LAN Interface User's Guide Supplement

Agilent Technologies 8712ET/ES and 8714ET/ES RF Network Analyzers

Part No. 08714-90013

Printed in USA

Print Date: June 2000

Supersedes October 1999

Notice

The information contained in this document is subject to change without notice. Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Key Conventions

This manual uses the following conventions:

FRONT PANEL KEY: This represents a key physically located on the analyzer (a “hardkey”).

Softkey: This indicates a “softkey”— a key whose label is determined by the instrument’s firmware, and is displayed on the right side of the instrument’s screen next to the eight unlabeled keys.

Firmware Revision

This manual documents analyzers with firmware revisions E.06.00 and later.

Acknowledgments

Excel™ is a product of Microsoft® Corporation.

Lotus® 1-2-3®, and Lotus Amipro® are U.S. registered trademarks of Lotus Development Corporation.

Microsoft Excel® and Microsoft Word® are U.S. registered trademarks of Microsoft Corporation.

QuickBasic™ is a product of Microsoft Corporation.

Windows® is a registered trademark of Microsoft Corporation.

Portions of the TCP/IP software are copyright Phil Karn, KA9Q.

GIF output routines are by John Silva (derived from Jef Poskanzer's PBMplus package).

Java™ is a U.S. trademark of Sun Microsystems, Incorporated.

Lotus® 1-2-3® are U.S. registered trademarks of Lotus Development Corporation.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

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

MS Windows®, Windows®, Windows 95®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

Netscape® is a U.S. registered trademark of Netscape Communications Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

Postscript™ is a trademark of Adobe Systems Incorporated which may be registered in certain jurisdictions.

Reflection™ is a U.S. trademark of Walker, Richer & Quinn, Incorporated.

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

Portions of the software include source code from the Info–ZIP group. This code is freely available on the Internet by anonymous ftp asftp.uu.net:/pub/archiving/zip/unzip51/.tar.Z, and from CompuServe asunz51.zip in the IBMPRO forum, library 10 (data compression).

Documentation Outline

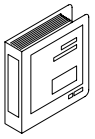
This *User's Guide Supplement* describes how to connect, use and troubleshoot the LAN interface on your analyzer. This supplement contains the following chapters:

1.	**Connecting and Configuring the Analyzer**	Describes how to connect the analyzer to the LAN, and how to configure the analyzer for use on the LAN. Basic user account and file administration is also described. To effectively use this chapter, you should be familiar with your network setup and operation.
2.	**Accessing the Analyzer's Web Pages**	Describes how to use a Web browser to access built-in Web pages.
3.	**Printing**	Describes how to configure and print to a network printer.
4.	**Accessing the Analyzer's File System**	Describes how to access the analyzer's file system using file transfer protocol (FTP). The directory structure of the analyzer is described here.
5.	**Accessing the Analyzer's Dynamic Data Disk**	Describes the analyzer's 'data' directory, the dynamic data disk. Includes an example program.
6.	**Controlling the Analyzer via the LAN**	Shows you methods for programming the analyzer via the network connection.
7.	**Using Network File System (NFS)**	Describes how to configure and use NFS.
8.	**General Troubleshooting**	Describes what to do if you have a problem using the analyzer on your network.
9.	**Quick Reference**	Provides useful information in summary form.
	Glossary	Definitions for networking and other terms used in this book.

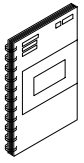
Agilent Technologies 8712ET/ES and 8714ET/ES
Network Analyzer
Documentation Map

The CDROM provides the contents of all of the documents listed below.

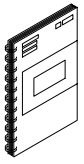
The User's Guide shows how to make measurements, explains commonly-used features, and tells you how to get the most performance from the analyzer.



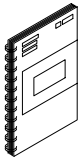
The LAN Interface User's Guide Supplement shows how to use a local area network (LAN) for programming and remote operation of the analyzer.

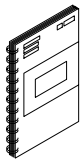


The Automating Measurements User's Guide Supplement provides information on how to configure and control test systems for automation of test processes.

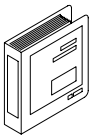


The Programmer's Guide provides programming information including GPIB and SCPI command references, as well as short programming examples.

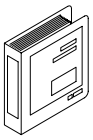




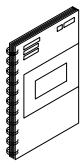
The Example Programs Guide provides a tutorial introduction using BASIC programming examples to demonstrate the remote operation of the analyzer.



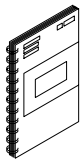
The Service Guide provides the information needed to adjust, troubleshoot, repair, and verify analyzer conformance to published specifications.



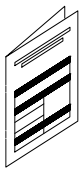
The HP Instrument BASIC User's Handbook describes programming and interfacing techniques using HP Instrument BASIC, and includes a language reference.



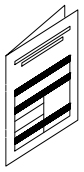
The HP Instrument BASIC User's Handbook Supplement shows how to use HP Instrument BASIC to program the analyzer.



The Option 100 Fault Location and Structural Return Loss Measurements User's Guide Supplement provides theory and measurement examples for making fault location and SRL measurements. (Shipped only with Option 100 analyzers.)



The CATV Quick Start Guide provides abbreviated instructions for testing the quality of coaxial cables. (Shipped only with Option 100 analyzers.)



The Cellular Antenna Quick Start Guide provides abbreviated instructions for verifying the performance of cellular antenna systems. (Shipped only with Option 100 analyzers.)

Contents

Contents

Contents

Contents

Contents

1 Connecting and Configuring the Analyzer

About This Chapter

This chapter describes how to

- connect your analyzer to your network
- set up a network
- configure your analyzer
- verify connectivity
- manage user names and passwords
- configure your analyzer automatically using BOOTP
- run programs automatically using BOOTP

In order to complete the steps in this chapter, you'll need

❏ A computer with a LAN interface, running an operating system that supports TCP/IP, like UNIX® or Microsoft Windows 95®. A typical computer would be an IBM-compatible Pentium®-based PC with a 10Base-T LAN card, or an HP J210 PA-RISC workstation.

❏ A computer program that communicates over the LAN using TCP/IP. This might be an FTP or telnet program, or a program that you write. This will be covered in detail in the following chapters.

❏ LAN cabling, and typically a LAN hub.

If you only wish to print to a LaserJet printer via the LAN, you'll need

❏ an HP LaserJet printer with an HP JetDirect LAN interface card

❏ LAN cabling, and typically, a LAN hub

NOTE

Older versions of Novell Netware used IPX networking protocol exclusively. IPX protocol is not compatible with TCP/IP protocol.

Newer versions of Novell Netware, such as version 3.1x and 4.xx accommodate add-on products which provide a gateway to a TCP/IP network. Consult your Novell network administrator for the latest information on using Novell Netware with TCP/IP protocol.

Intoducing the LAN Interface

With the LAN interface you can

- transfer IBASIC programs between your computer and your analyzer
- transfer files between your computer and your analyzer using file transfer protocol (FTP)
- save files from your analyzer to a computer using network file system (NFS)
- connect many analyzers to one computer
- automate the control of your analyzer
- program the analyzer using SCPI commands
- print hardcopy directly to an HP LaserJet printer
- use your analyzer's Web links to find
  - ✓ general information about the Agilent 87xx family of analyzers
  - ✓ online documentation such as SCPI command references
  - ✓ specific information about your analyzer such as your current firmware revision, installed options, even the analyzer's current screen image
  - ✓ general information about Agilent Technologies, and how to obtain assistance if you need it

LAN Client/Server Functions

Your analyzer acts as either a client or server when you use the client/server features of the analyzer. For example, if you use Network File System (NFS), your analyzer acts as an NFS *client* (see Chapter 7, "Using the Network File System (NFS)," on page 7-1). The table below lists the client/server features of the analyzer, and the function performed by the analyzer when you use each feature:

Client/Server Feature	Analyzer Function
BOOTP	client
FTP	server
NFS	client
SICL LAN	server

Connecting the Analyzer to the LAN

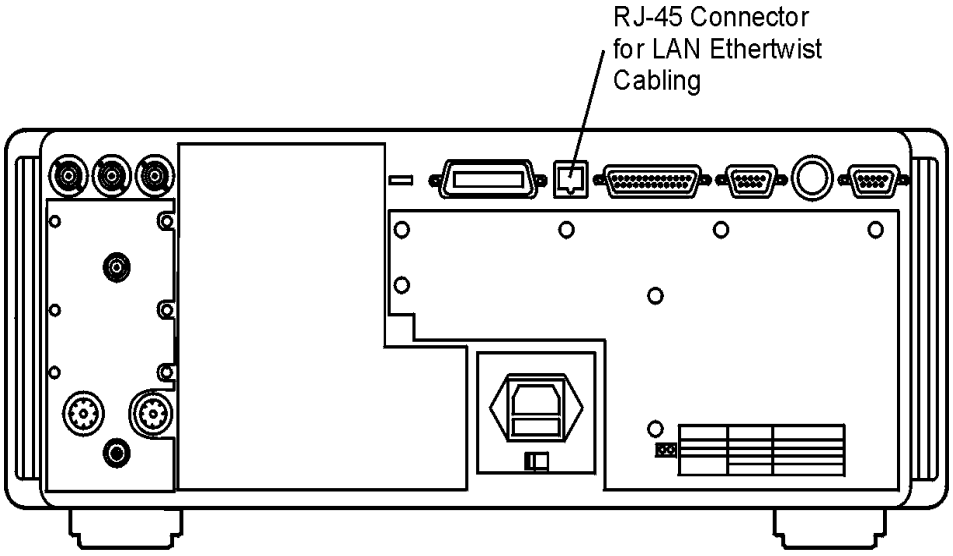
Your analyzer has an RJ-45 connector (see Figure 1-1) and connects to your network using 10Base-T unshielded twisted pair (UTP) cabling, also called Ethertwist. Ethertwist cables resemble standard modular phone cables.

NOTE If your network uses ThinLAN (10Base-2), you will need to purchase an adapter which converts the ThinLAN BNC connector to 10Base-T Ethertwist.

To connect the analyzer to your network:

1. Turn off the analyzer.
2. Connect the Ethertwist cable from your network to the LAN ETHERTWIST port on the rear of your analyzer.
3. Turn on the analyzer.

Figure 1-1 The LAN ETHERTWIST Port



hv61e

Setting Up a Network

If you do not already have a network, you will need to create one. A simple network consists of a central LAN hub with multiple Ethertwist cables, one connected to the LAN port of each network device. This is often called a star topology, with the LAN hub at the center.

- Typical 8-port hub

  HP J2610B AdvanceStack 10Base-T Hub-8U

- Typical 16-port hub

  HP J2611B AdvanceStack 10Base-T Hub-16U

- Typical Ethertwist cables

  92268A twisted-pair “straight-through” cable, 4 meters

  92268B twisted-pair “straight-through” cable, 8 meters

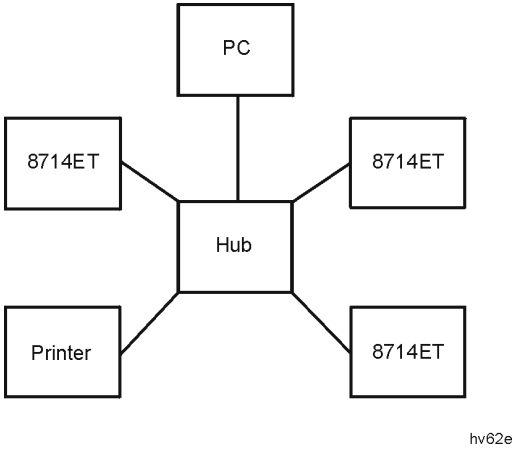
  92268C twisted-pair “straight-through” cable, 16 meters

  92268D twisted-pair “straight-through” cable, 32 meters

  92268N twisted-pair “straight-through” cable, 300 meters

To order cables, contact the nearest Agilent Technologies sales or service office. See Table 9-5 on page 9-11 for a list of sales and service offices.

Figure 1-2 Example of LAN Star Topology



Point-to-Point Connections

It is possible to connect a single computer to a single analyzer, and avoid using a LAN hub. To do this, you must use a special “cross-over” cable or adapter, which acts like a LAN hub. See “EIA/TIA 568B Wiring” on page 9-2 for wiring details. If you try to create a point-to-point connection using a standard “straight-through” cable, it will not work. For most applications, the use of a LAN hub is simpler, and additional devices can be added easily.

NOTE Some commercially-available cross-over cables do not implement the cross-over wiring required for your analyzer. Please refer to “EIA/TIA 568B Wiring” on page 9-2 and verify all connections before using cables not made by Agilent Technologies.

NOTE Point-to-point connections may not work when connecting to older laser printers. Older printers typically require a boot server for network use. For a point-to-point connection with a printer, use an HP LaserJet 4 or newer.

NOTE Point-to-point connections do not require the use of proxy servers, since no server is present in a point-to-point network connection. To use a point-to-point connection, first disable the use of a proxy server in your LAN software. Refer to your software documentation for instructions how to do this.

Configuring the Analyzer

Before you configure your analyzer, you will need to contact your network administrator to obtain the following information:

- ❏ an IP address for the analyzer
- ❏ a host name for the analyzer
- ❏ a gateway IP address
- ❏ a subnet mask

The Analyzer's IP Address and Hostname

Each device on your network must have a unique address so that all devices can communicate simultaneously over the same network. These unique addresses are called IP addresses, and are assigned by your network administrator. An IP address is a set of four decimal numbers, separated by periods, like `192.170.128.21`. In this document, the term "LAN address" refers to the IP address.

CAUTION It is important that no two devices are assigned the same IP address. Both devices may fail to communicate on the network.

You may also receive (or request) from your network administrator a hostname for your analyzer, like `my8712`.

The hostname is not required, but can be used on your computer so that you don't have to remember the IP address. Typically, the hostname is found in the `/etc/hosts` or `control panel/network` file on your computer or is returned by a name server.

Your network administrator will apply for a range of IP addresses from the Internet Network Information Center (InterNIC). InterNIC is responsible for registering domain names and assigning TCP/IP network numbers to networks that connect to the Internet. You may contact InterNIC via e-mail at `hostmaster@internic.net`, or by accessing their Web site at `http://www.networksolutions.com`.

The Gateway Address

If your analyzer will be communicating with devices on different physical networks, you may need to have your network administrator assign a gateway IP address for you. The gateway IP address is the address of a routing device that connects your analyzer's LAN with other LANs. Set the gateway address to `0.0.0.0` if a gateway is not required. See "To Configure the Analyzer" on page 1-10 to set this.

See "Subnets and Gateways" on page 8-15 for more information on gateway addresses.

The Subnet Mask

If your analyzer will be communicating with devices on different physical networks, you may need to have your network administrator assign a subnet mask number for you. The subnet mask tells your analyzer whether a remote device is on the same LAN as your analyzer. If your analyzer is attempting to communicate with another device, the subnet mask defines whether your analyzer needs to route communications through the gateway. Set the subnet mask to `0.0.0.0` if a subnet mask is not required. See "Configuring the Analyzer" on page 1-8to set this.

See "Subnets and Gateways" on page 8-15 for more information on subnet masks.

The Ethernet Address

Your analyzer has a unique built-in Ethernet address associated with the LAN hardware inside it. The Ethernet address is a 48-bit number assigned at the factory. You don't have to know the Ethernet address to configure and use the analyzer, unless you are using the BOOTP feature (see "Using BOOTP" on page 1-15 for details).

To Configure the Analyzer

1. Press SYSTEM OPTIONS **LAN** to access the LAN menu.

NOTE

After each of the following steps, the analyzer will prompt you to cycle power for the new setting to take effect. It is not necessary to cycle the power after each step. It only needs to be done once—when you are finished entering all of the settings.

2. Press **LAN Port Setup** **HP 871xxx IP Address**, and enter the IP address that your network administrator assigned to your analyzer. You may have also received a hostname (for example: `my8712`). You cannot enter the hostname into your analyzer, just the IP address. The hostname can be used on your computer so that you don't have to remember the IP address.
3. Press **Gateway IP Address**, and enter the numbers assigned to you by your network administrator. If you were not assigned a gateway IP address, leave the setting at `0.0.0.0` (default value) to disable gateway routing.
4. Press **Subnet Mask**, and enter the numbers assigned to you by your network administrator. If you were not assigned a subnet mask, leave the setting at `0.0.0.0` (default value) to disable subnet masking.
5. Once you have entered these settings, cycle the power on your analyzer to initialize the LAN interface with these new values.

Testing the LAN Communication

You should now test communication between your computer and your analyzer.

The `ping` utility is typically used to test LAN communication.

Running `Ping` under Windows 95

Enter the following at the command prompt of a DOS window on your computer or workstation:

```
ping <IP address>
```

or

```
ping <hostname>
```

`<IP address>` is the number that was assigned by your network administrator and was entered into your analyzer in "To Configure the Analyzer" on page 1-10. The `<hostname>` is the hostname assigned to your IP address. For example, type:

```
ping  my8712
```

where `my8712` is the `<hostname>`.

The `ping` utility has three common responses. If there is a valid working connection, you should see a response similar to this:

```
Pinging my8712 [15.4.43.5] with 32 bytes of data:

Reply from 15.4.43.5: bytes=32 time=37ms TTL=252
Reply from 15.4.43.5: bytes=32 time=30ms TTL=252
Reply from 15.4.43.5: bytes=32 time=30ms TTL=252
Reply from 15.4.43.5: bytes=32 time=31ms TTL=252
```

If you see a response similar to the following, your connection may have a problem. Refer to "Troubleshooting the Initial Connection" on page 8-3 for troubleshooting help and information.

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

The following response is generally caused by an incorrect subnet mask or IP address. It usually points to a software setting conflict, and does not signify a hardware problem.

```
Host Unreachable.
Host Unreachable.
Host Unreachable.
Host Unreachable.
```

Running Ping under UNIX

The `ping` program is typically found in the `/etc` or `/usr/etc` directory, so you must add the appropriate directory to your path, or type the full path:

```
/etc/ping <IP address> 64 5
```

or

```
/etc/ping <hostname> 64 5
```

This command tells ping to send 5 packets of 64 bytes each.

The output should look similar to this:

```
PING hostname: 64 byte packets
64 bytes from 15.4.43.5: icmp_seq=0. time=8. ms
64 bytes from 15.4.43.5: icmp_seq=1. time=4. ms
64 bytes from 15.4.43.5: icmp_seq=2. time=4. ms
64 bytes from 15.4.43.5: icmp_seq=3. time=3. ms
64 bytes from 15.4.43.5: icmp_seq=4. time=3. ms

 hostname PING Statistics
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 3/4/8
```

If you do not see any output after about 20 seconds, interrupt the ping command using `^C` (hold down the “Ctrl” key, and press “c”). Once you do this, the ping program should provide some statistics on how many packets were sent and received. If the statistics look like

```
 hostname PING Statistics
4 packets transmitted, 0 packets received, 100% packet loss
```

there is a communications problem. Refer to “Troubleshooting the Initial Connection” on page 8-3 for troubleshooting help and information.

Managing User Names and Passwords

Your analyzer implements a limited form of network security using user name and password pairs. Any remote access of the analyzer, including Telnet or FTP access, requires a valid user name and associated password.

A default user name and password pair is set for you prior to shipment:

User Name `network`

Password `analyzer`

NOTE You should change this user name and password if you want to use the security features of the analyzer, since the default user name and password is the same for all new analyzers, and is therefore public.

Constructing Valid User Names and Passwords

A valid user name must have 1 to 40 characters. A valid password must have 8 to 40 characters.

Adding New User Names and Passwords

NOTE You can add up to seven user name/password pairs to the analyzer's access list.

Perform the following steps to add a new user name and password to the access list:

1. Press (SYSTEM OPTIONS) **LAN** **Login User Setup** .
2. Press **Add Login User** .
3. Type the user name in the displayed dialog box.
4. Press **Enter** when you are done.
5. Type the password in the displayed dialog box.
6. Press **Enter** when you are done.

7. Type the password again (to confirm the password) in the displayed dialog box.
8. Press **Enter** when you are done.

If the entries are valid, the new user name and password will be confirmed with the following message:

```
User ... has been added to the list
```

Removing a User from the Access List

Perform the following steps to remove a user from the access list:

1. Press SYSTEM OPTIONS **LAN** **Login User Setup** .
2. Press **Delete Login User** .
3. Type the user name in the dialog box that is displayed.
4. Press **Enter** to confirm your entry.
5. Type the user password in the dialog box.
6. Press **Enter** to confirm your entry.

If the entries are valid you should see a confirmation message displayed on the screen:

```
User ... has been deleted from the list
```

NOTE

If you forget any of the user passwords, you will have to delete all users by pressing SYSTEM OPTIONS **LAN** **Login User Setup** **Delete All Users** and re-enter all user names and passwords.

Displaying the Access List

1. Press SYSTEM OPTIONS **LAN** **Login User Setup** .
2. Press **Display User List** .

A table of the login user names will be displayed on the screen.

Using BOOTP

BOOTP Fundamentals

The Bootstrap Protocol (BOOTP) is a simple and elegant method of automatically distributing network information and software via the LAN. BOOTP is built on the *client-server* model. The BOOTP *client* configures itself using configuration information obtained from a BOOTP *server*. Your analyzer has a built-in BOOTP client. The analyzer can use BOOTP to configure itself automatically, obtaining its network configuration information (IP address, gateway address, and subnet mask) from a central BOOTP server over the network. On power up, the analyzer broadcasts a request to boot from a remote server. If a BOOTP server is available on the LAN listening for BOOTP client requests, it transmits configuration parameters to the analyzer over the network. The analyzer uses those parameters automatically.

BOOTP can also be used to automatically retrieve and execute an IBASIC program at boot time. The boot file is transferred to the analyzer from the BOOTP server using FTP or trivial file transfer protocol (TFTP). If the transfer is successful, the file will be loaded into the analyzer's memory and executed. The boot file can be any valid IBASIC program.

Setting Up the BOOTP Server

To use the BOOTP client in your analyzer, you need a BOOTP server application running on a remote UNIX system or a PC. A BOOTP server, `bootpd` (BOOTP daemon), is an integral part of most UNIX operating systems. You will need to obtain a separate BOOTP server application for your PC. Consult your network administrator for obtaining a BOOTP server application for your PC, and for assistance setting up a BOOTP server.

The following steps are required to use BOOTP:

1. Assure that the analyzer and BOOTP server are not separated by a gateway. Consult your network administrator if you are not sure.
2. Set up a BOOTP server application on a remote host (UNIX system or PC). You will need the following information:

- ❏ The Ethernet address of the analyzer. To find out the Ethernet address of your analyzer, press SYSTEM OPTIONS **LAN** **LAN Port Setup** **Ethernet Address**.
- ❏ An IP address for the analyzer. This address is usually assigned by your network administrator.
- ❏ An optional BOOTP host name and IP address.
- ❏ An optional absolute (fully qualified) path to the boot file, which includes all the directories leading to it. If you want to retrieve an IBASIC boot program from your BOOTP server at boot time and execute it, you must know the absolute path to the boot file. The boot file must be accessible using FTP or TFTP.
- ❏ The LAN gateway address and the subnet mask.

Setting Up the BOOTP Client

Perform the following steps to set up the BOOTP client in your analyzer:

1. Press SYSTEM OPTIONS **LAN** **BOOTP Setup**.
2. Toggle [BOOTP] to ON if needed to enable BOOTP. The softkey label will change to **BOOTP ON off**.
3. Press **FTP** or **TFTP** to select either FTP or TFTP file transfer method. If your remote system requires a user name and password, you must use FTP, since TFTP does not implement any user validation.

