term, int on)

This routine controls whether characters typed by the user are echoed by input routines as they are typed. Initially input routines are in ECHO mode, so characters typed are echoed. Setting the parameter on to zero turns echo off. Setting it to 1 turns echo on. (Not currently implemented.)

Output Routines

The following routines are used to manipulate text in windows.

int crf_addch(int term, chtype ch)
int crf_mvaddch(int term, int y, int x, chtype ch)

The character ch is put into the window at the current cursor position of the window and the position of the window cursor is advanced. At the right margin, an automatic newline is performed. At the bottom of the scrolling region the scrolling region is scrolled up one line.

If ch is a tab, newline, or backspace, the cursor is moved appropriately within the window. A newline also does a crf_clrtoeol() before moving. Tabs are considered to be at every fourth column. If ch is another control character, it is drawn in the ^X notation (ctrl-X notation).

Video attributes can be combined with a character by ORing them into the parameter. This results in these attributes also being set. These attributes are combined with any attribute settings based on previous crf_attroff, crf_attron, crf_attrset, crf_standend, and crf_standout calls. See crf_standout(), etc., below.

Note that ch is actually of type chtype, not a character.

int crf_addstr(int term, char * str)
int crf_mvaddstr(int term, int y, int x, char * str)

These routines write all the characters of the null-terminated character string *str* on the given window. This is equivalent to calling *crf_addch()* once for each character in the string. At the right margin, an automatic newline is performed. At the bottom of the scrolling region the scrolling region is scrolled up one line.

int crf_attroff(int term, int attrs)
int crf_attron(int term, int attrs)
int crf_attrset(int term, int attrs)
int crf_standend(int term)
int crf_standout(int term)

These routines manipulate the current attributes of the named window. These attributes can be any combination of the constants A_STANDOUT, A_REVERSE, A_BOLD, A_DIM, A_BLINK, and A_UNDERLINE. These attributes are defined in <RFCurses.h> and can be combined with the C logical or (|) operator.

The current attributes of a window are applied to all characters that are written into the window with crf_addch() or crf_addstr(). Attributes are a property of the character, and move with the character through any scrolling and insert/delete line/character operations. To the extent possible on a particular terminal, they are displayed as the graphic rendition of the characters put on the screen.

crf_attrset(term,attrs) sets the current attributes of the window to attrs. crf_attroff(term,attrs) turns off the named attributes without turning on or off any other attributes. crf_attron(term,attrs) turns on the named attributes without affecting any others. standout(term) is the same as crf_attron(term,A_STANDOUT). crf_standend(term) is the same as crf_attrset(term,0), that is, it turns off all attributes.

Note that attrs is actually of type chtype, not a character.

int crf_beep(int term)

This routine is used to signal the terminal user. *beep()* sounds the audible alarm on the terminal.

int crf_erase(int term)

These routines copy the character *blank* to every position in the window.

int crf_clrtobot(int term)

All lines below the cursor in this window are erased. In addition, the current line to the right of the cursor, inclusive, is erased.

int crf_clrtoeol(int term)

The current line to the right of the cursor, inclusive, is erased.

int crf_delch(int term)
int crf_mvdelch(int term, int y, int x)

The character under the cursor in the window is deleted. All characters to the right on the same line are moved to the left one position and the last character on the line is filled with a blank. The cursor position does not change (after moving to (y,x), if specified).

int crf_deleteln(int term)

The line the cursor is on in the window is deleted. All lines below the current line are moved up one line. The bottom line of the window is cleared. The cursor position does not change.

int crf_insch(int term, chtype ch)
int crf_mvinsch(int term, int y, int x, chtype ch)

The character *ch* is inserted before the character under the cursor. All characters to the right are moved one space to the right, losing the rightmost character of the line. The cursor position does not change (after moving to (y, x), if specified).

Note that *ch* is actually of type *chtype*, not a character.

int crf_insertln(int term)

A blank line is inserted above the current line, and the bottom line is lost.

int crf_move(int term, int y, int x)

The cursor associated with the window is moved to line (row) y, column x. The position specified is relative to the upper left corner of the window, which is (0, 0).

int crf_printw(int term, char * fmt [, arg...])
int crf_mvprintw(int term, int y, int x, char * fmt [, arg...])

These routines are analogous to *printf(S)*. The string, which would be output by printf(S), is instead an output using crf_addstr*()*.

Extended Routines

int crf_print_mode(int term, int on)

Turns on or off printing to a printer attached to the terminal. If on is 1, any subsequent calls to crf_addch(),crf_addstr(), or crf_printw() sends the data specified to the printer instead of the terminal display until crf_print_mode is called again for that terminal with on set to 0.

int crf_cursor_mode(int term, int on)

Controls whether or not the cursor is visible on the terminal. Calling crf_cursor_mode with on set to 1 causes the cursor to be displayed. Calling crf_cursor_mode with on set to zero causes the cursor to be hidden.

int crf_barcode(int term, int code, int status, int minlen, int maxlen, int depend)

#defines…

DC_ALLOFF	DC_PDF_417
DC_CODABAR	DC_SUPPS
DC_CODE_11	DC_UPC_A
DC_CODE_128	DC_UPC_E0
DC_CODE_39	DC_UPC_E1
DC_CODE_49	DC_DISABLE
DC_CODE_93	DC_ENABLE
DC_CODE_D25	DC_CODE_I25
DC_EAN_13	
DC_EAN_8	
DC_MSI	

int crf_setbeep(int term, int beep_dur, int beep_frq, int sbeep_dur, int sbeep_frq)

Sets the frequency and duration of beeps on the terminal for application generated and scanner generated tones.

beep dur	Duration in milliseconds of a standard application beep generated with the crf_beep() function.
beep_frq	Frequency in hertz of a standard application beep generated with the crf_beep() function.
sbeep_dur	Duration in milliseconds of a scan decode beep.
sbeep_frq	Frequency in hertz of a scan decode beep.

int crf_prompt(int term, int y, int x, int length, unsigned long attrib)

Displays an input prompt on the terminal for data entry using the terminal keyboard or barcode scanner. The prompt starts at the coordinates specified by *y,x,* and will have a maximum length of *length* characters. Functionality of the prompt can be altered by setting prompt attributes using the *attrib* parameter. The following prompt attributes are defined and multiple attributes can be set by ORing them together.

Input Characters

PROMPT_ALPHA	Only alphabetic characters can be entered.
PROMPT_NUMERIC	Only numeric characters can be entered.
PROMPT_ALPHANUM	Only alphabetic and numeric characters can be entered.
PROMPT_ANYCHAR	Any characters can be entered (default).

Input Device

PROMPT_KEY	Data can only be entered using the keyboard.
PROMPT_SCAN	Data can only be entered using the scanner.
PROMPT_KEYORSCAN	Data is initially accepted from either the keyboard or scanner. Scanning is not allowed on a partially keyed field. If the operator clears the field, then scanning is again allowed.
PROMPT_KEYANDSCAN	Data is initially accepted from either the keyboard or scanner. Scanning is allowed on a partially keyed field. The keyed data is discarded and the scanned data is accepted. (default)

Other

PROMPT_HIDDEN	Data typed will not be displayed on the terminal screen. This is typically used for a password prompt.
PROMPT_NOLINE	By default underscores are displayed to indicate the position of the prompt. The attribute turns off the display of the underscores.
PROMPT_AUTOENTER	This attribute causes data to be transmitted from the terminal as soon as the prompt is filled, without the user having to press the enter key.

int crf_refresh(int term)

Causes all current RFCurses commands to be transmitted immediately to the terminal. The VTerm terminal handler will automatically send commands to the terminal after a short inactivity timeout, but this command causes VTerm not to wait for the timeout, but to go ahead and send the commands right away. This function is optional, but can result in slightly better system performance.

int crf_fontsize(int term, int doublehigh, int doublewide)

Sets the terminal screen to display in double high and/or double wide character mode. Calling crf_fontsize with the parameters set to 1 turns on double high and/or double wide. Calling crf_fontsize with the parameters set to 0 turns off double high and/or double wide. An application can call crf_query to determine the number of rows and columns currently available on the terminal.

int crf_query(int term, QUERY_BUF * qb)

Receives terminal specific data, including the number of rows and columns on the terminal screen, the number of keys on the terminal keyboard, the type of terminal, and the version of the terminal software. The parameter i is a pointer to a structure of type QUERY_BUF which is defined in **vtdefs.h** as follows:

```
typedef struct
{
        char termtype[5];
        char termver[6];
        char termkeys[4];
        int rows;
        int cols;

} QUERY_BUF;
```

termtype	The type of terminal in use.
termver	The version of the terminal software.
termkeys	The number of keys on the terminal keyboard.
rows	The number of rows on the terminal display.
cold	The number of columns on the terminal display.