   If you select **FTP**

   a. Press **FTP User Name** and enter a valid user name for your remote BOOTP host.

   b. Press **FTP Password** and enter a valid password for your remote BOOTP host.
4. Press **Timeout** and enter a timeout time, in seconds, for BOOTP requests. This value is typically between one and five seconds. The **Timeout** value is the number of seconds that your analyzer will spend transmitting BOOTP requests at boot time. If there is no response to the first BOOTP request, then the analyzer will retransmit

a request. The analyzer will continue to retransmit requests at exponentially increasing time intervals until it receives a response or the **Timeout** value has expired.

5. Press **Optional Boot Host** and enter a hostname or a host IP address if you want BOOTP requests sent to a specific remote host only. Otherwise, the analyzer will *broadcast* a BOOTP request at boot time, and will accept a response from *any* BOOTP server.

   If you do not want to use **Optional Boot Host**, make sure that it contains a null or empty string by pressing **Optional Boot Host** **Clear Entry** **Enter**.

NOTE If you use **Optional Boot Host**, you are also required to set up your analyzer's IP address. Refer to "To Configure the Analyzer" on page 1-10 for details on how to set up your analyzer's IP address.

NOTE A local HOSTS file is required to specify a boot host by *name*. A local HOSTS file is not required to specify a boot host by *IP address*. "Using a Local HOSTS File" on page 7-11 for details about creating and using a local HOSTS file.

6. Press **Optional File Name** and enter an absolute (fully qualified) path to the file to be loaded and executed when the analyzer boots. For example, if your file name is `test23.bas`, and the path to the file is `/server5/users/testeng/prodtests`, enter the following for the absolute (fully-qualified) path name:

   ```
   /server5/users/testeng/prodtests/test23.bas
   ```

NOTE You must use the UNIX-style forward slash (/) to separate names when you enter path names *in the analyzer* (the BOOTP client). You may need to use some other character to separate names when you enter path names in your particular *BOOTP server*.

Normally, your analyzer will obtain this file name from the BOOTP server, but you can override this using **Optional File Name**. If you do not want to use **Optional File Name**, make sure that it contains a null string. To do that, press **Optional File Name** **Clear Entry** **Enter**

NOTE

You can set up your BOOTP server to select the file to download when the analyzer boots. Consult your BOOTP server documentation or your system administrator about setting up a BOOTPTAB file on the BOOTP server.

Testing BOOTP

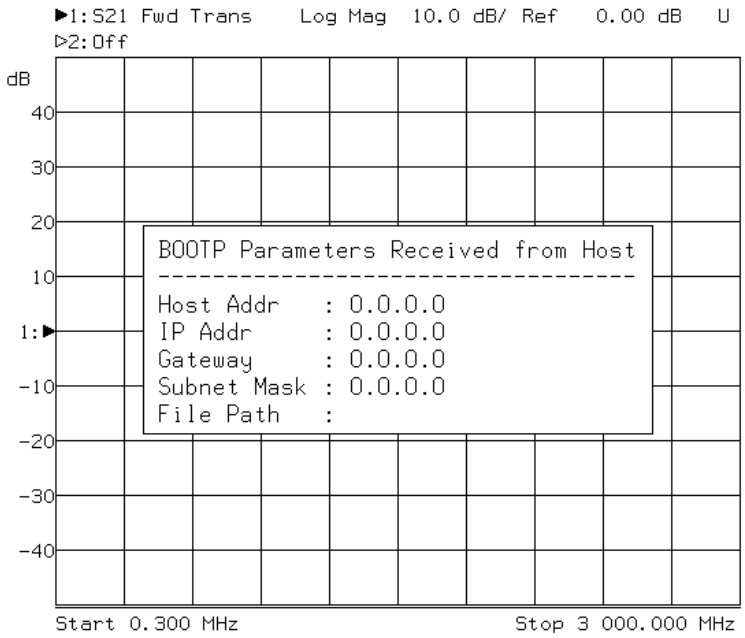
After your analyzer has been set up correctly as a BOOTP client, make sure that the BOOTP server is also set up correctly and is running. Consult your network administrator if you need help doing this.

Perform the steps below to verify that BOOTP works correctly:

1. Press SYSTEM OPTIONS **LAN** **BOOTP Setup** **Parameters Received**.

   The following dialog box will appear:

Figure 1-3 BOOTP Setup Dialog Box



The dialog box shown above displays the following information:

Host Addr	the host address of the BOOTP server
IP Addr	the analyzer IP address set by BOOTP process
Gateway	the analyzer gateway IP address set by the BOOTP process
Subnet Mask	the subnet mask set for the analyzer by the BOOTP process
File Path	the absolute (fully-qualified) path name received from the BOOTP server, or the [Optional File Path] if set

Step one shows the network parameters received from the BOOTP server. To verify that your IBASIC boot file is working correctly, perform steps two through four.

2. Clear your current network configuration information.

   a. Press SYSTEM OPTIONS **LAN** **LAN Port Setup** **871xxx IP Address** **Clear Entry** **Enter**.

   b. Press SYSTEM OPTIONS **LAN** **LAN Port Setup** **Gateway IP Address** **Clear Entry** **Enter**.

   c. Press SYSTEM OPTIONS **LAN** **LAN Port Setup** **Subnet Mask** **Clear Entry** **Enter**.

3. Create an IBASIC file that, when run, will clearly indicate the successful retrieval and execution of the file. Store it on the BOOTP server. Here's an example:

```
10 ASSIGN @Hp8712 TO 800
20 OUTPUT @Hp8712;"DISP:ANN:TITL ON"
30 OUTPUT @Hp8712;"DISP:ANN:CLOC:MODE OFF"
40 OUTPUT @Hp8712;"DISP:ANN:TITL1:DATA 'BOOTP is here!!!'"
50 END
```

4. Cycle power to your analyzer. After your analyzer boots, the network IP address, gateway address and subnet mask should be those values provided by the BOOTP server.

   If you have a boot file set up correctly, your analyzer should also retrieve and execute your boot file. If you used the example program above, the screen will display

   ```
   BOOTP is here!!!
   ```

Setting Up LAN Features with Wizards

IBasic LAN Wizard

An IBasic LAN wizard program is included with the analyzer to assist users in setting up the LAN features. It is located on the Example Program Disk, part number 08714-10003, under the name `lan_wiz`.

Windows LAN Wizard

A Windows-based LAN wizard program is also included with the analyzer to assist users in setting up the LAN features. It is located on the Example Program Disk, part number 08714-10003, under the name `wiz871x.exe`.

It is necessary to set up the analyzer's IP address and SCPI socket port number before using this program.

NOTE

Example programs for the analyzer can be found in the following two locations:

- Example Programs Disk, 8712ET/ES and 8714ET/ES (DOS format): part number 08714-10003
- Web site http://www.agilent.com. Use the search function to find Web pages related to 8712 and 8714 example programs and wizard programs.

2 Accessing the Analyzer's Web Pages

This page left intentionally blank.

About This Chapter

Your analyzer has built-in web pages that are accessible with a web browser such as Netscape Navigator or Microsoft® Internet Explorer. These web pages contain links to general product information, selected on-line documentation, benchmarks, information about your analyzer, and a list of Agilent Technologies offices. You can also e-mail us with your comments and feedback on the Agilent Technologies 87xx family of analyzers.

Before you can access your analyzer with a web browser, you need to connect and configure your analyzer as described in Chapter 1, "Connecting and Configuring the Analyzer."

If your analyzer is directly connected to a PC, without the use of a hub or a larger network, then you probably need to disable the proxy server in the browser. This is because most web browsers are configured to use proxy servers for accessing web pages. If your analyzer is directly connected to your computer, your computer cannot find the proxy server.

If you are communicating to your analyzer over a LAN, then the proxy setting can be left as it is.

Accessing the Analyzer with Your Web Browser

To access your analyzer, start your web browser and connect to `http://<hostname>`, where `<hostname>` is the hostname that has been assigned to the IP address of your analyzer. If you are making a connection to the analyzer without using a domain name system (DNS) server, you can use `http://<IP address>`, where `<IP address>` is the IP address of your analyzer. You can also use the IP address form when using a DNS server.

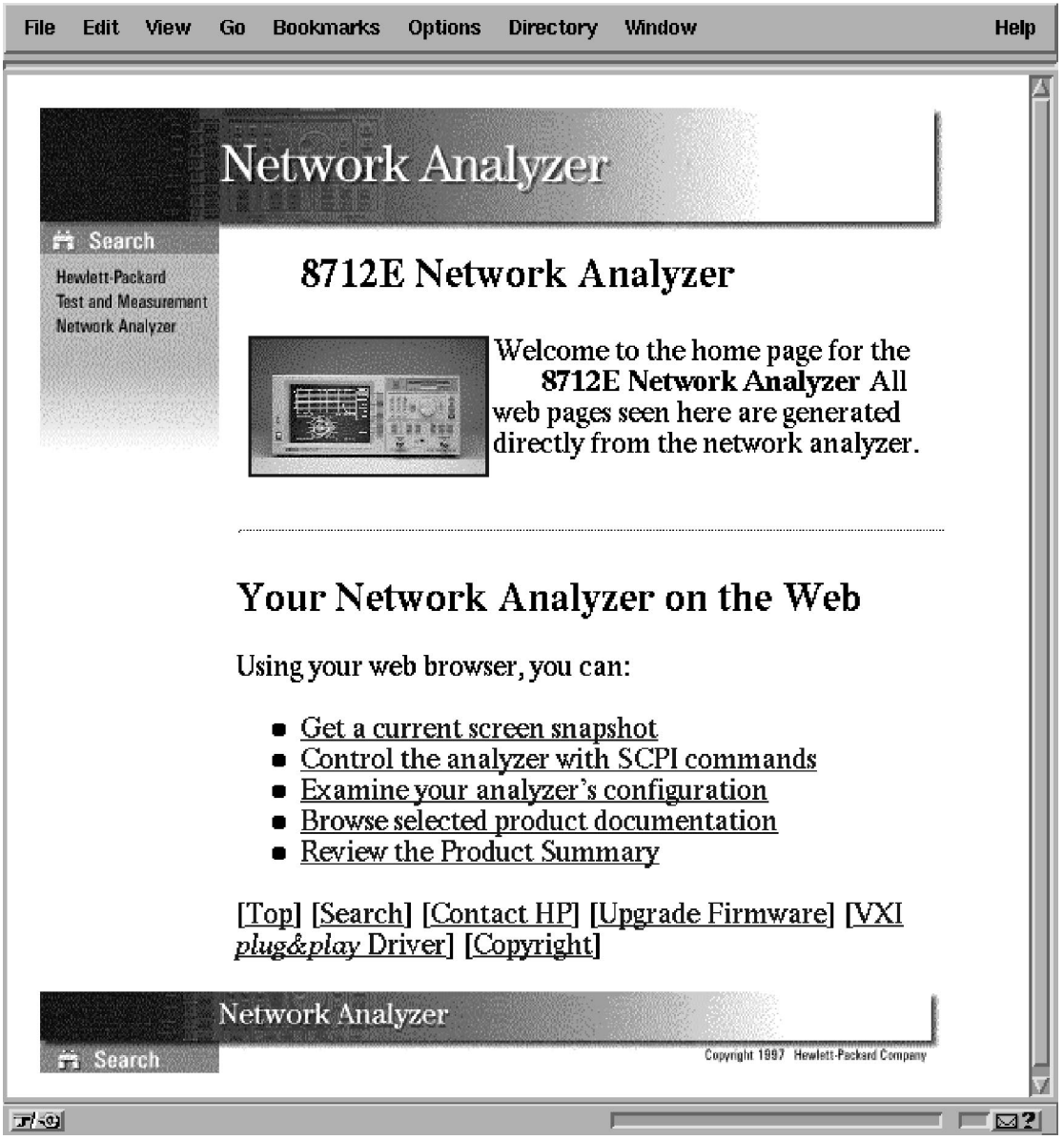
When you are connected to your analyzer, a web page will appear with the following information links:

- Get a current screen snapshot
- Control the Analyzer with SCPI Commands
- Examine your analyzer's configuration
- Browse selected product documentation
- Review the Product Summary
- Other links

Click on the hyperlinks (any underlined words) to browse through the analyzer's pages. See Figure 2-1.

The rest of this chapter explains some of the areas you can browse in further detail.

Figure 2-1 Analyzer Web Page



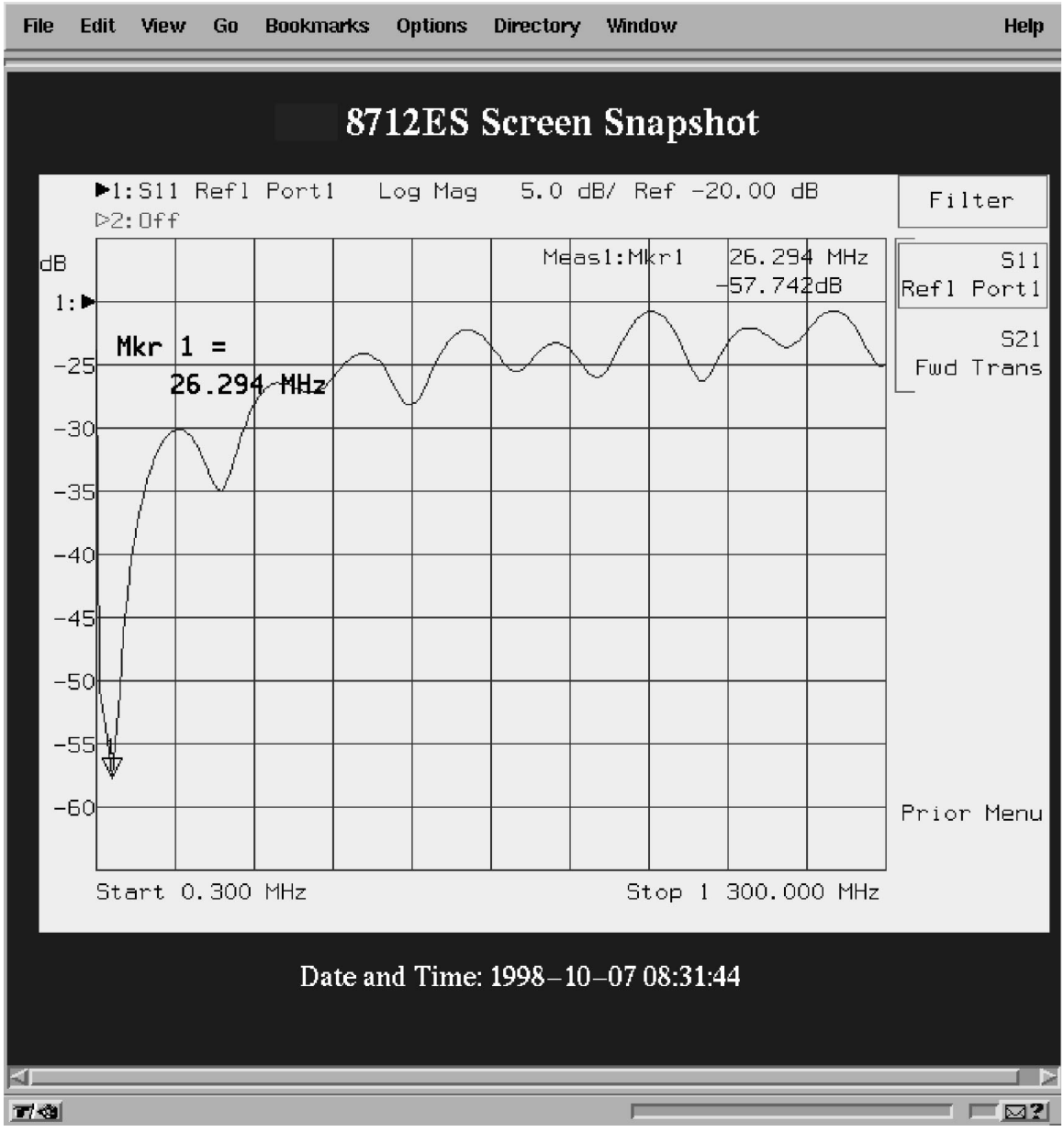
Screen Snapshot

Clicking on Get a current screen snapshot shows an exact copy of your analyzer's current screen image. Use your web browser's "reload" or "refresh" function to get the most current screen image.

CAUTION The screen image takes a few seconds to load. Do not push any buttons on the analyzer or send any programming commands to it while the snapshot is loading, or an inaccurate image may result.

NOTE Before capturing the screen image with your web browser, you may wish to customize the look of the image using the **Color Options** menu on your analyzer. (See your analyzer's *User's Guide* for more information.) In particular, you may want to choose **Inverse Video** to create a white background, especially if you plan to print the page from your web browser. See Figure 2-2.

Figure 2-2 Screen Snapshot



Control the Analyzer with SCPI Commands

Clicking on Control the Analyzer with SCPI Commands launches a Java applet. This applet creates a command-entry dialog box. You can control your analyzer over the LAN by entering SCPI commands in this dialog box. See Figure 2-3. Commands or queries are sent to the analyzer by entering the SCPI mnemonic in the SCPI Command: area, and the response from the analyzer is displayed in the Response Messages: area.

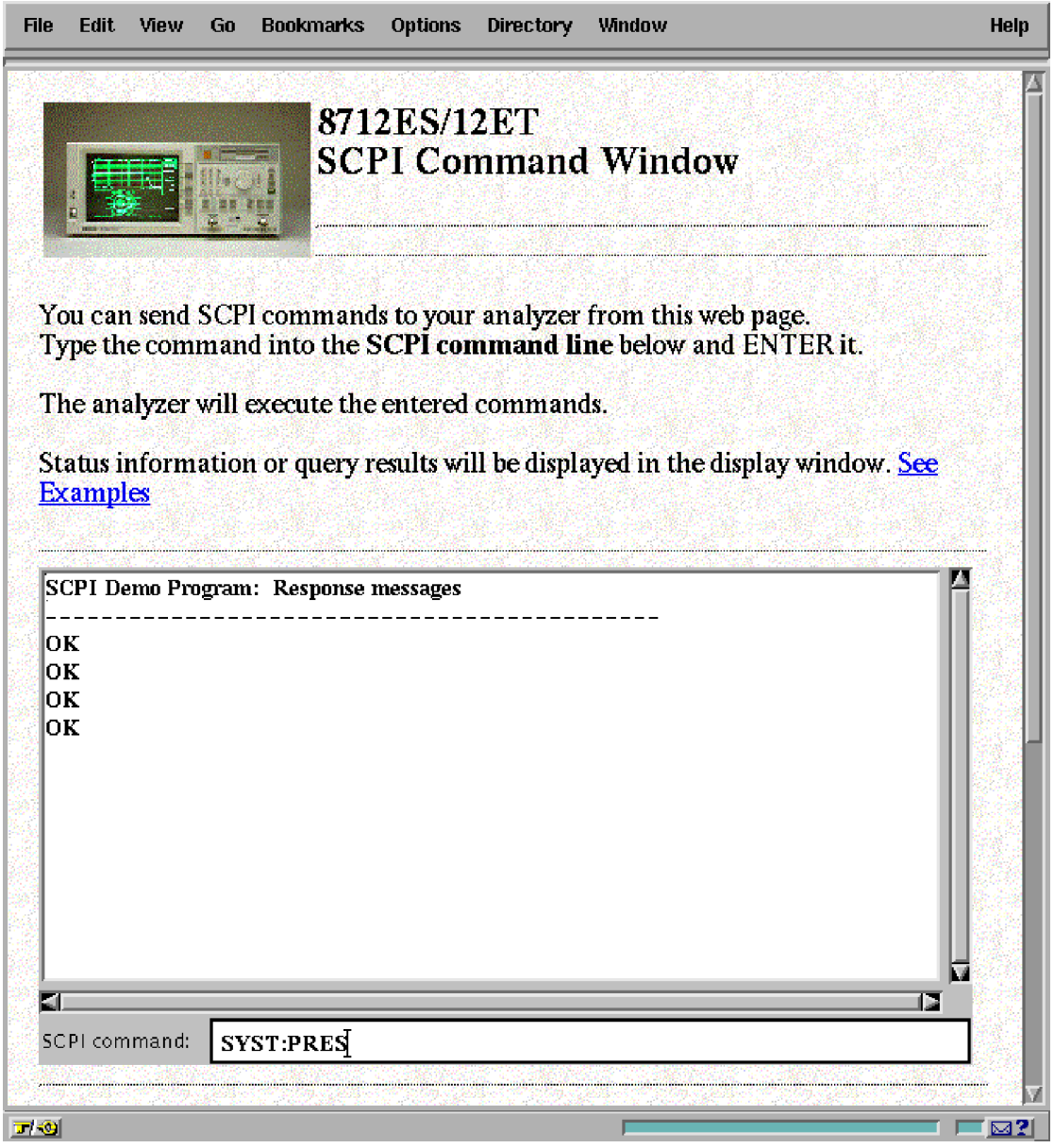
Example commands are provided on this web page as well as a link to the SCPI command reference.

It may be helpful to save frequently-sent SCPI command strings in a separate ASCII file. These commands can then be cut and pasted from the ASCII file into the command line of the dialog box, and edited, if necessary, before sending them to the analyzer.

NOTE

Java is a powerful, cross-platform programming language developed by Sun Microsystems. See `http://www.javasoft.com` for more details.

Figure 2-3 SCPI Command Screen



Analyzer Configuration

Clicking on Examine your analyzer's configuration brings up a screen of information that is equivalent to pressing SYSTEM OPTIONS **Service** **Instrument Info** on the analyzer. This screen shows the model and serial number of your analyzer, the firmware revision, installed options, and the amount of memory.

Product Documentation

This section provides selected portions of your analyzer's documentation on-line, as well as benchmark information and information about product upgrades and options.

The following list shows the links currently available on this page:

- Optimizing your Measurements
- Accessing Built-in Disks
- Controlling I/O Ports
- Accessing the Analyzer's file system via the LAN
- Accessing the Dynamic Data Disk via the LAN
- Controlling the Analyzer via the LAN
- Agilent Technologies 871xE SCPI command reference
- IEE 488.2 common commands
- Product Upgrades and Options
- Transfer Speeds using GPIB
- Transfer Speeds using LAN
- Printing Speed
- List of printed manuals

If there are additional portions of the analyzer's documentation that you think would be helpful to have on-line, please contact us via e-mail. Click on “Contact Agilent Technologies” and “Send us your feedback!” from your analyzer's web page.

Product Overview

The links in this area provide generic information about the Agilent Technologies 871xE family of analyzers. New features, compatibility issues, and available options are included here.

Other Links

At the bottom of every web page in the analyzer, you'll find the following links:

• Top	takes you to the top of the current page.
• Search	takes you to the “Product Documentation” page.
• Contact Agilent Technologies	takes you to a page that provides links to Agilent Technologies web sites, and gives you the opportunity to provide Agilent with feedback on your analyzer and its documentation.
• Upgrade Firmware	takes you to a page that helps you download firmware from Agilent Technologies websites.
• VXI plug&play Driver	takes you to a page that helps you download free VXI plug&play drivers from Agilent Technologies websites.
• Copyright	takes you to copyright information.

3 Printing

About This Chapter

Your analyzer can print directly to an HP LaserJet printer on your network. In order to print to a LAN printer, your analyzer must be communicating on the network. Refer to Chapter 1, “Connecting and Configuring the Analyzer,” on page 1-1 if you have not yet connected and configured your analyzer.

Compatible Printers

The HP LaserJet 4 and HP LaserJet 5 families of printers are compatible with your analyzer for printing directly via a point-to-point connection or over your network. These newer printers allow you to enter the printer's IP address directly from the analyzer front panel and do not require a boot server computer. Your printer should have a JetDirect LAN card installed.

NOTE Some older printers, such as an HP LaserJet III, do not allow you to enter an IP address from the analyzer front panel. They require a boot server computer on the network that configures (sets) the printer's IP address.