Windows NT/2000-Specific Routines

int crf_WaitForConnection(int * pterm, int timeout)

Causes the program to wait the specified timeout period for an RF terminal to connect to the RFCurses dll. A connection occurs whenever a terminal reboots and runs the VTERM application. The terminal identifier will be stored in the variable pointed to be pterm. This value should be used for all future RFCurses calls for this terminal.

Return Values:

CRF_OK	A terminal connected to the dll.
CRF_TIMEOUT	Call timed out.
CRF_ERR	Error occurred.
CRF_PARAM	One or more parameters are invalid.

HANDLE crf_GetConnectWaitHandle()

Returns a standard Windows NT/2000 event handle, which is signaled when a terminal connects to the RFCurses **dll**. This handle can be used in any Win32 WaitFor… functions. When the event handle becomes signaled, the application should immediately call crf_WaitForConnection(&term, 0).

int crf_Disconnect(int term)

Terminates the connection with the VTerm application associated with the specified terminal.

Return Values:

CRF_OK	Call completed successfully.
CRF_PARAM	One or more parameters are invalid.

int crf_CursesRecv(int type, int * pterm, char * buf, int * len, CRECV_INFO * pinfo,int timeout)

Causes the program to wait the specified timeout period for an RF terminal to connect to the RFCurses dll, or for data to be received for a terminal as specified by the type parameter. type can have one of the following values:

RECV_ANY	Return when any terminal connects or sends data.
RECV_CONNECT	Return when any terminal connects.
RECV_DATA	Return when any terminal sends data or type, can specify a particular terminal ID and returns only when data is received from that terminal.
***pterm**	Receives the value of the terminal which caused the call the return. (i.e., The terminal that connected or sent data).
***len**	Must initially contain the length of the buffer pointed to by buf and is changed to the number of bytes written to buf.
***pinfo**	Address of a CRECV_INFO structure which is filled with details of data received from a terminal prompt. The structure is defined in **vtdefs.h** as

```
typedef struct
{
        int terminator;
        char source;
        char symbology[5];

} CRECV_INFO;
```

terminator	Receives the value of the key which was pressed to terminate the data entry field. The value is one of those found in the function key reference below.

If the last character input has not yet been read by crf_CursesRecv, the value of *pterminator is zero;source - Set to either K if the data was keyed on the terminal, or S if the data was scanned by the barcode scanner.

Symbology	Receives the symbology of the barcode scanned. (Not yet implemented)

Return Values:

CRF_OK	Received data from a terminal.
CRF_CONNECT	New terminal connected.
CRF_TIMEOUT	Call timed out.
CRF_ERR	Error occurred.
CRF_PARAM	One or more parameters are invalid. Could be an invalid terminal ID was specified for *type.*
CRF_CLOSE	The connection to the terminal was closed.

HANDLE crf_GetRecvHandle(int term)

Returns a standard Windows NT/2000 event handle, which is signaled when data is received for the specified RF terminal or the connection to the terminal is closed. This handle can be used in any Win32 WaitFor… functions. When the event handle becomes signaled, the application should immediately call crf_CursesRecv() specifying the same terminal as in crf_GetRecvHandle() and with a timeout of zero (i.e., rf_CursesRecv(term,&term,buf,&len,&info,0)). The return value is zero if the term specified is invalid.

int crf_GetWaitHandles(HANDLE * phandles, int * pcount)

Returns an array of standard Windows NT/2000 event handles, which are signaled when a terminal connects to the RFCurses dll, when data is received for the specified RF terminal through the RFCurses dll, or when a terminal connection is closed. These handles can be used in the Win32 WaitForMultipleObjects function. When one of the event handles becomes signaled, the application should immediately call crf_CursesRecv with a type of RECV_ANY and with a timeout of 0. (i.e., crf_CursesRecv(RECV_ANY,&term,buf,&len,&info,0)). The size of the array must be specified in the variable pointed to by pcount, and the actual number of handles will be returned in the variable pointed to by pcount after the call completes. The application should check to see that the original count value is greater than or equal to the returned value, otherwise, not all handles could be fit into the array.

Return Values:

CRF_OK	The call completed successfully.
CRF_PARAM	One or more parameters are invalid.

Attributes

The following video attributes, defined in <**RFCurses.h**>, can be passed to the routines crf_attron(),crf_attroff(), and crf_attrset(), or ORed with the characters passed to crf_addch().

A_REVERSE	Reverse video.
A_BLINK	Blinking.
A_NORMAL	Bit mask to reset all attributes off.
A_STANDOUT	Terminal's best highlighting mode.
A_BOLD	Extra bright or bold.
A_DIM	Half bright.
A_UNDERLINE	Underlining.
A_ALTCHARSET	Alternate character set (*addch()* only).

Function Keys

The following function keys, defined in <**RFCurses.h**>, can be returned by crf_CursesRecv() as the terminator. Note that not all may be supported on a particular terminal if the terminal does not transmit a unique code when the key is pressed.

Name	Value (octal)	Value (decimal)	Key name
KEY_ENTER	0015	13	Enter or send
KEY_BACKSPACE	0010	8	Backspace (unreliable)
KEY_TAB	0011	9	Tab character
KEY_ESCAPE	0033	27	Escape key
KEY_DEL	0177	127	Delete key
KEY_DOWN	0402	258	Down arrow key
KEY_UP	0403	259	Up arrow key
KEY_LEFT	0404	260	Left arrow key
KEY_RIGHT	0405	261	Right arrow key
KEY_HOME	0406	262	Home key (up + left arrow)
KEY_F0	0410	264	Function keys. Space for 20 keys is reserved.
KEY_F(n)	(KEY_F0+(n))		Formula for *fn*
KEY_IC	0513	331	Insert char or enter insert mode.
KEY_BTAB	0541	353	Back tab key
KEY_FIND	0552	362	Find key
KEY_SELECT	0631	409	
KEY_REMOVE	0632	410	

Line Graphics

The following variables may be used to add line-drawing characters to the screen with *crf_addch()*. When defined for the terminal, the variable has the A_ALTCHARSET bit turned on. Otherwise, the default character listed below is stored in the variable. The names were chosen to be consistent with the DEC VT100 nomenclature.

Name	Default	Glyph Description
ACS_ULCORNER	+	upper left corner
ACS_LLCORNER	+	lower left corner
ACS_URCORNER	+	upper right corner
ACS_LRCORNER	+	lower right corner
ACS_RTEE	+	right tee (-|)
ACS_LTEE	+	left tee (|)
ACS_BTEE	+	bottom tee (|)
ACS_TTEE	+	top tee (|)
ACS_HLINE	-	horizontal line
ACS_VLINE	|	vertical line
ACS_PLUS	+	plus
ACS_DIAMOND	+	diamond
ACS_CKBOARD	:	checker board (stipple)
ACS_DEGREE	'	degree symbol
ACS_PLMINUS	#	plus/minus
ACS_BULLET	o	bullet
ACS_BOARD	#	board of squares
ACS_BLOCK	#	solid square block

Chapter 10 • Ethereal

What is Ethereal?

Every network manager at some time or other needs a tool that can capture packets off the network and analyze them. In the past, such tools were either very expensive, propietary, or both. However, with the advent of Ethereal, all that has changed.

Ethereal is perhaps one the best open source packet sniffers available today.

Platforms Ethereal runs on

Ethereal currently runs on most UNIX platforms and the various Windows platforms. It requires GTK+, GLIB and libpcap in order to run.

Where to get Ethereal

You can get the latest copy of the Ethereal from the Ethereal Website: http://www.ethereal.com. The website allows you to choose from among several mirrors for downloading.

Obtaining the source and binary distributions

You can obtain both source and binary distributions from the Ethereal web site: http://www.ethereal.com. Simply select the download link, and then select either the source package or binary package of your choice from the mirror site closest to you.

Installing Ethereal under Windows

In this section we explore installing Ethereal under Windows from the binary packages. You must follow two steps:

1. Install WinPcap. There are instructions at the WinPcap web site for installing it under Windows 9X, Windows NT and Windows 2000. These are located at: http://netgroup-serv.polito.it/winpcap/install/Default.htm.