Configuring the Printer

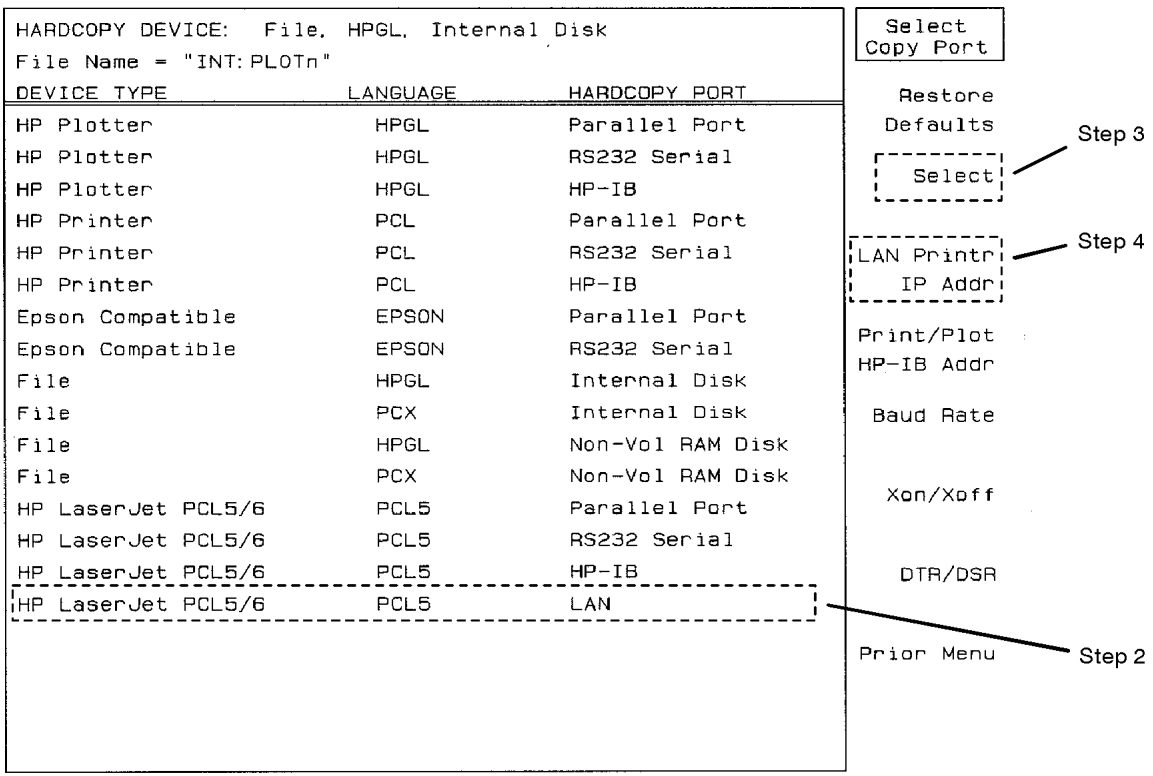
Refer to your printer's documentation for instructions on how to set up your printer for LAN usage. Typically, you will need to contact your network administrator to assign a unique IP address for your printer. Your printer software will configure the printer with the assigned IP address each time it is turned on.

Configuring the Analyzer for Printing to a LAN Printer

To set up your analyzer to print to a LAN printer:

1. Press HARDCOPY **Select Copy Port** .
2. Use the front panel knob, or the ⇧ ⇩ keys to highlight the LaserJet LAN printer in the table. See Figure 3-1.
3. Press **Select** . See Figure 3-1.

Figure 3-1 Selecting and Configuring the LAN Printer



php65c

4. Press **LAN Printr IP Addr**. Enter the IP address of the network printer you wish to use. Use the **Clear Entry** key to clear the current or default setting, and then enter the IP address using the analyzer's numeric keypad. (You can also use a keyboard connected to the rear panel DIN KEYBOARD connector to enter the IP address.)
5. Press **Prior Menu** and use the **Define PCL5** key to set up the printer configuration, and use the **Define Hardcopy** key to define the output. See your analyzer's *User's Guide* for information on configuring printers and defining output.

NOTE

You can print color screen dumps if you send the output to an HP Color LaserJet or HP Color LaserJet 5 printer. Press **Define PCL5** **Color**.

6. After you have completed the previous steps, you can send hardcopy to your LAN printer by simply pressing (HARDCOPY) **Start**.

If You Have Trouble Printing

- Make sure the analyzer's LAN IP address has been set (see "The Analyzer's IP Address and Hostname" on page 1-8).
- Make sure the printer is configured properly. Refer to your printer's documentation or your network administrator.
- Verify the LAN connection to the printer using the analyzer's built-in ping diagnostic utility (see "Troubleshooting the Initial Connection" on page 8-3).

4 Accessing the Analyzer's File System Using FTP

About This Chapter

This chapter shows you how to access the analyzer's file system using file transfer protocol (FTP). This chapter provides two simple examples: one example copies a file to the analyzer from your computer, and the other retrieves a file from the analyzer. The last section of this chapter contains a summary of commonly used ftp commands.

NOTE

It is important to distinguish among the several uses of the letters *ftp*:

FTP	*File Transfer Protocol*: a standardized service that provides methods to remotely transfer files among different computers and operating systems. The FTP service is implemented by many different computer applications, including programs named *ftp*.
ftp	The *name* given to many different computer programs, each implementing File Transfer Protocol (FTP). Programs with the name ftp are available for Windows 95, Windows NT, and Unix, for example.
ftp	The *letters you type* to start a program named ftp.

This chapter assumes that your analyzer is physically connected to your local area network. If it is not connected, refer to "Connecting the Analyzer to the LAN" on page 1- 5 for information on how to connect the system.

When you access the analyzer, you will have read and write access to the analyzer's file system (except for some files in the dynamic "data" disk, which are described in "The Dynamic Data Disk" on page 5- 2).

CAUTION

Avoid having more than one FTP session access your analyzer simultaneously. Files may be corrupted if both sessions attempt to use the same file at the same time.

This caution also applies to file system access performed via SCPI commands using LAN, GPIB, or IBASIC.

Using FTP to Access the Analyzer

If you are using a UNIX workstation, you have built-in networking software that includes ftp. The same is true if you are operating under Windows 95. If you are operating under Windows 3.1, you will need to have additional networking software that includes ftp.

NOTE

There are versions of FTP programs available with a graphical user interface (GUI). See "Using GUI FTP Software" on page 4- 10 for information on using these types of programs.

To access the analyzer's file system using FTP and the `ftp` utility

1. Enter the following command on your computer or workstation:

   ```
   ftp <hostname>
   ```

   or

   ```
   ftp <IP address>
   ```

   For example, type

   ```
   ftp  my8712
   ```

   or

   ```
   ftp 223.15.2.44
   ```

2. When the connection is made, you will be prompted for a login name and password. Enter your user name and password. The default login name is `network`, and the default password is `analyzer`. See "Managing User Names and Passwords" on page 1- 13.

3. You should now have a prompt on your computer display that looks like this:

   ```
   ftp>-
   ```

4. Type `dir` at the prompt. Your computer display should return something that looks like this:

   ```
   200 Port command okay
   150 Opening data connection for LIST /
   drwx------    2 root     sys         1024 Oct 9   int
   drwx------    2 root     sys         1024 Oct 9   nvram
   drwx------    2 root     sys         1024 Oct 9   ram
   drwx------    2 root     sys         1024 Oct 9   data
   226 File sent OK
   ```

The first character in the first field indicates the entry type. A “d” indicates that the entry is a directory. A “–” indicates that the entry is an ordinary file.

The next nine characters in the first field are interpreted as three sets of three bits each. The first three bits identify access permissions for the user (rwx). The second three bits are left blank. The final three bits identify the file type:

- A — archive file
- H — hidden file
- S — system file

You can read and write files to:

- int — a DOS-formatted floppy disk in the analyzer's 3.5” floppy disk drive
- nvram — the analyzer's internal non-volatile memory
- ram — the analyzer's internal volatile memory

The data directory is a dynamic data disk with files that are linked directly to analyzer operations. See “The Dynamic Data Disk” on page 5- 2 for information on accessing and using this directory.

5. Use the examples in this chapter to copy a file to the analyzer and to retrieve a file from the analyzer. Also see “Commonly Used FTP Commands” on page 4- 8.

Example 1: Copying a File to the Analyzer

You can copy files from your computer to your analyzer. For instance, you may want to develop an IBASIC program on your computer and then copy it to the analyzer so that you can run it from the front panel of the analyzer.

This example copies a file, “ib_prog”, from your computer to the analyzer's `nvram` disk:

1. On your computer or workstation change directories to the directory that contains the file “ib_prog.”
2. On your computer or workstation access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. The username and password pair must be one of the entries in the analyzer's access list. Refer to “Managing User Names and Passwords” on page 1- 13 for details.
3. Change to the non-volatile RAM disk in the analyzer by typing `cd nvram` at the `ftp` prompt.
4. Specify the type of file you will be transferring by typing either `binary` or `ascii` at the `ftp` prompt. For this example, use `ascii`.

CAUTION Binary files can be corrupted if you attempt to transfer them in “ascii” mode.

5. Type `put ib_prog` at the `ftp` prompt.
6. Type `bye` at the `ftp` prompt to exit `ftp`.

You can now recall and run the program from the front panel of your analyzer.

1. Press SAVE RECALL **Select Disk** **Non-Vol RAM Disk** .
2. Press **Prior Menu** **Programs** . Use the front panel knob to highlight the `IB_PROG` file.
3. Press **Recall Program** **Run** .

NOTE You can also download and automatically run IBASIC programs by accessing the `data` disk. See "Copying an IBASIC Program to or from the Analyzer" on page 5- 7.

NOTE When copying files from a UNIX environment to the analyzer, files that do not meet the DOS file-naming criteria (no more than eight (8) characters in filename, with no more than three (3) characters in extension) will be truncated to comply. For example, if you copy a file from UNIX named "ibasic_program.abcd", it will appear as "ibasic_p.abc" on the analyzer. There will be no indication from `ftp` that this has occurred.

Example 2: Retrieving a File from the Analyzer

You can copy files from your analyzer to your computer. For instance, you may want to retrieve saved measurement data from your analyzer (or a group of analyzers) for statistical analysis on your computer. In another scenario, you may have automated your measurement system using an IBASIC program to save data or instrument states to the analyzer's RAM disk. Your remote computer could asynchronously copy and delete files from the RAM disk, back up data, and prevent the RAM disk from filling up.

You may also want to copy instrument states and calibrations to your computer as a backup, eliminating the need for backups on floppy disks. Analyzer files can also be saved to a remote computer using NFS (see Chapter 7, "Using the Network File System (NFS)," on page 7-1).

This example copies a file "STATE2.STA" from your analyzer's `nvram` disk to a directory on your computer or workstation.

1. On your computer or workstation access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password.

2. Change to the non-volatile RAM disk in the analyzer by typing `cd nvram` at the `ftp` prompt.
3. If necessary, use the `lcd` command to change the local directory on your computer where you want to put the file. For example: type `lcd /users/myname/871x_data`.
4. Specify the type of file you will be transferring by typing either `binary` or `ascii` at the `ftp` prompt.

CAUTION Binary files can be corrupted if you attempt to transfer them in "ascii" mode. For this example, use `binary`.

5. Type `get state2.sta` at the ftp prompt.
6. Type `bye` at the ftp prompt to exit ftp.
7. Verify the file was copied by listing the contents of the directory it was copied to.

Commonly Used FTP Commands

The exact commands you use within `ftp` depend on the software. If you are not familiar with your ftp software, type "?" or "help" at the `ftp` prompt to see a list of commands.

The following table provides a list and brief description of some commonly used `ftp` commands. See "The FTP Command" on page 9- 6 for a summary of `ftp`.

`ftp` Commands

Command	Description
ascii	Sets the file transfer type to ASCII.
binary	Sets the file transfer type to binary.
bye	Closes the connection to the host and exits ftp.
cd *remote_directory*	Sets the working directory on the host to *remote_directory*.
delete *remote_file*	Deletes *remote_file* or empty *remote_directory*.
dir [*remote_directory*]	Lists the contents of the specified *remote_directory*. If *remote_directory* is unspecified, the contents of the current remote directory are listed.
get *remote_file* [*local_file*]	Copies *remote_file* to *local_file*. If *local_file* is unspecified, ftp uses the *remote_file* name as the *local_file* name.
help	Provides a list of ftp commands.
help *command*	Provides a brief description of *command*.
lcd [*local_directory*]	Sets the local working directory to *local_directory*.
ls [*remote_directory*]	Lists the contents of the specified *remote_directory*. If the *remote_directory* is unspecified, the contents of the current remote directory are listed.
mget *remote_file [local_file]*	Copies *remote_file* to the local system. If globbing is enabled, globbing metacharacters are expanded. If *local_file* is unspecified, ftp uses the *remote_file* name as the *local_file* name.
mput *local_file [remote_file]*	Copies *local_file* to remote file. If *remote_file* is unspecified, ftp uses the *local_file* name as the *remote_file* name. If globbing is enabled, globbing characters are expanded.
put *local_file* [*remote_file*]	Copies *local_file* to *remote file*. If *remote_file* is unspecified, ftp uses the *local_file* name as the *remote_file* name.
quit	Closes the connection to the host and exits ftp.

Using GUI FTP Software

There are versions of FTP programs available with a graphical user interface (GUI). These programs can make transferring files between the analyzer and your PC a simple “drag and drop” operation.

NOTE The procedures in this section were developed using Reflection™ FTP for Windows NT. They are intended as examples only. Other GUI FTP software may not be able to understand the analyzer's directory format, and will probably have different steps.

Example: Transferring Files between the Analyzer and Your PC

This example copies a file, “ib_prog”, from your computer to the analyzer's `nvram` disk.

1. Start the Reflection™ FTP program and set the program options as follows:
   - Set **View** to **Split Window**. (View both the command window and the normal window.)
   - Under the **Options** menu, set **Server Directory Format** to **Automatic Server Determination**.
2. Type your analyzer's hostname in the **Server Name** box.
3. Click on **Open**.
4. Enter your user name.
5. Enter your password.
6. To change to the non-volatile RAM disk in the analyzer, click inside the command window and then type `cd nvram` at the `ftp>` prompt. You can also double-click on a directory to expand the directory listing and access the files in that directory.
7. Use the **Client** side of the window to change directories on your PC to the directory that contains the file “ib_prog”.
8. Click on the file “ib_prog” and “drag” it over to the **Server** side of the window and “drop” it.

9. The file has been transferred to the non-volatile RAM disk on your analyzer.
10. To drag and drop multiple files, hold down the Ctrl key on your PC while selecting files with the mouse. When you drag and drop, your entire selection will be transferred to the analyzer.
11. You can also transfer files from the analyzer to your computer by dragging files in the other direction.

CAUTION Be sure to use the appropriate file transfer method (binary or ASCII) for the file(s) you are transferring. If you are transferring files to or from the analyzer's dynamic data disk, check Table 5-1 on page 5-2 for file types.

NOTE In an ftp session, your analyzer is configured as an ftp server, while your computer is an ftp client. Your analyzer cannot act as an ftp client. This means that you cannot type `ftp` from the analyzer's prompt. You can start an ftp session from IBASIC, but that will still configure the analyzer as the ftp server.

5 Accessing the Analyzer's Dynamic Data Disk

The Dynamic Data Disk

Your analyzer has an ftp directory called “data,” which is a dynamic data disk. The files in this directory trigger analyzer operations. For example, you can put an instrument state into this directory and the analyzer will automatically recall this state. You can do the same with an IBASIC program: copy it to the analyzer's data directory and it will automatically run. You can also transfer a screen-image file from the analyzer in either GIF, PCX, or HP-GL format.

The following files make up the contents of the dynamic data disk:

Table 5-1 Contents of the Dynamic Data Disk

File	File Type	Description
readme.txt	ASCII	This file contains a brief description of each file in this directory.
state.sta[1]	binary	This file contains the analyzer's current instrument state settings. Instrument state settings consist of all the stimulus and response parameters that set up the analyzer to make a specific measurement including markers, limit lines, and memory traces. Instrument state information is saved and recalled for both measurement channels. You can either retrieve this information from the analyzer, or you can put another analyzer's instrument state information into this file, which will cause the analyzer to immediately enter the new instrument state.
cal.sta[1]	binary	This file contains the analyzer's current calibration and instrument state settings. The measurement calibration information is the measurement correction data that the analyzer creates when you make a calibration. Measurement calibration information is saved and recalled for both measurement channels. You can either retrieve this information from the analyzer, or you can put another analyzer's calibration and instrument state information into this file, which will cause the analyzer to immediately enter the new cal and instrument state.

File	File Type	Description
data.sta[2]	binary	This file contains the measurement data for both measurement channels. You can either retrieve this information from the analyzer, or you can put data trace information from another analyzer into this file.
tset_cal.cal[1]	binary	*For use with multiport test sets only.* This file contains the test set calibration data that currently resides on the analyzer's non-volatile RAM disk. You can either retrieve this information from the analyzer, or you can put test set calibration data into this file.
prog.bas[2]	ASCII	This file contains the currently loaded IBASIC program. You can either retrieve the program that is currently in this file, or copy a new program to this file.
prog_run.bas[2]	ASCII	This file accepts a copy of an IBASIC program, copies it to prog.bas, and immediately runs the program.
prog_run.scp[2]	ASCII	This file accepts a copy of a file containing SCPI commands and immediately executes the commands.
screen.hgl[3]	ASCII	This file contains the current screen image in HP-GL format. It is available for uploading to a file on your computer.
screen.gif	binary	This file contains the current screen image in GIF format. It is available for uploading to a file on your computer.
screen.pcx[3]	binary	This file contains the current screen image in PCX format. It is available for uploading to a file on your computer.
screen_m.hgl[3]	ASCII	This file contains the current screen image, as well as the current softkey menu, in HP-GL format. It is available for uploading to a file on your computer.
screen_m.pcx[4]	binary	This file contains the current screen image, as well as the current softkey menu, in PCX format. It is available for uploading to a file on your computer.

File	File Type	Description
screen_m.gif[4]	binary	This file contains the current screen image, as well as the current softkey menu, in GIF format. It is available for uploading to a file on your computer.
parm_all.txt[5]	ASCII	This file contains a listing of all of the instrument's operating parameters in ASCII text format.
parm_screen.txt[5]	ASCII	This file contains the information in the current operating parameters screen in ASCII text format.
trace1.prn[6]	ASCII	This file contains the measurement channel 1 measurement data in ASCII spreadsheet format.
trace2.prn[6]	ASCII	This file contains the measurement channel 2 measurement data in ASCII spreadsheet format.
trace1.s1p[6]	ASCII	This file contains the measurement channel 1 measurement data in Touchstone format.
trace2.s1p[6]	ASCII	This file contains the measurement channel 2 measurement data in Touchstone format.

1. See "Saving and Recalling Analyzer States" on page 5-5 for information on how to use this file.
2. See "Copying Programs to and from the Analyzer" on page 5-7 for information on how to use this file.
3. See "Copying a Screen Image to a Local File" on page 5-10 for information on how to use this file.
4. See "Copying a Screen Image to a Local File" on page 5-10 for information on how to use this file.
5. See "Copying Instrument Parameters in ASCII Text Format" on page 5-13 for information on how to use this file.
6. See "Retrieving Measurement Data in ASCII Format" on page 5-14 for information on how to use this file.

Saving and Recalling Analyzer States

This section describes how to use the `state.sta`, `cal.sta`, and `data.sta` files that reside in the data directory of the analyzer. See Table 5-1 on page 5-2 for a brief description of each of these files.

You may have a particular instrument state set up on an analyzer and would like to set up that state on one or more additional analyzers. To do this you should do the following:

1. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Managing User Names and Passwords" on page 1-13 for instructions on how to do this.

2. Type `cd data` at the `ftp` prompt.
3. Type `dir` at the `ftp` prompt to see the listing of files in this directory, as well as a short description of each file.
4. Type `binary` at the `ftp` prompt to specify a binary file transfer.
5. Type `get state.sta` at the prompt to copy the current instrument state file from the analyzer to your computer.
6. Close the connection and exit `ftp` by typing `bye` or `quit` at the prompt.
7. Now you can put the instrument state into a different analyzer. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password.

8. Type `cd data` at the `ftp` prompt.
9. Type `put state.sta` at the `ftp` prompt. This copies the contents of the state.sta file from your computer to the new analyzer you are connected to. The new analyzer will immediately reinitialize itself with the new instrument state.

The above procedure can be performed with the `cal.sta` and `data.sta` files as well.

CAUTION When transferring `*.sta` files between instruments with different model numbers and/or option configurations, it is possible that some instrument state settings will not be compatible. For example, if you try to put an instrument state with a stop frequency of 3 GHz into an Agilent Technologies 8712ET/ES, the instrument will limit the frequency to 1.3 GHz (its high frequency limit). When you transfer this file over ftp, you will not receive any warning or indicator that this has occurred.

NOTE It is possible to have saved an instrument state file from the front panel of the analyzer that contains not only the instrument state settings, but the current calibration and measurement data as well. Putting this one file into the `state.sta` file will cause the analyzer to recall instrument state, cal state, and measurement data.

NOTE When copying files from a UNIX environment to the analyzer, files that do not meet the DOS file-naming criteria (no more than eight [8] characters in filename, with no more than three [3] characters in extension) will be truncated, and will not generate an error message. For example, if you copy a file from UNIX named "ibasic_program.abcd", the destination file name will be `ibasic_p.abc`.

Copying Programs to and from the Analyzer

This section describes how to use the `prog.bas`, `prog_run.bas`, and `prog_run.scp` files that reside in the data directory of the analyzer. See Table 5-1 on page 5-2 for a brief description of each of these files.

TIP

Refer to Chapter 6, "Controlling the Analyzer via the LAN" for information on controlling the analyzer from a computer, and how to interact with an IBASIC program running in the analyzer.

Copying an IBASIC Program to or from the Analyzer

You can create IBASIC programs on your computer and copy them to your analyzer. Conversely, you can retrieve a copy of the currently loaded IBASIC program from your analyzer to your computer. From there you might want to copy it into another analyzer, or edit it.

To copy an IBASIC program file named "ib_prog" to the analyzer:

1. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.
2. Type `cd data` at the `ftp` prompt.
3. Type `put ib_prog prog.bas` at the prompt to put a copy of your program into the analyzer.
4. Close the connection and exit `ftp` by typing `bye` or `quit` at the prompt.

5. To run your IBASIC program, press (SYSTEM OPTIONS) **IBASIC** **Run** on the analyzer.

NOTE

You can eliminate this last step, and have your program run automatically by using the dynamic data disk file named `prog_run.bas`. See "Copying and Running a Program with One Command" on page 5-9.

To copy the currently loaded IBASIC program from your analyzer to your computer:

1. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.

2. Type `cd data` at the `ftp` prompt.
3. Type `dir` at the `ftp` prompt to see the listing of files in this directory, as well as a short description of each of them.
4. Type `get prog.bas` at the prompt to retrieve the current IBASIC program file from the analyzer. This copies the program file prog.bas to your computer. You may want to give the file a unique name on your local computer by typing something like this:

   ```
   get prog.bas newfile
   ```

5. Close the connection and exit `ftp` by typing `bye` or `quit` at the prompt.

Copying and Running a Program with One Command

You can create an IBASIC program or a file with a list of SCPI commands on your computer, and then copy and automatically run it by using the `prog_run.bas` and `prog_run.scp` files.

To copy the IBASIC program `ib_prog` to the analyzer and immediately run it, follow the instructions below:

1. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.

2. Type `cd data` at the `ftp` prompt.

3. Type `put ib_prog prog_run.bas` at the prompt. This copies your program to the analyzer and immediately runs it.

4. You can also copy a file with a list of SCPI commands to the `prog_run.scp` data file and the commands will be executed immediately. See your analyzer's *Programmer's Guide* for a list of SCPI commands.