2. Install Ethereal. You may acquire a binary installable of Ethereal at http://www.ethereal.com/download.html#binaries. Download the installer (after installing WinPcap) and execute it.

Starting Ethereal

You can start Ethereal from the command line under UNIX, but it can also be started from most Window managers as well.

Ethereal is comprised of three main windows, or panes.

1. The top pane is the packet list pane. It displays a summary of each packet captured. By clicking on packets in this pane your control what is displayed in the other two panes.
2. The middle pane is the tree view pane. It displays the packet selected in the top pane in more detail.
3. The bottom pane is the data view pane. It displays the data from the packet selected in the top pane, and highlights the field selected in the tree view pane.

In addition to the three main panes, there are four elements of interest on the bottom of the Ethereal main window.

A. The lower leftmost button labeled "Filter:" can be clicked to bring up the filter construction dialog.

B. The left middle text box provides an area to enter or edit filter strings. This is also where the current filter in effect it displayed. You can click on the pull down arrow to select past filter string from a list.

C. The right middle button labeled "Reset" clears the current filter.

D. The right text box displays informational messages. These message may indicate whether or not you are capturing, what file you have read into the packet list pane if you are not capturing. If you have selected a protocol field from the tree view pane and it is possible to filter on that field then the filter label for that protocol field will be displayed.

The Ethereal menus

The Ethereal menu sits across the top of the Ethereal window.

It contains the following items:

File

This menu contains menu-items to open and reread capture files, save capture files, print capture files, print packets, and to quit from Ethereal.

Edit

This menu contains menu-items to find a frame and goto a frame,mark one or more frames, set your preferences, create filters, and enable or disable the dissection of protocols (cut, copy, and paste are not presently implemented).

Capture

This menu allows you to start and stop captures.

Display

This menu contains menu-items to modify display options, match selected frames, colorize frames, expand all frames, collapse all frames, show a packet in a separate window, and configure user specified decodes.

Tools

This menu contains menu-items to display loaded plugins, follow a TCP stream, obtain a summary of the packets that have been captured, and display protocol hierarchy statistics.

Help

This menu contains the About Ethereal... menu item and access to some basic Help.

Ethereal preferences

There are a number of preferences you can set from one place. Simply select the Preferences... menu item from the Edit menu, and Ethereal will pop up the Preferences dialog box.

The Ethereal Preferences dialog box is a tabbed dialog box that allows you to set preferences for each of the following elements:

Printing

This tab allows you to define the default printing command that Ethereal will use as well as the default output file name when you print to a file.

Columns

This tab allows you to select which columns appear in the Packet List Pane.

TCP Streams

This tab allows you to change the foreground and background colors used by the **Follow TCP Stream**.

GUI

This tab allows you to configure various characteristics of the GUI.

Other tabs

The remaining tabs allow you to configure various preferences for the dissection of various network protocols.

Files used by Ethereal

Ethereal uses a number of files while it is running. Some of these reside in $HOME/.ethereal and are used to maintain information between runs of Ethereal, while some of them are maintained in system areas.

The following are some of the files accessed by Ethereal:

$HOME/.ethereal/preferences

This file contains all your Ethereal preferences, including defaults for capturing and displaying packets. It is a simple text file containing statements of the form **variable: value**.

$HOME/.ethereal/filters

This file contains all the filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
"<filter name>" <filter string>
```

$HOME/.ethereal/colorfilters

This file contains all the color filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
@<filter name>@<filter string>@[<bg RGB(16-bit)>][<fg RGB(16-bit)>]
```

/usr/share/ethereal/plugins, /usr/local/share/ethereals/plugins, $HOME/.ethereal/plugins

Ethereal searches for plugins in the directories listed above. They are searched in the order listed.

/etc/ethers, $HOME/.ethereal/ethers

When Ethereal is trying to translate Ethernet hardware addresses to names, it consunts the files listed above in the order listed. If an address is not found in /etc/ethers, Etherereal looks in $HOME/.ethereal/etheres

Each line in these files consists of one hardware address and name separated by whitespace. The digits of hardware addressses are spearated by colons (:), dashes (-) or periods(.). The following are some examples:

```
ff-ff-ff-ff-ff-ff       Broadcast
c0-00-ff-ff-ff-ff       TR_broadcast
00.2b.08.93.4b.a1       Freds_machine
```

/usr/local/etc/manuf

Ethereal uses the file listed above to translate the first three bytes of an Ethernet address into a manufacturers name. This file has the same format as the ethers file, except addresses are three bytes long.