   The file you copy to "prog_run.scp" should simply be a list of SCPI commands. Following is an example file containing SCPI commands:

   ```
   SENS1:FUNC 'XFR:POW:RAT 2,0';DET NBAN;*WAI
   CALC1:MARK:FUNC MAX
   DISP:WIND1:TRAC:Y:AUTO ONCE
   ```

   These commands set the analyzer to measure transmission, place a marker on the maximum point, and then set the measurement trace to autoscale mode.

NOTE Avoid the use of queries, as there is no way to read back the analyzer's response.

Copying a Screen Image to a Local File

This section describes how to copy a screen image from the analyzer to a file on your computer.

To copy a screen image to your computer

1. On your computer or workstation, access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

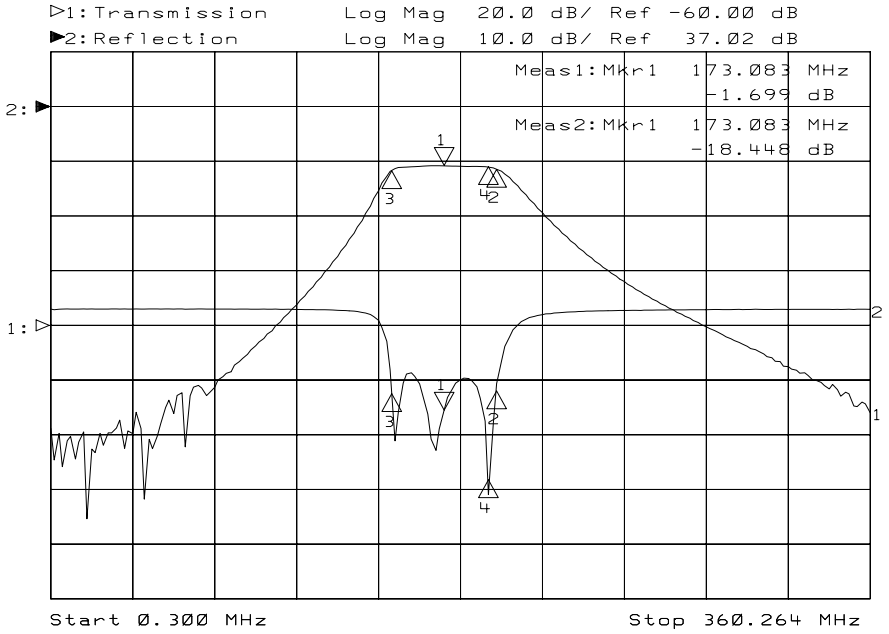
   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.

2. Type `cd data` at the `ftp` prompt.
3. Type `dir` at the `ftp` prompt to see the listing of files in this directory.
4. Type `binary` or `image` at the `ftp` prompt to specify a binary transfer.
5. Decide which screen image file you want to retrieve and then use the `get` command to transfer it to your computer. For example, type `get screen.pcx image.pcx` to retrieve the current screen image in PCX format, and place it into a file named `image.pcx` on your computer.

NOTE

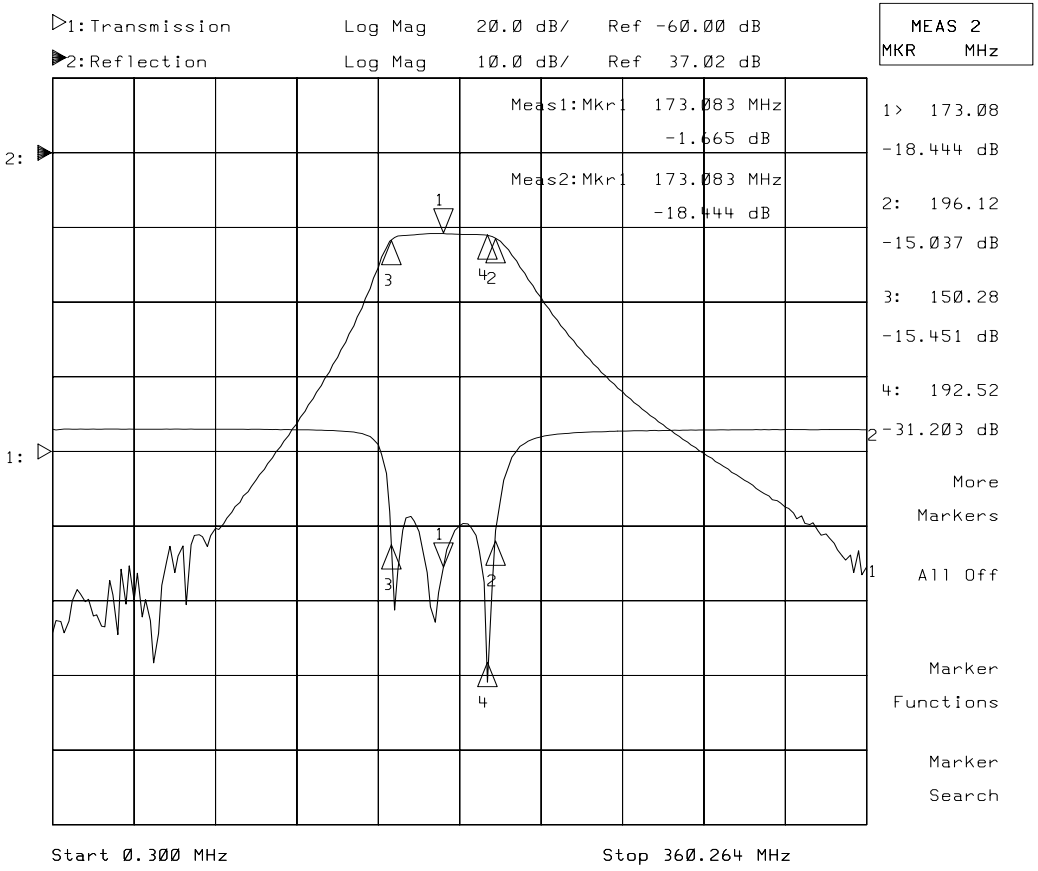
The appearance of the image you retrieve will depend on the selections in the Define Hardcopy menu on your analyzer, as well as the file you choose to retrieve from the data directory. For example, Figure 5-1 was retrieved with the "screen.hgl" file, using the analyzer's default hardcopy mode, which includes the measurement graph and the marker table. Figure 5-2 was retrieved with the analyzer's hardcopy mode defined as Graph Only, using the "screen_m.hgl" file. Figure 5-2 also includes the analyzer's softkeys. See Table 5-1 on page 5-2 for the filenames of screen images.

Figure 5-1 Screen Image with Marker Table Shown



1: Mkr (MHz)		dB	2: Mkr (MHz)		dB
1>	173.08	-1.699	1>	173.08	-18.448
2:	196.12	-2.827	2:	196.12	-15.076
3:	150.28	-3.354	3:	150.28	-15.438
4:	192.52	-2.113	4:	192.52	-31.236

Figure 5-2 Screen Image without Marker Table



Copying Instrument Parameters in ASCII Text Format

This section describes how to use the `parm_all.txt` and `parm_screen.txt` files that reside in the data directory of the analyzer. See Table 5-1 on page 5-2 for a brief description of each of these files.

Instrument parameters can be viewed on the analyzer by pressing (SYSTEM OPTIONS) **Operating Parameters**. Several screens of information are available (the exact number depends upon your model number and option configuration). These screens describe all the current settings and configurations of the analyzer. You can copy all of these screens using "parm_all.txt," or just the current screen with "parm_screen.txt" to an ASCII file on your computer.

To copy instrument operating parameters

1. On your computer or workstation access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.
2. Type `cd data` at the `ftp` prompt.
3. Type `get parm_all.txt` or `get parm_screen.txt` at the prompt to copy the desired parameters to your local computer. You can give the file a unique name on your local computer by typing:

   ```
   get parm_all.txt newfile
   ```
4. Close the connection and exit `ftp` by typing `bye` or `quit` at the prompt.

NOTE

If you do not rename the "parm_screen.txt" file when copying it to a DOS environment (as in step 3 above), it will be truncated to "parm_scr.txt" in order to comply with DOS file-naming conventions. There will be no indication from `ftp` that this has happened.

Retrieving Measurement Data in ASCII Format

This section describes how to use the `trace1.prn`, `trace2.prn`, `trace1.s1p` and `trace2.s1p` files that reside in the data directory of the analyzer. See Table 5-1 on page 5-2 for a brief description of each of these files.

Measurement data can be saved in ASCII formats that are compatible with many personal computer software packages. The files with the "`.prn`" extension in the data directory contain measurement data in a two-column format that can be directly imported into Lotus® 1-2-3®, as well as other spreadsheet programs. The files with an ".`s1p`" extension in the data directory contain measurement data in a format that can be directly imported into CAE programs such as EEsof's Microwave Design System (MDS) and Advanced Design System (ADS).

To retrieve measurement data

1. On your computer or workstation access the analyzer by typing `ftp <hostname>`. Enter your user name and password. For example, type

   ```
   ftp my8712
   user name
   password
   ```

   where `my8712` is the `<hostname>`, `user name` is your login name, and `password` is your user password. See "Using FTP to Access the Analyzer" on page 4-3 for instructions on how to do this.

2. Type `cd data` at the `ftp` prompt.

3. Type `get trace1.prn` at the prompt to copy the measurement channel 1 data in spreadsheet format. See Table 5-1 on page 5-2 for descriptions of the other trace data files. You can give the file a unique name on your local computer by typing:

   ```
   get trace1.prn newfile
   ```

4. Close the connection and exit `ftp` by typing `bye` or `quit` at the prompt.

Importing Graphics or Data into PC Applications

Some PC word processor and spreadsheet programs provide methods to import graphics and data from a LAN connection using the internet http: protocol. The following examples show how to import a screen image from your analyzer into Microsoft® Word 97, and how to import trace data from your analyzer into Microsoft® Excel 97.

Importing a Screen Snapshot into a Word Processor Program

This example steps you through importing a picture of the analyzer's current screen into a word processor. These steps were developed using Microsoft® Word 97. Other word processing programs may or may not have the same capability, and will probably have different steps:

1. Place the cursor at the point in your file where you want to place the imported graphic.
2. Click on **Insert**, **Picture**, **From File**. When the dialog box appears, type

   `http://my8712/data/screen.pcx`

   in the File name box.
3. Click on the Insert button in the dialog box.

NOTE

If you have previously imported a screen snapshot from the `screen.pcx` file, your computer may reload the file from its memory cache. To ensure that the screen snapshot is actually the current screen displayed on your analyzer, click the **Refresh Current Page** icon on the Web toolbar of Word.

You can also set the Windows operating system to refresh pages (or files) every time you access them. From your Windows desktop, select **Start>Settings>Control Panel>Internet**. Under **Temporary Internet Files**, select **Settings**. Under **Check for Newer Versions of Stored Pages**, select **Every Visit to the Page**.

Importing Trace Data into a Spreadsheet Program

This example steps you through importing the analyzer's current trace data into a spreadsheet program. These steps were developed using Microsoft® Excel 97. Other spreadsheet programs may or may not have the same capability, and will probably have different steps:

1. Click on **File**, **Open**. When the dialog box appears, type

   `http://my8712/data/trace1.s1p`

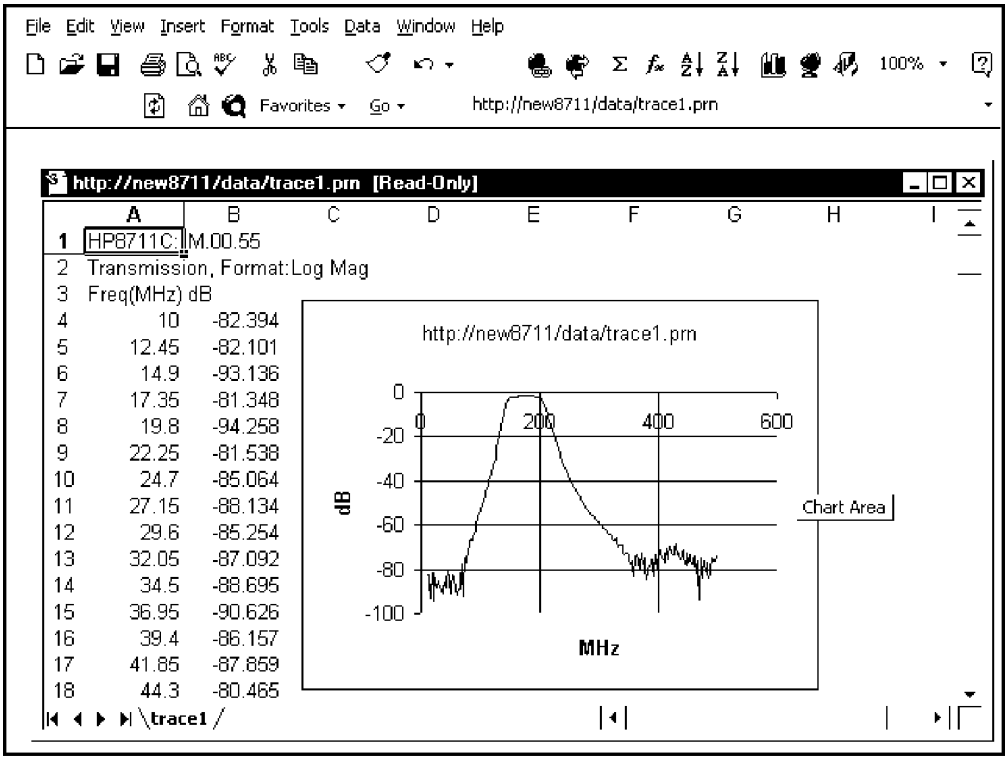
   in the File name box.
2. Click on the Open button in the dialog box.
3. A "Text Import Wizard" will guide you through customizing how you want the data to appear in the spreadsheet. Figure 5-3 shows trace data (and a screen snapshot) imported into a spreadsheet program.

NOTE

If you have previously imported trace data from the `trace1.s1p` file (or whichever of the four trace data files you are attempting to access), your computer may reload the file from its memory. To ensure that the screen snapshot is actually the current trace data displayed on your analyzer, click the **Refresh Current Page** icon on the Web toolbar of Excel.

You can also set the Windows operating system to refresh pages (or files) every time you access them. From your Windows desktop, select **Start>Settings>Control Panel>Internet.** Under **Temporary Internet Files**, select **Settings**. Under **Check for Newer Versions of Stored Pages**, select **Every Visit to the Page**.

Figure 5-3 Trace Data and Screen Snapshot Imported into a Spreadsheet



php66c

6 Controlling the Analyzer via the LAN

About This Chapter

NOTE

The example programs described in this chapter are on the *Example Programs Disk* that was shipped with your analyzer.

This chapter contains important information about how to control your analyzer. It includes a number of example programs, and has the following sections:

- Using Socket Programming to Control Your Analyzer
- Controlling the Analyzer via the Dynamic Data Disk
- Controlling the Analyzer with a C Program
- IBASIC Communication across the LAN
- Controlling Multiple Analyzers using a Perl Script
- Controlling the Analyzer using HP VEE
- Controlling the Analyzer with a Java™ Applet
- Controlling the Analyzer using SICL LAN

Using Socket Programming to Control Your Analyzer

Your analyzer implements a sockets applications programming interface (API) compatible with Berkeley sockets, Winsock, and other standard sockets APIs. You can write programs using sockets to control your analyzer by sending SCPI commands to a socket connection you create in your program. Refer to "Controlling the Analyzer with a C Program" on page 6-9 and "Controlling the Analyzer with a Java™ Applet" on page 6-33 for example programs using sockets to control the analyzer.

Setting Up Your Analyzer for Socket Programming

Before you can use socket programming, you must set up your analyzer by setting a socket port number. Follow the steps below to set up a port number for socket programming:

1. Press SYSTEM OPTIONS **LAN** **SCPI Sock. Setup** **Socket Port No.** .
2. Enter the port number that you will use for your socket connection to the analyzer.

   Port 5025 is set up for you as a default value.

NOTE Ports 1000 and below are reserved for common services such as Telnet, FTP, etc. It is good practice to select a port number greater than 2001 for the custom socket services you write. Consult your system administrator to find out what ports are available for your use.

If you would like to select or restore the default value for the **Socket Port No.** , press SYSTEM OPTIONS **LAN** **SCPI Sock. Setup** **Restore Defaults**

NOTE All of the example programs in this chapter that use socket programming use port 5025. If you change the port setting in your analyzer, and you use the example programs, be sure to change the port number in the programs source listings also.

Controlling the Analyzer via the Dynamic Data Disk

You can control your analyzer by accessing the data directory over the LAN. With this method you can do the following:

- load instrument states
- load and run IBASIC programs
- load trace data
- send SCPI command sequences to the analyzer

See "The Dynamic Data Disk" on page 5-2 and see "IBASIC Communication across the LAN" on page 6-24.

Entering Commands Directly with Telnet

Before connecting to your analyzer using telnet, you must have connected and configured your analyzer as described in Chapter 1, "Connecting and Configuring the Analyzer."

Using telnet to send commands to your analyzer works in a similar way to communicating over GPIB; you establish a connection with the analyzer, and then send or receive information using SCPI commands.

NOTE If you need to control the GPIB using "device clear" or SRQ's, you can use SICL LAN. SICL LAN provides control of your analyzer via GPIB over the LAN. See "Controlling the Analyzer using SICL LAN" on page 6-42.

The syntax of the telnet command is

```
telnet <hostname>
```

or

```
telnet <IP address>
```

NOTE You must have an entry in your `HOSTS` file for your analyzer in order to specify your analyzer by name in a `telnet` command. Alternately, you can enter the IP address directly in the `telnet` command, in place of the analyzer name. See "Using a Local HOSTS File" on page 7-11 for details.

For example, type

```
telnet 15.4.45.255
```

A brief message appears confirming the connection, and tells you the escape sequence for breaking the connection:

```
Trying...
Connected to 15.4.45.255.
Escape character is '^]'.
login:
```

When you connect to the analyzer, the analyzer will prompt you for a user name and password. Enter a user name and password that appear in the user access list. Refer to "Managing User Names and Passwords" on page 1-13 for information about the user access list. After you have entered a valid user name and password, the analyzer will display a welcome message and a command prompt:

```
Welcome to the HP871xE Network Analyzer at <IP address>
SCPI>
```

The analyzer is now ready to accept your SCPI commands. As you type analyzer SCPI commands, query results appear on the next line. When you are done, break the telnet connection using the escape character (in this case `Ctrl and "]"`), and type `quit`. To send a "Device Clear" command to the analyzer, enter `Ctrl c`. The analyzer will respond with the following message:

```
<Device Clear>
SCPI>
```

See the detailed example that follows.

Telnet Example

To connect to the analyzer named "my8712", enter the following command:

```
telnet my8712
```

The computer responds with the following messages:

```
Trying...
Connected to my8712
Escape character is '^]'.
login:
```

Enter a valid user name. The analyzer responds with the following prompt:

```
password:
```

Enter the password for the user name given above. The analyzer responds with a welcome message and the SCPI prompt:

```
Welcome to the HP871xE Network Analyzer at 15.4.45.255
SCPI>
```

You can immediately enter programming (SCPI) commands. Typical commands might be:

```
SENS1:FUNC 'XFR:POW:RAT 2,0':DET NBAN;*OPC?
CALC1:MARK:FUNC MAX
CALC1:MARK:POIN?
```

The above example sets the analyzer to measure transmission, places a marker on the maximum point, and then queries the analyzer for the amplitude of the marker.

You need to press Enter after typing in each command. After pressing Enter on the last line in the example above, the analyzer returns the amplitude level of the marker to your computer and displays it on the next line. For example, after typing `CALC1:MARK:POIN?` and pressing Enter, the computer would display:

```
+1.71000000000E+002
```

When you are done, close the telnet connection. Enter the escape character to get the telnet prompt. The escape character (Ctrl and "]" in this example) does not print.

At the telnet prompt, type `quit` or `close`.

The telnet connection closes and you see your regular prompt.

```
Connection closed.
$
```

NOTE You can also control your analyzer from your web browser. See "Accessing the Analyzer with Your Web Browser" on page 2-4.

NOTE If your telnet connection is in a mode called "line-by-line," there is no local echo. This means you will not be able to see the characters you are typing on your computer's display until *after* you press the Enter key.

To remedy this, you need to change your telnet connection to "character-by-character" mode. This can be accomplished in most systems by escaping out of telnet to the `telnet>` prompt and then typing `mode char`. If this does not work, consult your telnet program's documentation for how to change to "character-by-character" mode.

Controlling the Analyzer with a C Program

The following example program demonstrates simple socket programming. It is written in C, and compiles in the HP-UX UNIX environment, or the WIN32 environment. It is portable to other UNIX environments with only minor changes.

In UNIX, LAN communication via sockets is very similar to reading or writing a file. The only difference is the openSocket( ) routine, which uses a few network library routines to create the TCP/IP network connection. Once this connection is created, the standard fread( ) and fwrite( ) routines are used for network communication.

In Windows, the routines send( ) and recv( ) must be used, since fread( ) and fwrite( ) may not work on sockets.

The program reads the analyzer's hostname from the command line, followed by the SCPI command. It then opens a socket to the analyzer using port 5025, and sends the command. If the command appears to be a query, the program queries the analyzer for a response, and prints the response.

NOTE Port 5025 is the default port for SCPI socket programming. To use a different port, refer to "Using Socket Programming to Control Your Analyzer" on page 6-3.

This example program can also be used as a utility to talk to your analyzer from the command prompt on your UNIX workstation or Windows 95 PC, or from within a script.

NOTE This program is included on the *Example Programs Disk* shipped with your analyzer as `lanio.c`.

```
/*
**************************************************************************
*  $Header: lanio.c,v 1.5 96/10/04 20:29:32 roger Exp $
*  $Revision: 1.5 $
*  $Date: 96/10/04 20:29:32 $
*
*  $Contributor:     LSID, MID $
*
*  $Description:     Functions to talk to an HP 8711C/12C/13C/14C/30A
*                    analyzer via TCP/IP.  Uses command-line arguments.
*
*                    A TCP/IP connection to port 5025 is established and
*                    the resultant file descriptor is used to "talk" to the
*                    instrument using regular socket I/O mechanisms. $
*
*
*
*  871xC Examples:
*
*   Query the center frequency:
*        lanio 15.4.43.5 'sens:freq:cent?'
*
*   Select Lin Mag format:
*        lanio my8711.sr.hp.com 'CALC:FORM  MLIN'
*
*   Take a sweep, wait for end of sweep, move mkr to peak and query x pos:
*        lanio my8711 ':abort;INIT:CONT OFF;:INIT1;*wai; :calc:mark:max; x?'
*
*   Query X and Y values of marker 1 and marker 2 (assumes they are on):
*        lanio my8711 'calc:mark1:x?;y?; :calc:mark2:x?;y?'
*
*   Check for errors (gets one error):
*        lanio my8711 'syst:err?'
*
*   Send a list of commands from a file, and number them:
*        cat scpi_cmds | lanio -n my8711
*
**************************************************************************
*
*  This program compiles and runs under
*     - HP-UX 9.05 (UNIX), using HP cc or gcc:
*          + cc -Aa    -O -o lanio  lanio.c
*          + gcc -Wall -O -o lanio  lanio.c
*
*     - Windows 95, using Microsoft Visual C++ 4.0 Standard Edition
*     - Windows NT 3.51, using Microsoft Visual C++ 4.0
*          + Be sure to add  WSOCK32.LIB  to your list of libraries!
*          + Compile both lanio.c and getopt.c
*          + Consider re-naming the files to lanio.cpp and getopt.cpp
*
*  Considerations:
*     - On UNIX systems, file I/O can be used on network sockets.
*       This makes programming very convenient, since routines like
*       getc(), fgets(), fscanf() and fprintf() can be used.  These
*       routines typically use the lower level read() and write() calls.
*
*     - In the Windows environment, file operations such as read(), write(),
*       and close() cannot be assumed to work correctly when applied to
*       sockets.  Instead, the functions send() and recv() MUST be used.
*/
```

```
/* Support both Win32 and HP-UX UNIX environment */
#ifdef _WIN32     /* Visual C++ 4.0 will define this */
#  define WINSOCK
#endif

#ifndef WINSOCK
#  ifndef _HPUX_SOURCE
#  define _HPUX_SOURCE
#  endif
#endif

#include <stdio.h>         /* for fprintf and NULL  */
#include <string.h>        /* for memcpy and memset */
#include <stdlib.h>        /* for malloc(), atol() */
#include <errno.h>         /* for strerror          */

#ifdef WINSOCK

#include <windows.h>

#  ifndef _WINSOCKAPI_
#  include <winsock.h>   // BSD-style socket functions
#  endif

#else /* UNIX with BSD sockets */

#  include <sys/socket.h>    /* for connect and socket*/
#  include <netinet/in.h>    /* for sockaddr_in       */
#  include <netdb.h>         /* for gethostbyname     */

#  define SOCKET_ERROR (-1)
#  define INVALID_SOCKET (-1)

   typedef  int SOCKET;

#endif /* WINSOCK */

#ifdef WINSOCK
  /* Declared in getopt.c.  See example programs disk. */
  extern char *optarg;
  extern int  optind;
  extern int getopt(int argc, char * const argv[], const char* optstring);
#else
#  include <unistd.h>           /* for getopt(3C) */
#endif

#define COMMAND_ERROR  (1)
#define NO_CMD_ERROR  (0)

#define SCPI_PORT  5025
#define INPUT_BUF_SIZE (64*1024)


/*********************************************************************
 * Display usage
 *********************************************************************/
static void usage(char *basename)
{
    fprintf(stderr,"Usage: %s [-nqu] <hostname> [<command>]\n", basename);
```

```
    fprintf(stderr,"       %s [-nqu] <hostname> < stdin\n", basename);
    fprintf(stderr,"  -n, number output lines\n");
    fprintf(stderr,"  -q, quiet; do NOT echo lines\n");
    fprintf(stderr,"  -e, show messages in error queue when done\n");
}


#ifdef WINSOCK
int init_winsock(void)
{
    WORD wVersionRequested;
    WSADATA wsaData;
    int err;
    wVersionRequested = MAKEWORD(1, 1);
    wVersionRequested = MAKEWORD(2, 0);

    err = WSAStartup(wVersionRequested, &wsaData);

    if (err != 0) {
        /* Tell the user that we couldn't find a useable */
        /* winsock.dll.     */
        fprintf(stderr, "Cannot initialize Winsock 1.1.\n");
        return -1;
    }
    return 0;
}

int close_winsock(void)
{
    WSACleanup();
    return 0;
}
#endif /* WINSOCK */


/*********************************************************************
 *
 > $Function: openSocket$
 *
 * $Description:  open a TCP/IP socket connection to the instrument $
 *
 * $Parameters:  $
 *    (const char *) hostname . . . . Network name of instrument.
 *                                    This can be in dotted decimal notation.
 *    (int) portNumber  . . . . . . . The TCP/IP port to talk to.
 *                                    Use 5025 for the SCPI port.
 *
 * $Return:     (int)  . . . . . . . . A file descriptor similar to open(1).$
 *
 * $Errors:     returns -1 if anything goes wrong $
 *
 *********************************************************************/
SOCKET openSocket(const char *hostname, int portNumber)
{
    struct hostent *hostPtr;
    struct sockaddr_in peeraddr_in;
    SOCKET s;
```

```
    memset(&peeraddr_in, 0, sizeof(struct sockaddr_in));

    /*********************************************/
    /* map the desired host name to internal form. */
    /*********************************************/
    hostPtr = gethostbyname(hostname);
    if (hostPtr == NULL)
    {
        fprintf(stderr,"unable to resolve hostname '%s'\n", hostname);
        return INVALID_SOCKET;
    }

    /******************/
    /* create a socket */
    /******************/
    s = socket(AF_INET, SOCK_STREAM, 0);
    if (s == INVALID_SOCKET)
    {
        fprintf(stderr,"unable to create socket to '%s': %s\n",
                hostname, strerror(errno));
        return INVALID_SOCKET;
    }

    memcpy(&peeraddr_in.sin_addr.s_addr, hostPtr->h_addr, hostPtr->h_length);
    peeraddr_in.sin_family = AF_INET;
    peeraddr_in.sin_port = htons((unsigned short)portNumber);

    if (connect(s, (const struct sockaddr*)&peeraddr_in,
                sizeof(struct sockaddr_in)) == SOCKET_ERROR)
    {
        fprintf(stderr,"unable to create socket to '%s': %s\n",
                hostname, strerror(errno));
        return INVALID_SOCKET;
    }

    return s;
}


/*****************************************************************************
 *
 > $Function: commandInstrument$
 *
 * $Description:  send a SCPI command to the instrument.$
 *
 * $Parameters:  $
 *     (FILE *) . . . . . file pointer associated with TCP/IP socket.
 *     (const char *command)  . . SCPI command string.
 * $Return:  (char *) . . . . . a pointer to the result string.
 *
 * $Errors:   returns 0 if send fails $
 *
 *****************************************************************************/
int commandInstrument(SOCKET sock,
                      const char *command)
{
    int count;
```

```
    /* fprintf(stderr, "Sending \"%s\".\n", command);  */
    if (strchr(command, '\n') == NULL) {
       fprintf(stderr, "Warning: missing newline on command %s.\n", command);
    }

    count = send(sock, command, strlen(command), 0);
    if (count == SOCKET_ERROR) {
        return COMMAND_ERROR;
    }

    return NO_CMD_ERROR;
}


/*************************************************************************
 * recv_line(): similar to fgets(), but uses recv()
 *************************************************************************/
char * recv_line(SOCKET sock, char * result, int maxLength)
{
#ifdef WINSOCK
    int cur_length = 0;
    int count;
    char * ptr = result;
    int err = 1;

    while (cur_length < maxLength) {
        /* Get a byte into ptr */
        count = recv(sock, ptr, 1, 0);

        /* If no chars to read, stop. */
        if (count < 1) {
            break;
        }
        cur_length += count;

        /* If we hit a newline, stop. */
        if (*ptr == '\n') {
            ptr++;
            err = 0;
            break;
        }
        ptr++;

    }

    *ptr = '\0';

    if (err) {
        return NULL;
    } else {
        return result;
    }
#else
    /*************************************************************************
     * Simpler UNIX version, using file I/O.  recv() version works too.
     * This demonstrates how to use file I/O on sockets, in UNIX.
     *************************************************************************/
    FILE * instFile;
    instFile = fdopen(sock, "r+");
    if (instFile == NULL)
```

```
    {
        fprintf(stderr, "Unable to create FILE * structure : %s\n",
                strerror(errno));
        exit(2);
    }
    return fgets(result, maxLength, instFile);
#endif
}


/*****************************************************************************
 *
 > $Function: queryInstrument$
 *
 * $Description:  send a SCPI command to the instrument, return a response.$
 *
 * $Parameters:  $
 *     (FILE *) . . . . . file pointer associated with TCP/IP socket.
 *     (const char *command)  . . SCPI command string.
 *     (char *result) . . . . . where to put the result.
 *     (size_t) maxLength . . . . maximum size of result array in bytes.
 *
 * $Return:  (long) . . . . . The number of bytes in result buffer.
 *
 * $Errors:   returns 0 if anything goes wrong. $
 *
 *****************************************************************************/
long queryInstrument(SOCKET sock,
                     const char *command, char *result, size_t maxLength)
{
    long ch;
    char tmp_buf[8];
    long resultBytes = 0;
    int command_err;
    int count;

    /*******************************************************
     * Send command to analyzer
     *******************************************************/
    command_err = commandInstrument(sock, command);
    if (command_err) return COMMAND_ERROR;


    /*******************************************************
     * Read response from analyzer
     *******************************************************/
    count = recv(sock, tmp_buf, 1, 0); /* read 1 char */
    ch = tmp_buf[0];

    if ((count < 1) || (ch == EOF)  || (ch == '\n'))
    {
        *result = '\0';  /* null terminate result for ascii */
        return 0;
    }

    /* use a do-while so we can break out */
    do
    {
        if (ch == '#')
```

```
{
    /* binary data encountered - figure out what it is */
    long numDigits;
    long numBytes = 0;
    /* char length[10]; */

    count = recv(sock, tmp_buf, 1, 0); /* read 1 char */
    ch = tmp_buf[0];
    if ((count < 1) || (ch == EOF)) break; /* End of file */

    if (ch < '0' || ch > '9') break;  /* unexpected char */
    numDigits = ch - '0';

    if (numDigits)
    {
        /* read numDigits bytes into result string. */
        count = recv(sock, result, (int)numDigits, 0);
        result[count] = 0;  /* null terminate */
        numBytes = atol(result);
    }

    if (numBytes)
    {
        resultBytes = 0;
        /* Loop until we get all the bytes we requested. */
        /* Each call seems to return up to 1457 bytes, on HP-UX 9.05 */
        do {
            int rcount;
            rcount = recv(sock, result, (int)numBytes, 0);
            resultBytes += rcount;
            result      += rcount;  /* Advance pointer */
        } while ( resultBytes < numBytes );

        /*********************************************************
         * For LAN dumps, there is always an extra trailing newline
         * Since there is no EOI line.  For ASCII dumps this is
         * great but for binary dumps, it is not needed.
         *********************************************************/
        if (resultBytes == numBytes)
        {
            char junk;
            count = recv(sock, &junk, 1, 0);
        }
    }
    else
    {
        /* indefinite block ... dump til we can an extra line feed */
        do
        {
            if (recv_line(sock, result, maxLength) == NULL) break;
            if (strlen(result)==1 && *result == '\n') break;
            resultBytes += strlen(result);
            result += strlen(result);
        } while (1);
    }
}
else
{
    /* ASCII response (not a binary block) */
    *result = (char)ch;
```

```
            if (recv_line(sock, result+1, maxLength-1) == NULL) return 0;

            /* REMOVE trailing newline, if present.  And terminate string. */
            resultBytes = strlen(result);
            if (result[resultBytes-1] == '\n') resultBytes -= 1;
            result[resultBytes] = '\0';
        }
    } while (0);

    return resultBytes;
}



/*****************************************************************************
 *
 > $Function: showErrors$
 *
 * $Description: Query the SCPI error queue, until empty.  Print results. $
 *
 * $Return:  (void)
 *
 *****************************************************************************/
void showErrors(SOCKET sock)
{
    const char * command = "SYST:ERR?\n";
    char result_str[256];

    do {
        queryInstrument(sock, command, result_str, sizeof(result_str)-1);

        /*****************************************************************
         * Typical result_str:
         *     -221,"Settings conflict; Frequency span reduced."
         *     +0,"No error"
         * Don't bother decoding.
         *****************************************************************/
        if (strncmp(result_str, "+0,", 3) == 0) {
            /* Matched +0,"No error" */
            break;
        }
        puts(result_str);
    } while (1);

}


/*****************************************************************************
 *
 > $Function: isQuery$
 *
 * $Description: Test current SCPI command to see if it a query. $
 *
 * $Return:  (unsigned char) . . . non-zero if command is a query.  0 if not.
 *
 *****************************************************************************/
unsigned char isQuery( char* cmd )
{
    unsigned char q = 0 ;
```

```
    char *query ;

    /*********************************************************/
    /* if the command has a '?' in it, use queryInstrument.  */
    /* otherwise, simply send the command.                   */
    /* Actually, we must a little more specific so that      */
    /* marker value queries are treated as commands.         */
    /* Example:  SENS:FREQ:CENT (CALC1:MARK1:X?)             */
    /*********************************************************/
    if ( (query = strchr(cmd,'?')) != NULL)
    {
        /* Make sure we don't have a marker value query, or
         * any command with a '?' followed by a ')' character.
         * This kind of command is not a query from our point of view.
         * The analyzer does the query internally, and uses the result.
         */
        query++ ;       /* bump past '?' */
        while (*query)
        {
            if (*query == ' ') /* attempt to ignore white spc */
                query++ ;
            else break ;
        }

        if ( *query != ')' )
        {
            q = 1 ;
        }
    }
    return q ;
}



/*****************************************************************************
 *
 > $Function: main$
 *
 * $Description: Read command line arguments, and talk to analyzer.
                 Send query results to stdout. $
 *
 * $Return:  (int) . . . non-zero if an error occurs
 *
 *****************************************************************************/
int main(int argc, char *argv[])
{

    SOCKET instSock;
    char *charBuf = (char *) malloc(INPUT_BUF_SIZE);
    char *basename;
    int chr;
    char command[1024];
    char *destination;
    unsigned char quiet = 0;
    unsigned char show_errs = 0;
    int number = 0;

    basename = strrchr(argv[0], '/');
    if (basename != NULL)
        basename++ ;
```

```
    else
        basename = argv[0];

    while ( ( chr = getopt(argc,argv,"qune")) != EOF )
        switch (chr)
        {
            case 'q':  quiet = 1; break;
            case 'n':  number = 1; break ;
            case 'e':  show_errs = 1; break ;
            case 'u':
            case '?':  usage(basename); exit(1) ;
        }

    /* now look for hostname and optional <command> */
    if (optind < argc)
    {
        destination = argv[optind++] ;
        strcpy(command, "");
        if (optind < argc)
        {
            while (optind < argc) {
                /* <hostname> <command> provided; only one command string */
                strcat(command, argv[optind++]);
                if (optind < argc) {
                    strcat(command, " ");
                } else {
                    strcat(command, "\n");
                }
            }
        }
        else
        {
            /* Only <hostname> provided; input on <stdin> */
            strcpy(command, "");

            if (optind > argc)
            {
                usage(basename);
                exit(1);
            }
        }
    }
    else
    {
        /* no hostname! */
        usage(basename);
        exit(1);
    }

    /*******************************************/
    /* open a socket connection to the instrument */
    /*******************************************/
#ifdef WINSOCK
    if (init_winsock() != 0) {
        exit(1);
    }
#endif /* WINSOCK */

    instSock = openSocket(destination, SCPI_PORT);
    if (instSock == INVALID_SOCKET) {
```

```
    fprintf(stderr, "Unable to open socket.\n");
    return 1;
}
/* fprintf(stderr, "Socket opened.\n"); */

if (strlen(command) > 0)
{
    /*******************************************************/
    /* if the command has a '?' in it, use queryInstrument. */
    /* otherwise, simply send the command.                  */
    /*******************************************************/
    if ( isQuery(command) )
    {
        long bufBytes;
        bufBytes = queryInstrument(instSock, command,
                                   charBuf, INPUT_BUF_SIZE);
        if (!quiet)
        {
            fwrite(charBuf, bufBytes, 1, stdout);
            fwrite("\n", 1, 1, stdout) ;
            fflush(stdout);
        }
    }
    else
    {
        commandInstrument(instSock, command);
    }
}
else
{
    /* read a line from <stdin> */
    while ( gets(charBuf) != NULL )
    {
        if ( !strlen(charBuf) )
            continue ;

        if ( *charBuf == '#' || *charBuf == '!' )
            continue ;

        strcat(charBuf, "\n");

        if (!quiet)
        {
            if (number)
            {
                char num[10];
                sprintf(num,"%d: ",number);
                fwrite(num, strlen(num), 1, stdout);
            }
            fwrite(charBuf, strlen(charBuf), 1, stdout) ;
            fflush(stdout);
        }

        if ( isQuery(charBuf) )
        {
            long bufBytes;

            /* Put the query response into the same buffer as the
             * command string appended after the null terminator.
             */
```

```
                bufBytes = queryInstrument(instSock, charBuf,
                                           charBuf + strlen(charBuf) + 1,
                                           INPUT_BUF_SIZE -strlen(charBuf) );
                if (!quiet)
                {
                    fwrite("  ", 2, 1, stdout) ;
                    fwrite(charBuf + strlen(charBuf)+1, bufBytes, 1, stdout);
                    fwrite("\n", 1, 1, stdout) ;
                    fflush(stdout);
                }
            }
            else
            {
                commandInstrument(instSock, charBuf);
            }
            if (number) number++;
        }
    }

    if (show_errs) {
        showErrors(instSock);
    }

#ifdef WINSOCK
    closesocket(instSock);
    close_winsock();
#else
    close(instSock);
#endif /* WINSOCK */

    return 0;
}

/* End of lanio.c */
```

```
/*****************************************************************************

 getopt(3C)                                                        getopt(3C)


 NAME
      getopt - get option letter from argument vector

 SYNOPSIS
      int getopt(int argc, char * const argv[], const char *optstring);

      extern char *optarg;
      extern int optind, opterr, optopt;

 DESCRIPTION
      getopt returns the next option letter in argv (starting from argv[1])
      that matches a letter in optstring.  optstring is a string of
      recognized option letters; if a letter is followed by a colon, the
      option is expected to have an argument that may or may not be
      separated from it by white space.  optarg is set to point to the start
      of the option argument on return from getopt.

      getopt places in optind the argv index of the next argument to be
      processed.  The external variable optind is initialized to 1 before
      the first call to the function getopt.

      When all options have been processed (i.e., up to the first non-option
      argument), getopt returns EOF.  The special option -- can be used to
      delimit the end of the options; EOF is returned, and -- is skipped.

 *****************************************************************************/


#include <stdio.h>      /* For NULL, EOF */
#include <string.h>     /* For strchr() */

char    *optarg;        /* Global argument pointer. */
int     optind = 0;     /* Global argv index. */

static char     *scan = NULL;   /* Private scan pointer. */

int getopt( int argc, char * const argv[], const char* optstring)
{
    char c;
    char *posn;

    optarg = NULL;

    if (scan == NULL || *scan == '\0') {
        if (optind == 0)
            optind++;

        if (optind >= argc || argv[optind][0] != '-' || argv[optind][1] == '\0')
            return(EOF);
        if (strcmp(argv[optind], "--")==0) {
            optind++;
            return(EOF);
        }

        scan = argv[optind]+1;
```

```
    optind++;
}

c = *scan++;
posn = strchr(optstring, c);        /* DDP */

if (posn == NULL || c == ':') {
    fprintf(stderr, "%s: unknown option -%c\n", argv[0], c);
    return('?');
}

posn++;
if (*posn == ':') {
    if (*scan != '\0') {
        optarg = scan;
        scan = NULL;
    } else {
        optarg = argv[optind];
        optind++;
    }
}

return(c);
              }
```

IBASIC Communication across the LAN

You may need a way for an IBASIC program running on the analyzer to signal a remote computer that it has completed some operation.

IBASIC cannot communicate directly across LAN using the ASSIGN and OUTPUT or ENTER commands. However, IBASIC can use the following SCPI command to send a message to a remote computer via LAN:

```
DIAGnostic:COMMunicate:LAN:SEND <IP_ADDR>,<PORT_NUM>,<STRING>
```

This command opens a socket to the remote computer, and sends the specified string. The <IP_ADDR> argument specifies the IP address of the remote computer. The <PORT_NUM> argument specifies the port number to use. The <STRING> is the message to be sent.

For example:

```
DIAGnostic:COMMunicate:LAN:SEND '15.4.40.49',8001,'Ready!'
```

If the remote computer is not listening for a LAN connection at the specified port, this command will block, and wait for the remote computer to accept the connection. After about 75 seconds, it will time out. This is the standard TCP/IP timeout period.

The following IBASIC example program demonstrates LAN communication using IBASIC.

The section "Controlling Multiple Analyzers using a Perl Script" on page 6-28 shows an example of a program that monitors the LAN for a response from an IBASIC program.

NOTE

This program is included on the *Example Programs Disk* shipped with your analyzer as LAN_SEND.

```
100  !
110  !  This program demonstrates how IBASIC can communicate
120  !  with a remote computer via LAN.  This is done using a
130  !  SCPI command that sends a LAN message to the computer:
140  !     DIAG:COMM:LAN:SEND '15.4.40.49',8003,'Ready!'
150  !  8003 is an arbitrary unused port number.
160  !
170  DIM Cmd$[256]
180  DIM Msg$[128]
190  DIM Snum$[16]
200  !
210  ! Initialize the instrument
220  !
230  ASSIGN @Na TO 800
240  OUTPUT @Na;"SYST:PRES;*WAI"
250  OUTPUT @Na;"SENS1:STAT ON; *WAI"
260  OUTPUT @Na;"POW1:MODE FIXed"   ! Freq sweep
270  OUTPUT @Na;"DISP:ANN:FREQ1:MODE CSPAN"
280  OUTPUT @Na;"SENS1:FREQ:CENT 177e6;SPAN 200e6;*WAI"
290  ! Put sweep in hold
300  OUTPUT @Na;"ABOR;:INIT1:CONT OFF;*WAI"
310  ! Sync up with analyzer
320  OUTPUT @Na;"*OPC?"
330  ENTER @Na;Opc
340  !
350  !
360  ! Get serial number
370  OUTPUT @Na;"DIAG:SNUM?"
380  ENTER @Na;Snum$
390  Snum$=Snum$[2,11]    ! remove quotes
400  !
410  ! Begin infinite loop:
420  !   - Take sweep
430  !   - Compute bandwidth
440  !   - Send signal to computer
```

```
450  !
460 Loop: !
470  DISP "Taking sweep..."
480  Count=Count+1
490  ! Take sweep, and wait for it to finish.
500  OUTPUT @Na;"INIT1;*OPC?"
510  ENTER @Na;Opc
520  ! Autoscale trace to give feedback.
530  OUTPUT @Na;"DISP:WIND1:TRAC:Y:AUTO ONCE"
540  ! Perform a search for the -3 dB bandwidth of the filter
550  ! This function uses several markers to find 4 key values.
560  OUTPUT @Na;"CALC1:MARK:BWID -3;FUNC:RES?"
570  ! Read the four values:  the bandwidth, center
580  ! frequency, Q and the insertion loss.
590  ENTER @Na;Bwid,Center_f,Q,Loss
600  !
610  ! Signal computer that we are done,
620  ! so that it can come grab the meas results
630  !
640  ! Create a string that looks like this:
650  !    Ready!,"US36100007",6.159E+7,1.7248E+8,-1.6088<LF>
660  ! Could send any string.  Could also save meas results to
670  ! a file, and send filename, and computer could FTP the file.
680  !
690  Msg$="'Ready!,"&Snum$&","
700  Msg$=Msg$&val$(Bwid)&","&val$(Center_f)&","&val$(Loss)
710  Msg$=Msg$&chr$(10)&"'"
720  !
730  ! Send the message to the computer, via LAN
740  !
750  Cmd$="DIAG:COMM:LAN:SEND '15.4.40.49',8003,"&Msg$
760  OUTPUT @Na;Cmd$
770  DISP "Done with loop ";Count;"  Continuing..."
780  !
790  ! Pause, and wait for computer to grab data.
```

```
800  ! Computer will send 'PROG:STAT CONT' when ready
810  !
820  PAUSE
830  GOTO Loop
840  END
```

Controlling Multiple Analyzers using a Perl Script

The following Perl script demonstrates how you can control a network of analyzers from your workstation.

The script downloads an IBASIC program to a group of analyzers. The IBASIC program makes a measurement, and then signals the computer that it needs service. (See the previous section, "IBASIC Communication across the LAN" on page 6-24 to see how the IBASIC program accomplishes this.)

The computer receives this signal, then queries the analyzer for measurement data, and then tells the IBASIC program to continue.