$HOME/.ethereal/ipxnets

Ethereal uses the above file to translate IPX network numbers into names.

An example is:

```
C0.A8.2C.00      HR
c0-a8-1c-00      CEO
00:00:BE:EF      IT_Server1
110f             FileServer3
```

The Ethereal section above is comprised of excerpts from **Ethereal User's Guide: V1.1 for Ethereal 0.9.7** to which the following license applies.

The GNU Free Document Public License

Copyright

Version 1.1, March 2000

This page is intentionally blank.

Chapter 11 • TightVNC

VNC (Virtual Network Computing) is a client/server software package allowing remote network access to graphical desktops. With VNC, you can access your Internet-connected machine from anywhere. VNC is free (released under the GNU General Public License) and available on most platforms.

An enhanced version of VNC, TightVNC, contains new features, improvements, optimizations and bugfixes. TightVNC is compatible with standard VNC.

TightVNC is used to perform remote control and administration tasks in Windows, Unix and mixed network environments. It is helpful in distance learning and remote customer support.

The following are instructions on installation, remote upgrade, running the server and viewer, and uninstalling.

Installation

TightVNC is available in the self-installing form, starting from its 1.2.1 release. Run the executable to install. The installation wizard will allow you to choose an installation directory and a name for the TightVNC group under the **Start->Programs** menu. By default, TightVNC installs into the Program Files\TightVNC directory, but you may choose any other location during installation.

Upgrading Remotely

TightVNC servers can be upgraded remotely, starting from its 1.2.5 version, meaning that the TightVNC installation can be performed in an active TightVNC session. You cannot replace the executable files in place while the TightVNC service is running, so the installer will copy the new files to a temporary location, and these new files will replace the older versions during the next reboot. The installer prompts for reboot if unable to replace the executables.

Reboot the computer before using this feature. If you want to access your computer after the reboot, run WinVNC as a service, not in the application mode.

Note: There is no warranty of absolute reliability of the remote upgrade procedure. Close all running applications (besides the WinVNC service) before launching the TightVNC installer to minimize risks.

Running the Server and Viewer

Like normal VNC, TightVNC is comprised of the server (WinVNC), which shares the screen of the machine on which it's running, and the viewer, which shows the remote screen received from the server. To get started, run a server on the machine to be accessed remotely and connect to it with a viewer. TightVNC Win32 distribution includes both the server and viewer parts.

Running a Server (WinVNC)

WinVNC can run in the application mode and as a Windows service. In the application mode, the server runs only during the current user session and closes on logout. To start WinVNC in the application mode, choose **Start ->Programs->TightVNC->Launch TightVNC Server**.

Right-click on the tray icon to bring up a menu with the following options.

- **Properties** - displays the Properties dialog, allowing the user to change WinVNC parameters.
- **Add New Client** - allows outgoing connections from the server to any viewer started in "listening" mode.
- **Kill All Clients** - disconnects all currently connected clients from the server.
- **Disable New Clients** - disables new client connection to the server.
- **About WinVNC** - shows the "About..." box.
- **Close** – shuts down the server.

Running a Viewer

To view and control a remote desktop on which a TightVNC server is running, run the TightVNC viewer.

Choose one of the following under **Start->Programs->TightVNC**:

- **TightVNC Viewer** - for a slow network connection to the server (best compression)
- **TightVNC Viewer** - for high-speed networks (fast compression)
- **TightVNC Viewer** - starts the viewer in Listen Mode.

After starting the viewer, enter the host name and optional display number of the remote server you want to access at the prompt.

Note: The TightVNC server displays the IP address as the mouse passes over its tray icon.

Uninstalling

Uninstall TightVNC using Add/Remove Programs under Control Panel. You may also remove the directory into which you have installed it (e.g. C:\Program Files\TightVNC).

Note: The TightVNC installation program does not copy files into the system directory. Before uninstalling, check that WinVNC is not running and not installed as a service.

Reference: http://www.tightvnc.com/winst.html