NOTE This program is included on the *Example Programs Disk* shipped with your analyzer as `lan_serv`.

```
#! /usr/bin/perl

#
# Perl script to listen on a port, and print received messages.
#
# This script is based on the "server" example in the
# book "Programming perl" by O'Reilley & Associates, Inc.
#

# require 'sys/socket.ph';      # Not needed on HP-UX
require 'ctime.pl';             # Allow use of ctime() to get date


# Get the port number from the command line (first arg).
# If no argument, default to a high-numbered port.
# Users can use ports above 1024 or so.
($port) = @ARGV;
$port = 8003 unless $port;


$AF_INET = 2;           # from /usr/include/sys/socket.h
$SOCK_STREAM = 1;       # from /usr/include/sys/socket.h
$PF_INET = $AF_INET;    # from /usr/include/sys/socket.h


# Is this line noise?  No, it's the pack format:
#  S = unsigned short, n = short in network order
#  a4 = 4 ascii characters, null padded,
#  x8 = 8 null bytes (?)
$sockaddr = 'S n a4 x8';

chop($this_hostname = `hostname`);
```

```
($name, $alias, $proto) = getprotobyname('tcp');

#
# Create arguments for bind() and connect() calls, below.
#   0000 = Wildcard address
#
$thisport = pack($sockaddr, $AF_INET, $port, "\0\0\0\0");


select(NEW_SOCK);  $| = 1;   select(stdout);

#
# Open a network connection via a socket
#
socket(SOCK, $PF_INET, $SOCK_STREAM, $proto) ||
    die "cannot create socket: $!\n";
bind(SOCK, $thisport)                        ||
    die "cannot bind socket: $!\n";
listen(SOCK, SOMAXCONN)                      ||
    die "cannot listen socket: $!\n";


printf "Listening on port %d.\n", $port;
#
#
for ($con = 1;; $con++) {
   #
   # Wait for incomming connections
   #
   $client_addr = accept(NEW_SOCK, SOCK) ||
       die "cannot accept socket: $!\n";

   # We have a connection!
   # printf("Accepted connection #$con!\n");


   #
   # Here we used to call fork() to fork a child process.
   # However, this causes problems if the parent doesn't wait()
   # for the child -- zombie child processes are left behind!
   # To fix this, it might be better to exec() the child process
   # code so that it doesn't wait for the parent.
   # This way, we can handle multipe overlapping messages.
   # Even if we don't fork(), listen() will allow multiple
   # pending connections.
   #
   # if (($child = fork()) == 0) {

   #
   # Get info about incoming connection, and print it
   #
   ($af, $port, $ipaddr) = unpack($sockaddr, $client_addr);
   @ipaddr = unpack('C4', $ipaddr);
   $IP_addr = sprintf("%d.%d.%d.%d",
                      @ipaddr[0], @ipaddr[1], @ipaddr[2], @ipaddr[3]);
   $Date = &ctime(time());  # "Mon Oct 21 21:52:22 PDT 1996\n"
   printf "%d: Got message from %s at %s", $con, $IP_addr, $Date;

   #
   # Read incoming message, and save it to a file
```

```
    # Append it to a file named   data.IP_addr  giving
    # each analyzer its own data file.
    #
    $file = "./data.$IP_addr";
    # print "Routing input to $file.\n";
    open(FILE_OUT, ">> $file") || die "Cannot open $file: $!";
    print FILE_OUT $Date;
    while (<NEW_SOCK>) {
        print FILE_OUT "$_";
    }
    close(NEW_SOCK);
    close(FILE_OUT);

    # Tell the instrument's IBASIC program to continue
    system("lanio $IP_addr 'PROGram:STATe CONT'");

}

exit 0;
```

Controlling the Analyzer using HP VEE

To control your analyzer via LAN using HP VEE, click on the VEE menu titled "I/O." Then select "To/From Socket" and position the I/O object box on the screen. Fill in the following fields:

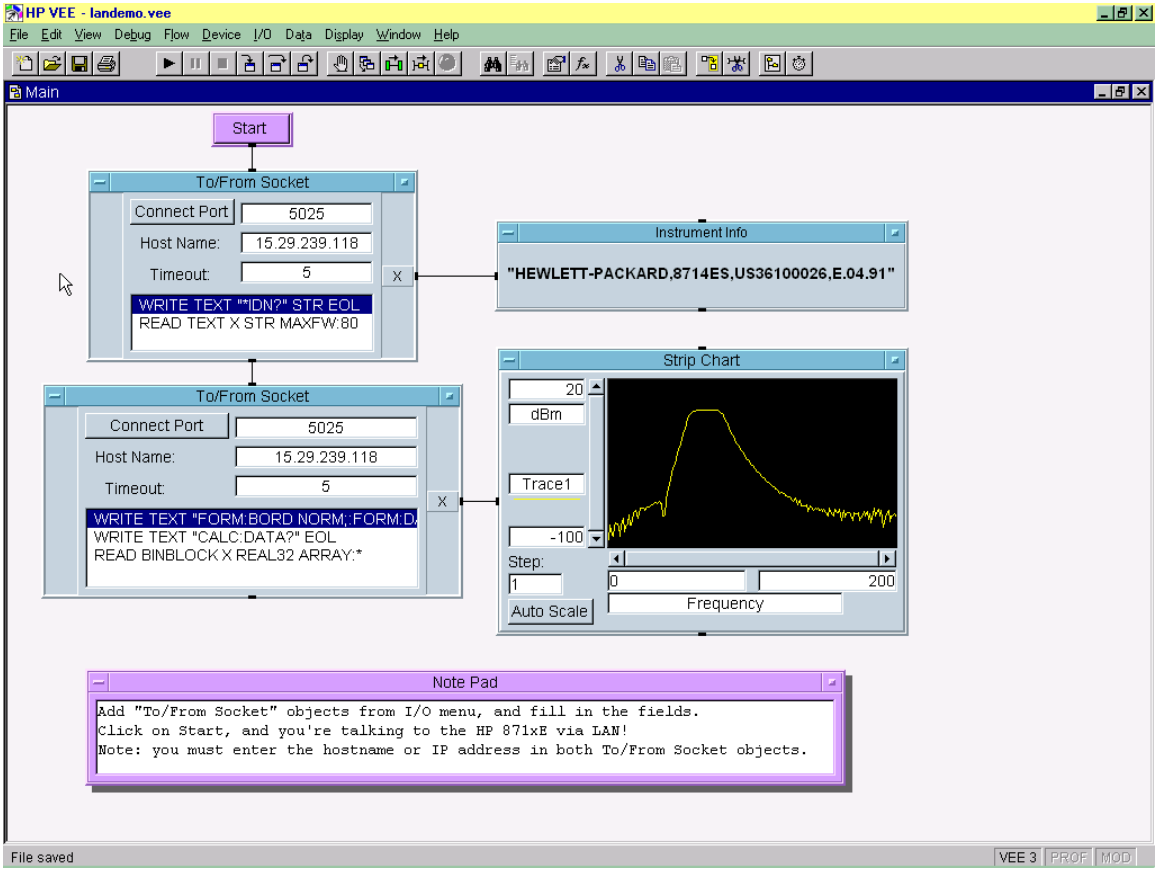
```
Connect Port:   5025
Host Name:      <hostname>
Timeout:        15
```

For faster troubleshooting, you may want to set the timeout to a smaller number. If the hostname you enter doesn't work, try using the IP address of your analyzer (example: `15.4.43.5`). Using the IP address rather than the hostname may also be faster. See Figure 6-1 for an example of an HP VEE screen.

NOTE

If you need to control the GPIB using “device clear” or SRQs, you can use SICL LAN. SICL LAN provides control of your analyzer via GPIB. See “Controlling the Analyzer using SICL LAN” on page 6-42.

Figure 6-1 Sample HP VEE Screen



Controlling the Analyzer with a Java™ Applet

The following example program demonstrates simple socket programming with Java. It is written in Java programming language, and will compile with Java compilers versions 1.0 and above.

This program is on the *Example Programs Disk* that was shipped with your analyzer. Please read the README file on the *Example Programs Disk* before using this program.

```
import java.awt.*;
import java.io.*;
import java.net.*;
import java.applet.*;

// This is a SCPI Demo to demonstrate how one can communicate with the
// HP87xx network analyzer with a JAVA capable browser.  This is the
// Main class for the SCPI Demo.  This applet will need Socks.class to
// support the I/O commands and a ScpiDemo.html for a browser to load
// the applet.

// To use this applet, either compile this applet with a Java compiler
// or use the existing compiled classes.  copy ScpiDemo.class,
// Socks.class and ScpiDemo.html to a floppy.  Insert the floppy into
// your instrument.  Load up a browser on your computer and do the
// following:

//     1. Load this URL in your browser:

//        http://<Your instrument's IP address or name>/int/ScpiDemo.html

//     2. There should be two text windows show up in the browser:
//        The top one is the SCPI response text area for any response
//        coming back from the instrument.  The bottom one is for you
//        to enter a SCPI command.  Type in a SCPI command and hit enter.
//        If the command expects a response, it will show up in the top
//        window.

public class ScpiDemo extends java.applet.Applet implements Runnable {

    Thread       responseThread;

    Socks        sck;

    URL          appletBase;

    TextField    scpiCommand = new TextField();

    TextArea     scpiResponse = new TextArea(10, 60);
```

```
    Panel        southPanel = new Panel();
    Panel        p;
// Initialize the applets
    public void init() {
        SetupSockets();
        SetupPanels();
        // Set up font type for both panels
        Font font = new Font("TimesRoman", Font.BOLD,14);
        scpiResponse.setFont(font);
        scpiCommand.setFont(font);
        scpiResponse.appendText("SCPI Demo Program:  Response messages\n");
        scpiResponse.appendText("-------------------------------------\n");
    }
    // This routine is called whenever the applet is actived
    public void start() {
        // Open the sockets if not already opened
        sck.OpenSockets();
        // Start a response thread
        StartResponseThread(true);
    }
    // This routine is called whenever the applet is out of scope
    // i.e. minimize browser
    public void stop() {
        // Close all local sockets
        sck.CloseSockets();
        // Kill the response thread
        StartResponseThread(false);

    }
    // Action for sending out scpi commands
```

```
// This routine is called whenever a command is received from the
// SCPI command panel.
public boolean action(Event evt, Object what) {
    // If this is the correct target
    if (evt.target == scpiCommand) {
        // Get the scpi command
        String str = scpiCommand.getText();
        // Send it out to the Scpi socket
        sck.ScpiWriteLine(str);
        // Query for any error
        sck.ScpiWriteLine("syst:err?");
        return true;
    }
    return false;
}
// Start/Stop a Response thread to display the response strings
private void StartResponseThread(boolean start) {
    if (start) {
        // Start a response thread
        responseThread = new Thread(this);
        responseThread.start();
    }
    else {
        // Kill the response thread
        responseThread = null;
    }
}
// Response thread running
public void run() {
    String str = "";  // Initialize str to null
```

```
    // Clear the error queue before starting the thread
    // in case if there's any error messages from the previous actions
    while ( str.indexOf("No error") == -1 ) {
        sck.ScpiWriteLine("syst:err?");
        str = sck.ScpiReadLine();
    }

    // Start receiving response or error messages
    while(true) {
        str = sck.ScpiReadLine();
        // If response messages is "No error", do no display it
        if ( str.indexOf("No error") == -1 ) {
            // Display the error message in the Response panel
            scpiResponse.appendText(str+"\n");
            // Query for any error messages
            sck.ScpiWriteLine("syst:err?");
        }
    }
}
// Set up and open the SCPI sockets
private void SetupSockets() {
    // Get server url
    appletBase = (URL)getCodeBase();
    // Open the sockets
    sck = new Socks(appletBase);
}
// Set up the SCPI command and response panels
private void SetupPanels() {
    // Set up SCPI command panel
    southPanel.setLayout(new GridLayout(1, 1));
```

```
        p = new Panel();
        p.setLayout(new BorderLayout());
        p.add("West", new Label("SCPI command:"));
        p.add("Center", scpiCommand);
        southPanel.add(p);
        // Set up the Response panel
        setLayout(new BorderLayout(2,2));
        add("Center", scpiResponse);
        add("South", southPanel);
    }
}
//  Socks class is responsible for open/close/read/write operations
//  from the predefined socket ports.  For this example program,
//  the only port used is 5025 for the SCPI port.
class Socks extends java.applet.Applet {
    // Socket Info
    // To add a new socket, add a constant here,
    // change MAX_NUM_OF_SOCKETS
    // then, edit the constructor for the new socket.
    public final int SCPI=0;
    private final int MAX_NUM_OF_SOCKETS=1;
    // Port number
    // 5025 is the dedicated port number for HP8711's SCPI port
    private final int SCPI_PORT = 5025;
    // Socket info
    private URL appletBase;
    private Socket[] sock = new Socket[MAX_NUM_OF_SOCKETS];
    private DataInputStream[] sockIn=new DataInputStream[MAX_NUM_OF_SOCKETS];
    private PrintStream[] sockOut = new PrintStream[MAX_NUM_OF_SOCKETS];
    private int[] port = new int[MAX_NUM_OF_SOCKETS];
```

```
private boolean[] sockOpen = new boolean[MAX_NUM_OF_SOCKETS];
// Constructor
Socks(URL appletB)
{
    appletBase = appletB;
    // Set up for port array.
    port[SCPI] = SCPI_PORT;

    // Initialize the sock array
    for ( int i = 0; i < MAX_NUM_OF_SOCKETS; i++ ) {
        sock[i] = null;
        sockIn[i] = null;
        sockOut[i] = null;
        sockOpen[i] = false;
    }
}
//***** Sockects open/close routines
// Open the socket(s) if not already opened
public void OpenSockets()
{
    try {
        // Open each socket if possible
        for ( int i = 0; i < MAX_NUM_OF_SOCKETS; i++ ) {
            if ( !sockOpen[i] ) {
                sock[i] = new Socket(appletBase.getHost(),port[i]);
                sockIn[i]=new DataInputStream(sock[i].getInputStream());
                sockOut[i]=new PrintStream(sock[i].getOutputStream());
                if ( (sock[i] != null) && (sockIn[i] != null) &&
                     (sockOut[i] != null) ) {
                    sockOpen[i] = true;
```

```
                }
            }
        }
    }
    catch (IOException e) {
        System.out.println("Sock, Open Error "+e.getMessage());
    }
}
// Close the socket(s) if opened
public void CloseSocket(int s)
{
    try {
        if ( sockOpen[s] == true ) {
            // write blank line to exit servers elegantly
            sockOut[s].println();
            sockOut[s].flush();
            sockIn[s].close();
            sockOut[s].close();
            sock[s].close();
            sockOpen[s] = false;
        }
    }
    catch (IOException e) {
        System.out.println("Sock, Close Error "+e.getMessage());
    }
}
// Close all sockets
public void CloseSockets()
{
    for ( int i=0; i < MAX_NUM_OF_SOCKETS; i++ ) {
```

```
            CloseSocket(i);
        }
    }
    // Return the status of the socket, open or close.
    public boolean SockOpen(int s)
    {
        return sockOpen[s];
    }
    //************* Socket I/O routines.
//*** I/O routines for SCPI socket

    // Write an ASCII string with carriage return to SCPI socket
    public void ScpiWriteLine(String command)
    {
        if ( SockOpen(SCPI) ) {
            sockOut[SCPI].println(command);
            sockOut[SCPI].flush();
        }
    }
    // Read an ASCII string, terminated with carriage return
    // from SCPI socket
public String ScpiReadLine()
    {
        try {
            if ( SockOpen(SCPI) ) {
                return sockIn[SCPI].readLine();
            }
        }
        catch (IOException e) {
            System.out.println("Scpi Read Line Error "+e.getMessage());
        }
```

```
    return null;
}
// Read a byte from SCPI socket
public byte ScpiReadByte()
{
    try {
        if ( SockOpen(SCPI) ) {
            return sockIn[SCPI].readByte();
        }
    }
    catch (IOException e) {
        System.out.println("Scpi Read Byte Error "+e.getMessage());
    }
    return 0;
}
              }
```

Controlling the Analyzer using SICL LAN

SICL LAN is a LAN protocol using the Standard Instrument Control Library (SICL). It provides control of your analyzer over the LAN, using a variety of computing platforms, I/O interfaces, and operating systems. With SICL LAN, you control your remote analyzer over the LAN with the same methods you use for a local analyzer connected directly to the controller with the GPIB.

Your analyzer implements a SICL LAN *server*. To control the analyzer, you need a SICL LAN *client* application running on a computer or workstation that is connected to the analyzer over a LAN. Typical applications implementing a SICL LAN client include:

- HP VEE
- HP BASIC
- National Instrument's LabView with HP VISA/SICL client drivers

NOTE

The SICL LAN protocol is Agilent's implementation of the VXI-11 Instrument Protocol, defined by the VXIbus Consortium working group.

At the time of the publication of this manual, National Instruments' VISA does not support the VXI-11 Instrument Protocol. However, future revisions of National Instruments' VISA will support the VX-11 protocol. Contact National Instruments for their release date.

SICL LAN can be used with Windows 95, Windows 98, Windows NT, and HP-UX.

Collecting SICL LAN Setup Information

Before you set up your analyzer as a SICL LAN server, you will need to collect some information about your HP VISA/SICL LAN client application. Record the following parameters from your HP VISA/SICL LAN client application after you have set it up:

GPIB name	The GPIB name is the name given to a device used to communicate with the analyzer. `hpib` and `gpib` are typical GPIB names. Your analyzer is shipped with `gpib` as the GPIB name. The GPIB name is the same as the remote SICL address.
GPIB logical unit	The logical unit number is a unique integer assigned to the device to be controlled using SICL LAN. Your analyzer is shipped with the logical unit number set to 7. Numbers 0 through 30, excluding 21, are valid logical unit numbers for your analyzer. Logical unit number 21 is used for the analyzer's internal emulation mode. (If you are using HP VEE and SICL LAN, the logical unit number is limited to the range of 0-8.)
GPIB device address	The device address is the GPIB device address (bus address) assigned to the device to be controlled using SICL LAN. Your analyzer is shipped with the GPIB device address set to 18. You can enter any address from 0 to 1024 as an GPIB address.

The SICL LAN server uses the GPIB name, GPIB logical unit number, and GPIB address configuration on the SICL LAN client to communicate with the client. You must match these parameters *exactly* when you set up the SICL LAN client and server.

Configuring Your Analyzer as a SICL LAN Server

After you have collected the required information from the SICL LAN client, perform the following steps to set up your analyzer as a SICL LAN server:

1. Enter the GPIB name

   Press (SYSTEM OPTIONS) **LAN** **SICL LAN Setup** **GPIB Name**, enter the GPIB name from the SICL LAN client, and press **ENTER**. Press **Clear Entry** if you need to replace the existing entry.

2. Enter the GPIB logical unit number

   Press **GPIB Log. Unit**, enter the GPIB logical unit number from the SICL LAN client, and press **ENTER**.

3. Enter the GPIB device address

   Press **GPIB Dev. Address**, enter the GPIB device address from the SICL LAN client, and press **ENTER**.

   If you want to restore the default settings, press **Restore Defaults**.

4. Turn the analyzer off, then on.

Configuring Your PC as a SICL LAN Client

The descriptions here are based on HP's VISA revision G.02.02, model number HP2094G. A copy of HP's VISA can be found on the following website:

`http://www.tm.agilent.com/tmo/software/English/HP_IO_Libraries.html`.

These descriptions assume a LAN connection between your computer and network analyzer. They are not written for the GPIB-to-LAN gateway.

1. Install HP VISA revision G.02.02 or higher.
2. Run I/O configuration.
3. Select LAN Client from the Available Interface Types.
4. Press Configure.

5. Enter an interface name, such as lan1.
6. Enter a logical unit number, such as 7.
7. Select OK.
8. Select VISA LAN Client from the Available Interface Types.
9. Press Configure.
10. Enter a VISA Interface name, such as GPIB1.
11. Enter the hostname or IP address of your analyzer in the Hostname field, such as my8712.hp.com
12. Enter a remote SICL address, such as gpib1.
13. Set the LAN interface to match the defined LAN client (lan1 in this example).
14. Select OK.
15. Close I/O Configuration by selecting OK.

Controlling Your Analyzer with SICL LAN and HP VEE

Before you can use SICL LAN with HP VEE, you need to set up HP VISA/SICL LAN I/O drivers for use with your HP VEE application. Consult your HP VEE documentation for information how to do this.

NOTE

If you are using HP VEE and SICL LAN, the logical unit number is limited to the range of 0–8.

The logical unit number is the same as the interface select code (ISC). HP VEE reserves ISC values 9--18, and does not allow you to use them for SICL/LAN communications with your analyzer. HP VEE also does not allow any ISC values higher than 18.

After you have the VISA/SICL LAN I/O drivers installed, perform the steps below to set up HP VEE to control your analyzer:

1. On your computer or workstation, select `I/O|Instrument Manager`.
   (See Figure 6-2.)
2. Add a new GPIB device with an address of 7XX, where XX is the GPIB device address from your analyzer.

Figure 6-2 I/O | Instrument Manager Menu

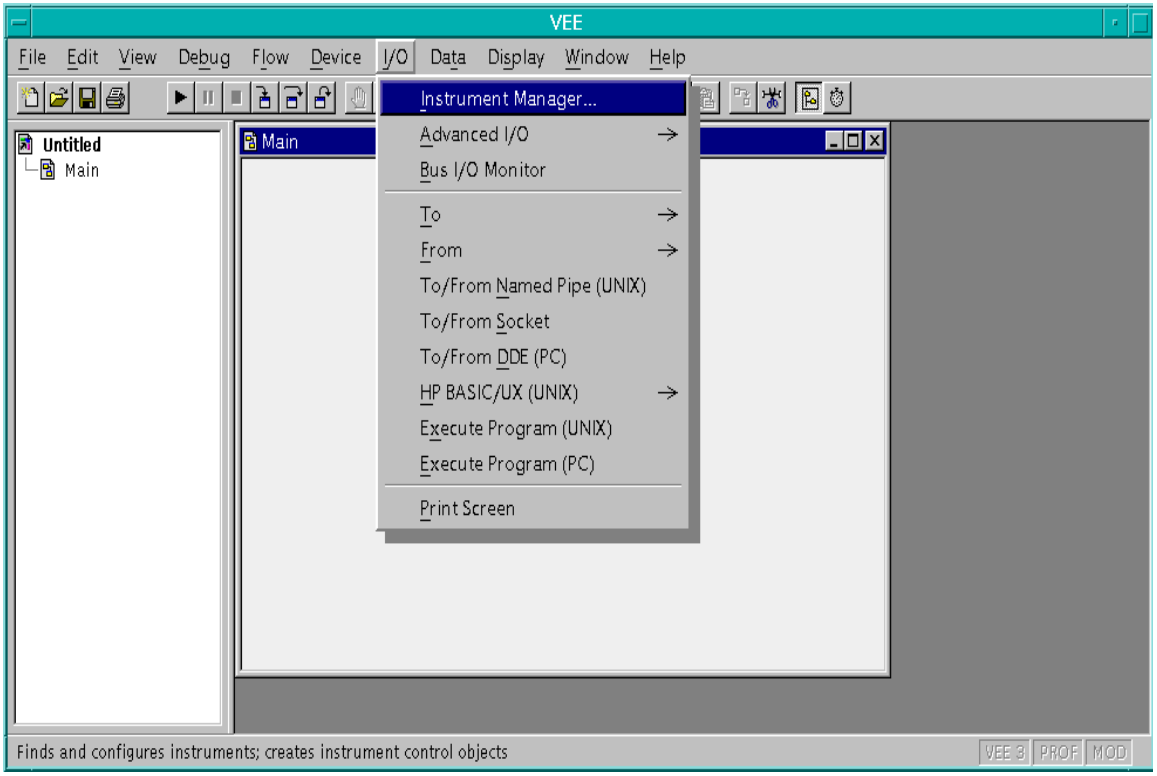
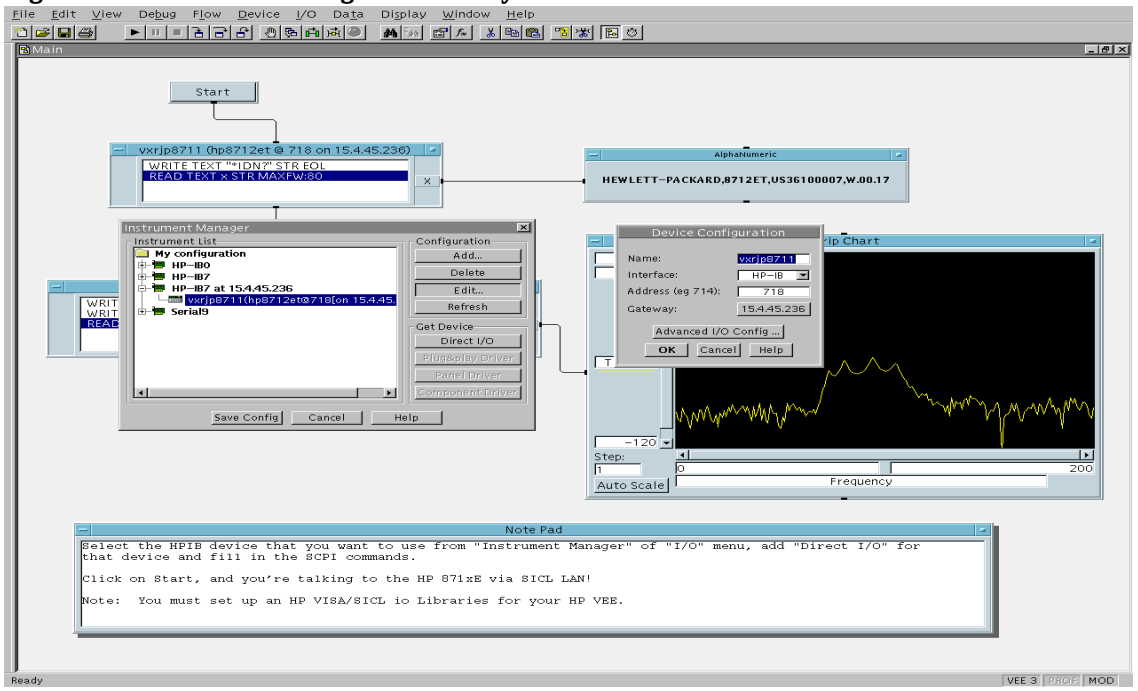
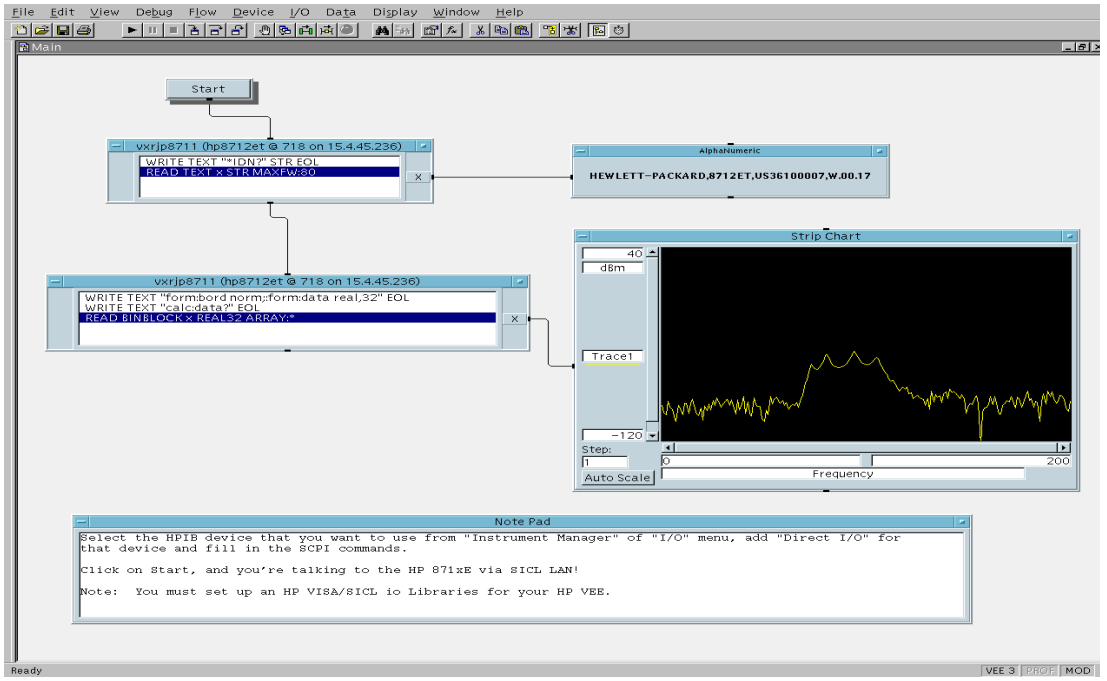


Figure 6-3 Adding Your Analyzer as an HP VEE Device



To send SCPI commands to the analyzer, select I/O | Instrument Manager, and the GPIB device just added. Select Direct I/O. You can now type SCPI commands in the command window, and they will be sent over the LAN to your analyzer.

Figure 6-4 Sending SCPI Commands Directly to Your Analyzer



See the HP VEE example program for more details.

Controlling Your Analyzer with SICL LAN and HP BASIC for Windows

Before you can use HP BASIC for Windows with SICL LAN, you need to set up HP VISA/SICL LAN I/O drivers for use with your HP BASIC applications. Consult your HP BASIC documentation for information how to do this.

To set up SICL LAN for HP BASIC, add the following statement to the AUTOST program on your PC (all on a single line):

```
LOAD BIN "HPIBS;DEV lan[analyzer IP address]:HP-IB name
TIME 30 ISC 7"
```

Replace `analyzer IP address` with the IP address of your analyzer, `HP-IB name` with the GPIB name given to your analyzer, and 7 with the logical unit number.

For example, the following `LOAD` statement should be added to your `AUTOST` program for the parameters listed below:

analyzer IP address	**12.22.344.225**
analyzer GPIB name	**test02**
logical unit number	**7**
timeout value (seconds)	**30**

`LOAD` statement (all on a single line)

```
LOAD BIN "HPIBS;DEV lan[12.22.344.225]:test02 TIME 30
ISC 7"
```

Consult your HP BASIC documentation to learn how to load the SICL driver for HP BASIC.

After the SICL driver is loaded, you control your analyzer using commands such as the following:

```
OUTPUT 718; “*IDN?”
ENTER 718; S$
```

where 18 is the device address for the analyzer.

See the HP BASIC example program in this chapter for more information.

Controlling Your Analyzer with SICL LAN and HP BASIC for UNIX (Rocky Mountain BASIC)

Before you can use HP Rocky Mountain Basic (HP RMB) with SICL LAN, you will need to set up the HP SICL LAN I/O drivers for HPRMB. Consult your system administrator for details.

Create a `.rmbrc` file in your root directory of your UNIX workstation with the following entries:

```
SELECTIVE_OPEN=ON
Interface 8= “lan[analyzer IP address]:HPIB name”;NORMAL
```

Replace `analyzer IP address` with the IP address of your analyzer, and `GPIB name` with the GPIB name given to your analyzer. Also replace the “8” of `Interface 8` with the logical unit number. Consult your HPRMB documentation for the exact syntax.

After your SICL driver is configured correctly on your UNIX workstation, you control your analyzer using commands such as the following:

```
OUTPUT 818; “*IDN?”
ENTER 818; S$
```

where 18 is the device address for the analyzer.

7 Using the Network File System (NFS)

About This Chapter

This chapter provides a short introduction to the network file system (NFS), and describes how to configure your analyzer to use NFS. The following topics are included:

- Introduction to NFS
- Setting Up NFS
- Using NFS Automount—Connecting to Network Resources Automatically

Introduction to NFS

Network file system (NFS) is a client/server application that provides access to remote[1] files and directories using the LAN. With NFS, remote files and directories behave like local[2] files and directories. The remote file system can be used from your analyzer's Save/Recall menu as if it were a local device. The remote file system can be part of a PC, workstation, or other computing device.

NFS allows you to save test data from your analyzer directly to a remote directory; the remote machine does not have to initiate an ftp session to retrieve the data. This can make saving data in an automated environment, with many analyzers running independently, very easy. As a result, NFS simplifies central management of analyzer files and directories.

NFS requires an NFS *server* and an NFS *client*. The server is a computer that makes its local file system available to NFS clients, using a process called **exporting** (or sharing). An NFS client is a computer that uses the file system made available by the NFS server using a process called **mounting**. The file system that is *exported* by the NFS server is *mounted* by the NFS client and behaves like part of the client's local file system. When using NFS, your analyzer comprises the NFS client part of the Network File System. You must provide a workstation, PC or other computer to serve as the NFS server, and it must include an NFS server application.

1. *Remote* files and directories are part of a file system different than the analyzer file system—they are stored remotely in a computer.
2. *Local* files and directories are part of the analyzer file system—they are stored locally within the analyzer.

NFS Protocols

Current implementations of NFS use transmission control protocol (TCP) as the transport protocol[1] over the network. TCP is a reliable protocol designed to provide guaranteed data delivery. Your analyzer uses TCP/IP over the LAN.

NFS also uses remote procedure call (RPC) protocol. RPC is a client/server protocol providing remote services to a local application (program). The local application can request and execute a procedure on a remote machine, pass data to the remote procedure, and retrieve data from the remote procedure. RPC is a key component of the NFS.

1. The transport protocol governs how data is transmitted over a network.

Setting Up NFS

Configuring the Analyzer as an NFS Client

NFS Fundamentals

Your analyzer implements the client side of the network file system (NFS). As an NFS client, your analyzer accesses the remote[1] file system by *mounting* the remote file system. If the remote file system is mounted successfully, it can be accessed from your analyzer's Save/Recall menu as if it were a local[2] device.

To configure the NFS client on your analyzer, you will need to set up an NFS server on the remote system. NFS servers are readily available with most UNIX operating systems. To use NFS with a PC, you will need NFS server software for the PC. Consult your network administrator about obtaining an NFS server for your PC.

NFS uses remote procedure call (RPC) authentication for file access permissions. To be able to read and write to your remote file system, you must enter a user ID and a group ID for your analyzer. A default user ID (2001) and a default group ID (100) are assigned for you before the analyzer is shipped. If the defaults do not work for you, consult your network administrator about obtaining a user ID and group ID.

1. *Remote* files and directories are part of a file system different than the analyzer file system—they are stored remotely in a computer.
2. *Local* files and directories are part of the analyzer file system—they are stored locally within the analyzer.

Preliminary Requirements

Before setting up the NFS client on your analyzer, do the following:

- Set up an NFS server on the remote system. Consult your system administrator if you are unsure how to do this.
- Collect the following information required to configure your analyzer as the NFS client:

  ✓ The host name and IP address for the remote system (the server).

  For example: `host1` and `123.046.025.221`

  ✓ The name of your remote file system or subdirectory.

  For example: `/users/yourname/na_setups`

  ✓ The user ID and the group ID to allow access to the remote host file system.

  For example: `user ID = 2001, group ID = 100`

  ✓ A local name to assign to the remote file system. Choose a local name which is short and easy to remember. Append a colon (:) or some other special character if you need to distinguish a device name from a subdirectory name on your Save/Recall menu.

  For example: `c:\nfs\setups`

NOTE You may find it convenient to use an external keyboard when performing the following procedure.

Mounting a Remote Host File System

The file system that is exported by the NFS server is mounted by the NFS client, and behaves like part of the local file system. Perform the following procedure to set up your analyzer to mount to a remote host file system:

1. Press (SYSTEM OPTIONS) **LAN** **NFS Device Setup**.
2. Press **Mount NFS Device**.
3. Type the remote host IP address or remote hostname in the dialog box displayed on the screen.
4. Press **Enter** to confirm your entry when you are done.

NOTE In order to use a remote hostname , you must set up a HOSTS file on your local non-volatile RAM Disk. A HOSTS file is not required to use a host IP address. See "Using a Local HOSTS File" on page 7-11 for details.

5. Type the name of the remote host file system or subdirectory in the dialog box now displayed.
6. Press **Enter** to confirm your entry when you are done.

NOTE All NFS directory names and file names are case-sensitive. This is unlike the local analyzer file system where directory and file names are not case-sensitive.

7. If required, press **Clear Entry** and type in a new name for the default local file system name provided.
8. Press **Enter** when you are done changing the name, or if the default name is acceptable.

   To retype the file system names, first press [Cancel], then press [Remote IP Addr/Host] to start from the beginning.

   At this point, the analyzer will attempt to mount the remote file system. If the remote file system is successfully mounted, the analyzer will display

   ```
   NFS Mount to <host> successful
   ```

   where <host> is the remote host name or IP address.

NOTE You can mount up to seven remote NFS file systems as servers. An error will occur if you try to mount more than seven devices.

NOTE When you press **Enter** on the **NFS Setup** entry line without inputting a name or address, the analyzer will attempt to mount your NFS device using the existing entries.

NOTE If the local file system name is empty because **Clear Entry** has been used, or if **Enter** was pressed without changing the default name, the remote file system name will be used as the local file system name.

9. If the remote file system is mounted successfully, you can press [Automount At Powerup] to mount the remote file system automatically on powerup. You can also set up automount devices in the [Automount Setup] menu. See "Using NFS Automount—Connecting to Network Resources Automatically" on page 7-13 for details.

Confirming Remote File System Mounting

To find out if a remote file system has been successfully mounted, press

SYSTEM OPTIONS **LAN** **NFS Device Setup** **NFS Device Table**

The analyzer will display an NFS device table containing three columns (left to right):

1. Device number, in the order mounted
2. Name of the local file system
3. Remote file system name and IP address

Press **Local Path** to view an expanded version of your local file system.

Press **Remote Path** to view an expanded version of your remote file system.

Setting up NFS Client Authentication IDs

Perform the following steps to set up authentication IDs for your NFS client:

1. Press SYSTEM OPTIONS **LAN** **NFS Device Setup** **Authentication** .
2. Press **User ID** and type in the user ID for your remote file system.
3. Press **Group ID** and type in the group ID for your remote file system.

NOTE

Some systems use only the user ID. If your remote file system does not use a group ID, use the default Group ID value.

Unmounting a Remote File System

Perform the following steps to unmount a remote file system from your analyzer:

1. Press SYSTEM OPTIONS **LAN** **Unmount NFS Device**.

NOTE If the displayed NFS device table is empty, there are no mounted file systems to unmount.

2. Select the device that you want to unmount and press the softkey corresponding to that device.

If the device is successfully unmounted, the device will be removed from the displayed NFS device table. If the NFS device is not present, an error message will be displayed after the analyzer times out, and the device will be removed from the NFS device table. The device will also be removed from the NFS device table if it is unmounted unsuccessfully due to disconnected links.

NOTE The NFS device table always compacts itself and refreshes the displayed information after a successful unmount.

Using a Local HOSTS File

You can add one or more host names of other network devices to a local HOSTS file. This file associates host IP addresses with host names, so that you can use the host name instead of the host IP address to mount and automount Network File System (NFS) devices.

On powerup, your analyzer will load the local HOSTS file from non-volatile RAM. You can then use the host name in SYSTEM OPTIONS **LAN** **NFS Device Setup** and SYSTEM OPTIONS **LAN** **NFS Device Setup** **Automount Dev Setup** .

Creating and Saving a Local HOSTS File

You can create a local HOSTS file using an ASCII text editor on your computer or workstation. Save the file locally, and transfer it to your analyzer's non-volatile RAM disk using a 3.5" disk, FTP or NFS. Refer to "Using FTP to Access the Analyzer" on page 4-3 for information about disk or FTP file transfers.

HOSTS File Format Rules

The HOSTS file is an ASCII text file formatted according to the following rules:

1. The file name must be HOSTS, with no file extension.
2. Any combination of upper case and lower case letters can be used in the file name. The analyzer file system is case-insensitive, and will accept and use files saved with the name HOSTS, regardless of the case of the letters originally in the name.
3. Each IP address and host name pair must be on a single line with the IP address first and the corresponding host name next.
4. The IP address and the corresponding host name must be separated by at least one space character.
5. Optional comments can be included, and must begin with a "#" character.

The following is an example of a valid HOSTS file:

```
#
#
# This is a sample hosts file
#
#
#

15.4.45.232        host1       # John Doe's workstation
15.4.45.233        host2       # Jane Doe's PC
```

If you place the above HOSTS file in non-volatile RAM and power cycle your analyzer, you can use the name `host1` to mount an NFS device by name instead of the IP address `15.4.45.232`.

NOTE

The HOSTS file is loaded only at power-up. If you add new entries to the file, you will have to cycle power to the instrument in order for the new entries to take effect.

Using NFS Automount—Connecting to Network Resources Automatically

NFS Automount allows your analyzer to automatically mount one or more NFS devices on power-up without user intervention.

Adding Devices to the Automount Device Table

NOTE

Adding an NFS device to the automount device table does not guarantee a successful mount. To ensure a working NFS mount, add NFS devices to the automount list only after a successful test of the mount process. See "Configuring the Analyzer as an NFS Client" on page 7-5 for details.

Follow the steps below to add a device to the automount device table:

1. Press (SYSTEM OPTIONS) **LAN** **NFS Device Setup** **Automount Setup**.
2. Press **Remote IP Addr/Host** and enter the remote host name or the IP address in the provided dialog box.

NOTE

Designating an automount NFS host by name requires a local HOSTS file. See "Using a Local HOSTS File" on page 7-11 for details.

3. Press **Enter** to confirm your entry.
4. Press **Remote Path** and enter the remote host file system name in the provided dialog box.
5. Press **Enter** to confirm your entry.
6. Press **Local Path**. The remote file system name is provided as the default local file system name.
   a. Press **Enter** to use the *default* local file system name.
   b. To use your own local file system name, press **Clear Entry** and enter a unique local file system name. Press **Enter** to confirm your entry.

NOTE

If the local file system name is NULL or is an empty string (entered by pressing **Clear Entry** and **Enter** without typing anything), the remote file system name will be used as the local file system name.

7. Press **Automount At Powerup** to add the NFS file system to the automount device table.

NOTE

Pressing **Automount At Powerup** does not trigger a NFS mount process. See "Configuring the Analyzer as an NFS Client" on page 7-5 to mount NFS devices.

Verifying Automount Entries

Perform the following steps to verify that the NFS device has been successfully added to the automount device table:

1. Press SYSTEM OPTIONS **LAN NFS Device Setup Automount Setup**.
2. Press **Automount Dev Table** to display the automount device table.

   The automount device table has three columns. The first column lists the device numbers. The second column lists the given name for your local file system. The third column lists remote file system names and the remote host IP addresses. The lines in the automount device table are listed in the order they were entered.
3. Press **Local Path** to verify the local file system information.
4. Press **Remote Path** to verify the remote file system information.

Removing an NFS Device from the Automount Table

Perform the following steps to remove an NFS device from the automount table:

1. Press SYSTEM OPTIONS **LAN NFS Device Setup Automount Setup**.
2. Press **Remove Automount** . The automount device table will be displayed on your screen.

   The numbers on the automount device table match the device numbers displayed on the softkeys. If the table is empty, you do not have any NFS devices in the automount device table.
3. Select the device that you want to remove from the table and press the softkey which corresponds to that device.

NOTE

The automount device table always compacts itself after a successful removal.

Using Save/Recall with NFS

NFS Fundamentals

To access file systems with NFS, you will have to set up at least one NFS device. See "Configuring the Analyzer as an NFS Client" on page 7-5 in this chapter for details on how to do this. The following procedures assume you have already set up at least one NFS device.

Selecting a Remote Device

To save or recall a state, a program, or measurement data to or from an NFS device, you must first select an NFS device from the [Select Disk] menu. Since NFS provides transparent access to the remote file system, the procedure for saving or recalling an instrument state or program *remotely* is the same as that used to save or recall an instrument state or program *locally*. See Chapter 4, "Accessing the Analyzer's File System Using FTP," as well as "Copying Programs to and from the Analyzer" on page 5-7 and "Saving and Recalling Analyzer States" on page 5-5 for information on how to save or recall a state or program to a local device. Also see "Saving and Recalling Measurement Results" in Chapter 4 of the User's Guide for your analyzer.

To select an NFS device do the following:

1. Press SAVE RECALL **Select Disk** **NFS Device** .

   The analyzer displays a two-column NFS device table. The first column contains the device numbers which correspond to the device numbers on the front panel softkeys. The second column contains the local path or device names for the NFS devices.

2. Choose an NFS device to use for save/recall and press the corresponding front panel softkey. If the remote NFS device you selected is still available, the analyzer will display the directory for your remote NFS device.

NOTE

The analyzer will report a disk error and switch back to **Non-Vol RAM Disk** if there is a network problem, or if the remote device is not available. You will have to set up the NFS device again before using it with save/recall.

Some of the remote files may show the word unknown in the file attributes column. This may be due to the lack of appropriate file access permissions. NFS authentication must be set up correctly for you to have permission to access certain files on your remote NFS device. To make sure that all required files are accessible from your analyzer, confirm that your NFS authentication user ID and group ID match the corresponding IDs on the remote system. See "Configuring the Analyzer as an NFS Client" on page 7-5 for details.

Copying Files to a Remote NFS Device

Perform the following steps to copy files from a local device to a remote NFS device:

1. Press SAVE RECALL **Select Disk** .
2. Choose a local device that you want to copy files from and select it by pressing the corresponding softkey.
3. Press SAVE RECALL **File Utilities** to use the file utilities menu.
4. Select the file that you want to copy.
5. Press **Copy File** to copy the selected file or [Copy All Files] to copy all the files in the current directory.
6. Press **Copy to NFS Device** to copy the file or files to an NFS device. A dialog box with the currently selected NFS device and selected file name is displayed on top of the screen.
7. Press **Enter** if the destination device and destination file name is correct. Otherwise, enter the correct destination file path and press **Enter** .

   The device name and file name are case-sensitive, and you can append the file name with either a '/' or '\' between the device name and the file name.

   If you encounter a file access error, make sure that your network and the remote system are working correctly and that the NFS authentication IDs are set up correctly.

NOTE The NFS device name for the destination must match the NFS path name used when the NFS device was mounted.

Copying Files from a Remote NFS Device

Perform the following steps to copy files from a remote NFS device to a local device:

NOTE Your analyzer does not support file copy from a remote NFS device to another remote NFS device.

1. Press SAVE RECALL **Select Disk** **NFS Device**.
2. Choose an NFS device that you want to copy files from and select it by pressing the corresponding softkey.
3. Press SAVE RECALL **File Utilities** to use the file utilities menu.
4. Select the file that you want to copy.
5. Press **Copy File** to copy the selected file or [Copy All Files] to copy all the files in the current directory.
6. Press **Copy to NonVol RAM**, **Copy to Vol RAM**, or **Copy to 3.5" Disk** to copy the file or files to the desired local device. A dialog box with the currently selected local device and selected file name is displayed on top of the screen.
7. Press **Enter** if the destination device and destination file name is correct. Otherwise, enter the correct destination file path and press **Enter**.

   The device name and file name are case-sensitive, and you can append the file name with either a '/' or '\' between the device name and the file name.

   If you encounter a file access error, make sure that your network and the remote system are working correctly and that the NFS authentication IDs are set up correctly.

8 General Troubleshooting

About This Chapter

This chapter provides troubleshooting information for the LAN interface. It has four sections:

Troubleshooting the Initial Connection

Getting the analyzer to work with your network often requires detailed knowledge of your local network software. This section attempts to help you with some common problems. Contact your network administrator for additional assistance.

Assess the Problem

The analyzer LAN interface does not need or include any proprietary driver software. It was designed to operate with common network utilities and drivers.

Either a hardware problem or a software problem can prevent the analyzer's remote file server from communicating over the LAN. The following common problems may be encountered:

Timeout Errors

Timeout errors such as "Device Timeout," "File Timeout," and "Operation Timeout," are symptoms of one or both of the following problems:

- The currently configured timeout limits are too short compared to the time it takes the LAN to complete some operations. This problem may occur during periods of increased LAN traffic.
- The LAN connection has failed, or fails occasionally.

To increase your timeout period, refer to your computer documentation for instructions. Contact your LAN administrator if problems continue.

Packets Routinely Lost

If packets are routinely lost, proceed to the troubleshooting section in this chapter relating to your network.

Problems Transferring or Copying Files

If you have problems copying files out of or into the analyzer, you might be experiencing timeout problems. See the previous section on "Timeout Errors."

Communications Not Established

If you have just installed and configured the LAN interface and you have never been able to access the analyzer via ftp or telnet, go directly to "Ping the Analyzer from Your Computer or Workstation" on page 8-5.

If you have previously been able to access the analyzer via ftp or telnet and now cannot do so, check the following:

- Has any hardware been added or moved on your network? This includes adding or removing any workstations or peripherals, or changing any cabling.
- Have software applications been added to the network?
- Have any configuration files been modified?
- Have any of the following files been deleted or overwritten?

  ✓ UNIX:

  ❑ /etc/hosts

  ❑ /etc/inetd.conf

  ❑ /etc/services

  ✓ PCs:

  ❑ dependent network files

If you know or suspect that something has changed on your network, consult with your network administrator.

Ping the Analyzer from Your Computer or Workstation

Verify the communications link between the computer and the analyzer remote file server using the ping utility.

From a UNIX workstation, type:

```
ping hostname 64 10
```

where 64 is the packet size, and 10 is the number of packets transmitted.

From a DOS or Windows environment, type:

```
ping hostname 10
```

where 10 is the number of echo requests.

Normal Response for UNIX

A normal response to the ping will be a total of 9, 10, or possibly 11 packets received with a minimal average round-trip time. The minimal average will be different from network to network. LAN traffic will cause the round-trip time to vary widely.

Because the number of packets received depends on your network traffic and integrity, the normal number might be different for your network.

Normal Response for DOS or Windows

A normal response to the ping will be a total of 9, 10, or possibly 11 packets received if 10 echo requests were specified.

Because the number of packets received depends on your network traffic and integrity, the normal number might be different for your network.

Error Messages

- If error messages appear, then check the command syntax before continuing with the troubleshooting. If the syntax is correct, then resolve the error messages using your network documentation, or by consulting your network administrator.
- If an unknown host error message appears, then check the node names database to see that the hostname and IP address for your analyzer are correctly entered.

No Response

No packets received indicates no response from a ping.

- If there is no response, try typing in the IP address with the ping command, instead of using the hostname. Check that the typed address matches the IP address assigned in the **LAN Port Setup** menu, then check the other addresses in the menu.
- Check that the hostname and IP address are correctly entered in the node names database.
- If you are using a UNIX environment, ping each node along the route between your workstation and the analyzer, starting with the your workstation. Ping each gateway, then attempt a ping of the remote file server.
- If the analyzer still does not respond to ping, then you should suspect a hardware problem with the analyzer. To check the analyzer performance, refer to the analyzer's *Service Guide*.
- Make sure that you have used the appropriate cable to connect your analyzer to the LAN. A patch cable can be used to connect your analyzer to a hub, or a cross-over cable can be used for direct analyzer–PC links.

Intermittent Response

If you received 1 to 8 packets back, there is probably a problem with the network. Because the number of packets received depends on your network traffic and integrity, the number might be different for your network.

Use a LAN analyzer or LAN management software to monitor activity and determine where bottlenecks or other problems are occurring. The analyzer will still function, but communications over the LAN will be slower.

On a single-client/single-server network, the most likely cause of intermittent response to an echo request is a hardware problem with the LAN module installed in the PC, the cable, or the analyzer. To check the analyzer performance, refer to the analyzer's *Service Guide*.

Ping Your Computer or Other Device from Your Analyzer

The last section helped you verify connectivity from your computer to your analyzer. This section helps you verify the connectivity path in the opposite direction — from your analyzer to your computer.

To Use the Built-In Ping Utility

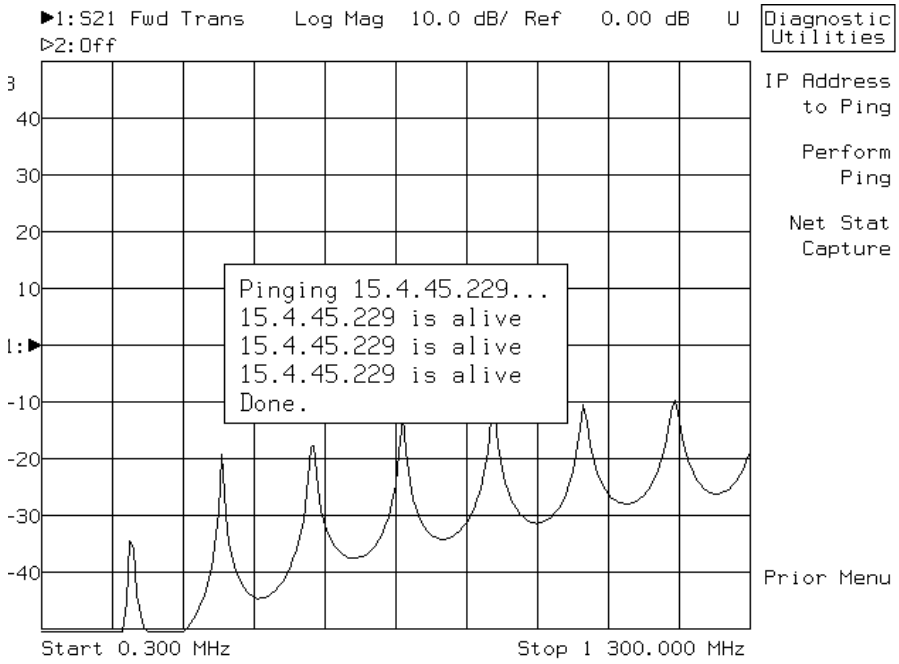
To check for connectivity to your computer or any other device on your network from your analyzer:

1. Press SYSTEM OPTIONS **LAN** **LAN Port Setup** **Diagnostic Utilities** .
2. Press **IP Address to Ping** and enter the IP address of the computer or device you are trying verify connectivity to.
3. Press **Perform Ping** .

Normal Response

A normal response after pressing **Perform Ping** is shown below. The analyzer successfully attempts four cycles of communications with the indicated network device, and displays the response time for each cycle.

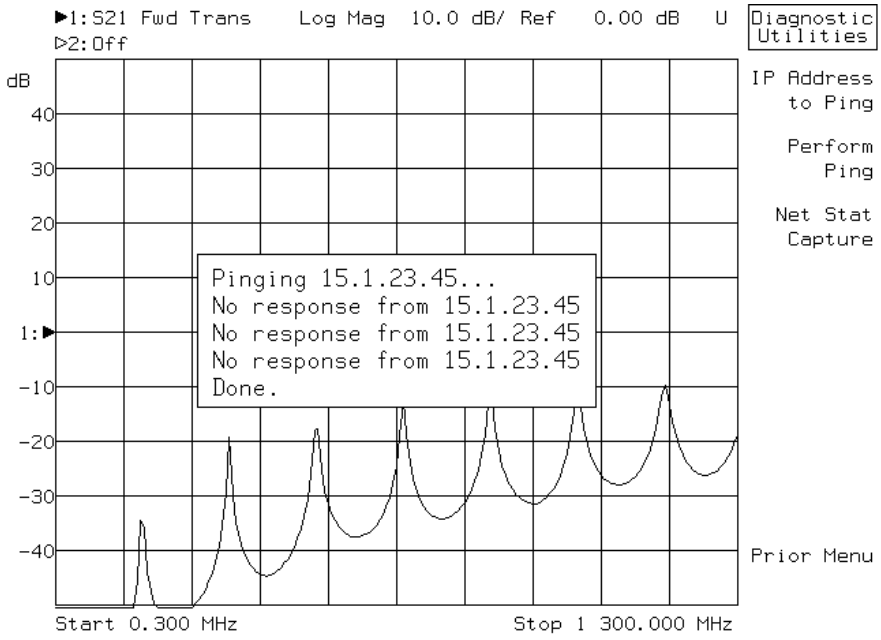
Figure 8-1 Example of a Successful Ping



Timeout Response

If communication is not established with the selected device within one second for each cycle, the display will look like this:

Figure 8-2 Example of a Failed Ping



Capturing Network Statistics

Your analyzer can capture a wide range of network statistics to help find network problems if they occur. This collection and recording is called *network statistic capturing*. It captures the network statistics at the moment that this feature is enabled, like a snapshot. If you enable this feature, the analyzer will capture network statistics in the following categories:

- TCP statistics
- IP statistics
- UDP statistics
- local routing table statistics

To enable network statistic capturing, press

(SYSTEM OPTIONS) **LAN** **LAN Port Setup** **Diagnostic Utilities** **Netstat Capture**

If you enable network statistic capturing, a file named NETSTAT.CAP will be maintained on your non-volatile RAM disk. NETSTAT.CAP is an ASCII text file containing network statistics for your analyzer. This information could be useful if network problems occur, and you should have a printed copy of this file available if you call Agilent Technologies for support with a network problem.

The following is an example of a NETSTAT.CAP file:

```
Logged Date: 1999/4/9  15:47:51
->hostShow - host names
hostname         inet address       aliases
--------         ------------       -------
HP871xE          15.4.45.2xx
localhost        127.0.0.1
nvBootLine       0.0.0.0
->ifShow "lo0" - statistic for interface lo0
lo (unit number 0):
     Flags: (0x69) UP LOOPBACK ARP RUNNING
     Internet address: 127.0.0.1
     Netmask 0xff000000 Subnetmask 0xff000000
     Metric is 0
     Maximum Transfer Unit size is 4096
```

```
     24 packets received; 24 packets sent
     0 input errors; 0 output errors
     0 collisions


->ifShow "sn0" - statistic for interface sn0
sn (unit number 0):

     Flags: (0x63) UP BROADCAST ARP RUNNING
     Internet address: 15.4.45.2xx
     Broadcast address: 15.4.47.255
     Netmask 0xff000000 Subnetmask 0xfffff800
     Ethernet address is 00:60:b0:84:xx:xx
     Metric is 0
     Maximum Transfer Unit size is 1500
     1348 packets received; 0 packets sent
     0 input errors; 0 output errors
     0 collisions

->inetstatShow - activities for internet protocol sockets
Active Internet connections (including servers)

PCB      Proto Recv-Q Send-Q  Local Address      Foreign Address    (state)
-------- ----- ------ ------  ------------------ ------------------ -------
c1fe8e8c TCP        0      0  0.0.0.0.1024       0.0.0.0.0          LISTEN
c1fe818c TCP        0      0  0.0.0.0.5025       0.0.0.0.0          LISTEN
c1fe8a0c TCP        0      0  0.0.0.0.23         0.0.0.0.0          LISTEN
c1fe8b0c TCP        0      0  0.0.0.0.80         0.0.0.0.0          LISTEN
c1fe840c TCP        0      0  0.0.0.0.998        0.0.0.0.0          LISTEN
c1fe8e0c TCP        0      0  0.0.0.0.1000       0.0.0.0.0          LISTEN
c1fe870c TCP        0      0  0.0.0.0.111        0.0.0.0.0          LISTEN
c1fe8f8c TCP        0      0  0.0.0.0.513        0.0.0.0.0          LISTEN
c1fe940c TCP        0      0  0.0.0.0.21         0.0.0.0.0          LISTEN
c1fe830c UDP        0      0  0.0.0.0.1003       0.0.0.0.0
c1fe868c UDP        0      0  0.0.0.0.111        0.0.0.0.0
c1fe860c UDP        0      0  0.0.0.0.1004       0.0.0.0.0
c1fe858c UDP        0      0  0.0.0.0.1005       0.0.0.0.0
c1fe850c UDP        0      0  0.0.0.0.1006       0.0.0.0.0
c1fe848c UDP        0      0  0.0.0.0.1007       0.0.0.0.0
c1fe838c UDP        0      0  0.0.0.0.2049       0.0.0.0.0
c1fe828c UDP        0      0  0.0.0.0.69         0.0.0.0.0
c1fe900c UDP        0      0  127.0.0.1.1024     127.0.0.1.17185
c1fe908c UDP        0      0  0.0.0.0.17185      0.0.0.0.0
->tcpstatShown - TCP statistics

TCP:

0 packet sent
0 data packet (0 byte)
0 data packet (0 byte) retransmitted
0 ack-only packet (0 delayed)
0 URG only packet
0 window probe packet
```

```
0 window update packet
0 control packet
0 packet received
0 ack (for 0 byte)
0 duplicate ack
0 ack for unsent data
0 packet (0 byte) received in-sequence
0 completely duplicate packet (0 byte)
0 packet with some dup. data (0 byte duped)
0 out-of-order packet (0 byte)
0 packet (0 byte) of data after window
0 window probe
0 window update packet
0 packet received after close
0 discarded for bad checksum
0 discarded for bad header offset field
0 discarded because packet too short
0 connection request
0 connection accept
0 connection established (including accepts)
1 connection closed (including 0 drop)
0 embryonic connection dropped
0 segment updated rtt (of 0 attempt)
0 retransmit timeout
0 connection dropped by rexmit timeout
0 persist timeout
0 keepalive timeout
0 keepalive probe sent
0 connection dropped by keepalive

->udpstatShow - UDP statistics

UDP:

368 total packets
346 input packets
22 output packets
0 incomplete header
0 bad data length field
0 bad checksum
321 broadcasts received with no ports
0 full socket

->ipstatShow - IP statistics

               total  415
              badsum    0
            tooshort    0
            toosmall    0
             badhlen    0
              badlen    0
           fragments    0
         fragdropped    0
         fragtimeout    0
             forward    0
         cantforward   67
        redirectsent    0
```

```
->icmpstatShow - ICMP statistics
ICMP:

2 calls to icmp_error
0 error not generated because old message was icmp
Output histogram:
destination unreachable: 2
0 message with bad code fields
0 message < minimum length
0 bad checksum
0 message with bad length

Input histogram:
destination unreachable: 2
0 message response generated

->arptabShow - arp entries
->mbufShow - memory buffer statistics
type        number

---------   ------

FREE    :     37
DATA    :      0
HEADER  :      0
SOCKET  :      0
PCB     :     28
RTABLE  :      3
HTABLE  :      0
ATABLE  :      0
SONAME  :      0
ZOMBIE  :      0
SOOPTS  :      0
FTABLE  :      0
RIGHTS  :      0
IFADDR  :      2
TOTAL   :     70

number of mbufs: 70

number of clusters: 4

number of interface pages: 0

number of free clusters: 4

number of times failed to find space: 0

number of times waited for space: 0

number of times drained protocols for space: 0

->routeShow - network routes
```

```
ROUTE NET TABLE
destination      gateway          flags  Refcnt  Use           Interface
------------------------------------------------------------------------
0.0.0.0          15.4.45.2xx      1      0       0             sn0
15.4.40.0        15.4.45.2xx      1      1       24            sn0
------------------------------------------------------------------------

ROUTE HOST TABLE
destination      gateway          flags  Refcnt  Use           Interface
------------------------------------------------------------------------
127.0.0.1        127.0.0.1        5      1       0             lo0
------------------------------------------------------------------------
```

Subnets and Gateways

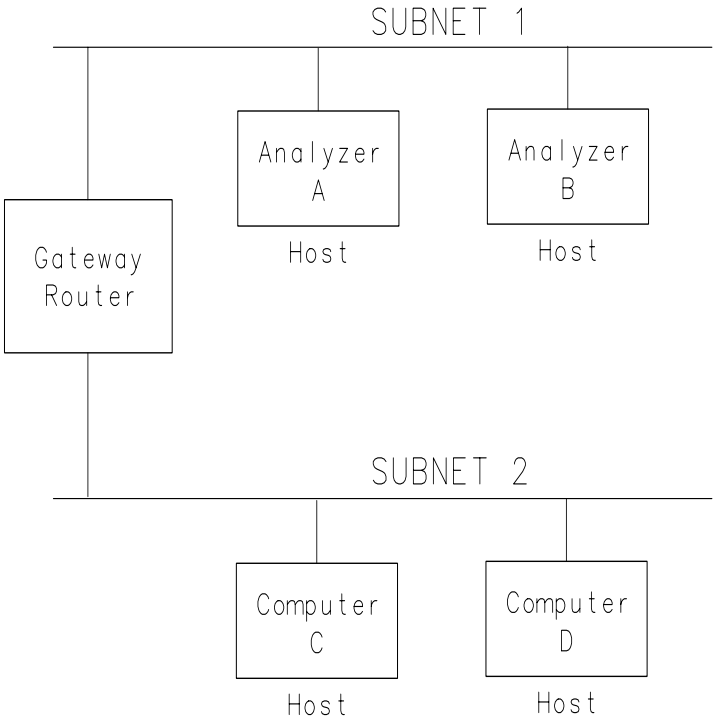
When you configure your analyzer as described in "Configuring the Analyzer" on page 1-8, you should enter a value for Gateway IP Address and Subnet Mask if they are required with your LAN. Your network administrator can tell you if you need to enter these values, and will supply you with the values to enter. This section provides some basic information on subnets and gateways.

In large systems, the LAN is often split into subnets. Each subnet is isolated from other subnets by a router. Each subnet uses a unique and contiguous range of IP addresses for its hosts. The router acts as the "gateway" between the subnets. The router decides whether or not LAN traffic is allowed from one subnet to another.

See Figure 8-3 on page 8-16. This illustration shows a portion of a LAN system that includes a router (gateway) and two subnets, each including two hosts.

Each host is a unique device (such as a computer or an analyzer) with a unique IP address. The router also has a unique IP address.

Figure 8-3 Example of a LAN with Two Subnets



In order for a host on Subnet 1 to communicate with a host on Subnet 2, there are two configuration parameters that must be set up correctly:

- Gateway IP Address — the address of the router
- Subnet Mask — a number that allows the host to determine if direct communication is allowed or whether it must communicate through a router

These parameters are used by the analyzer (host) and the router to define the IP address ranges used by each subnet, and by the router.

Refer again to Figure 8-3 for the following discussion:

You would like Analyzer A to communicate with Computer C. Note that they exist on different subnets, separated by a router.

Before Analyzer A tries to access Computer C, the analyzer looks at its subnet mask setting, and uses this mask to determine if Computer C is on the same LAN subnet.

If the analyzer determines from the subnet mask setting that Computer C is on the same subnet, then the analyzer establishes direct communication with Computer C (it sends LAN packets directly to Computer C's IP address).

If the analyzer determines that Computer C is on a different subnet from the analyzer (see Figure 8-3), then the analyzer must send LAN packets to the router's IP address. The router then forwards the packets to Computer C.

Troubleshooting Subnet Problems

If your analyzer and computer are on separate LAN segments (subnets), separated by a gateway router, and you are experiencing difficulties in communicating, try the following solutions:

- Be sure the analyzer's **Gateway IP Address** and **Subnet Mask** under the **LAN Port Setup** menu have been configured properly.

  See "Configuring the Analyzer" on page 1-8.

  Your network administrator should be able to tell you whether or not you need to enter these parameters, and should provide you with the correct numbers if you do.

- If you have configured the gateway address, your analyzer's IP address, and the subnet mask properly, but are still having problems

  1. Connect your computer and analyzer directly to each other (with no gateway routers between them). This can be done by connecting the computer and analyzer to the same subnet, or by the use of a "cross-over" cable. See "Point-to-Point Connections" on page 1-7.
  2. Configure *both* your computer and your analyzer so that they are both using a subnet mask value of 0.0.0.0, thus disabling gateway routing.
  3. Now try the ping test in both directions as described in "Troubleshooting the Initial Connection" on page 8-3. If it works, and it didn't before, you've determined that you have a problem with subnetting. Contact your network administrator for assistance.

Solutions to Common Problems

This section describes common problems you may encounter when using the analyzer on a LAN. It assumes you have been able to connect to the analyzer in the past. If this is not so, refer to the previous sections first.

If you cannot connect to the analyzer

If you suspect a bad LAN connection between your computer and analyzer, you can verify the network connection by using the ping command described earlier in this chapter or another similar echo request utility.

If a bad connection is revealed, try the following solutions:

- Make sure the analyzer is turned on.
- Check the physical connection to the LAN.
- Make sure the Internet (IP) Address of the analyzer is set up correctly in the LAN Port Setup menu. (Press SYSTEM OPTIONS **LAN** **LAN Port Setup** .)
- If the analyzer and the computer are on different networks or subnets, make sure the gateway address and subnet mask values are set correctly. See "Troubleshooting Subnet Problems" on page 8-17.

If you cannot access the file system via ftp

If you get a "connection refused" message, try the following solutions:

- If the power to the analyzer was just turned on, make sure that you wait about 25 seconds before attempting the connection.

If you get a "connection timed out" message

- Verify the LAN connection between your computer and the analyzer. Refer to "If you cannot connect to the analyzer" earlier in this section.

If you cannot telnet to the command parser port

If you get a "connection refused" message

- Try including the telnet port number (23) in the command.

If you get a "connection timed out" or "no response from host" message

- Verify the LAN connection between your computer and the analyzer. Refer to "If you cannot connect to the analyzer" earlier in this section.

If you get a "connection refused" or "no response from host" message

- If the analyzer was just turned on, make sure that you wait about 25 seconds before attempting the connection.

If you get an "operation timed-out" message

- Check the LAN connection between the computer and the analyzer. Refer to "If you cannot connect to the analyzer" in this section.
- Increase the file time-out value on your PC or workstation.

If you cannot access internal web pages or import graphic images when using a point-to-point connection

- Disable the use of proxy servers. You may have to specify this in a number of locations, depending on the operating system and software you are using.
- Disable the use of cached copies of web pages to ensure that you always get a new copy of the analyzer's screen image.

If all else fails

- Contact your network administrator.
- If you still cannot solve the problem, contact an Agilent Technologies Service Center for repair information (see Table 9-5 on page 9-11).

9 Quick Reference

EIA/TIA 568B Wiring

Table 9-1 **Straight-Through Cable (Unshielded-twisted-pair (UTP) cable with RJ-45 connectors)**

Standard, Straight-Through Wiring (each end)			
Signal Name	**RJ-45 Pin #**	**Wire Color**	**Pair #**
RX+	1	white/orange	2
RX-	2	orange	
TX+	3	white/green	3
TX-	6	green	
Not Used	4	blue	1
	5	white/blue	
	7	white/brown	4
	8	brown	

Table 9-2 **Cross-Over Cable (Unshielded-twisted-pair (UTP) cable with RJ-45 connectors)**

Cross-Over Wiring[a]			
Connector A		**Connector B**	
Signal Name	**RJ-45 Pin #**	**RJ-45 Pin #**	**Signal Name**
RX+	1	3	TX+
RX-	2	6	TX-
TX+	3	1	RX+
TX-	6	2	RX-

Cross-Over Wiring[a]			
Connector A		**Connector B**	
Signal Name	**RJ-45 Pin #**	**RJ-45 Pin #**	**Signal Name**
Not Used	4	4	Not Used
	5	5	
	7	7	
	8	8	

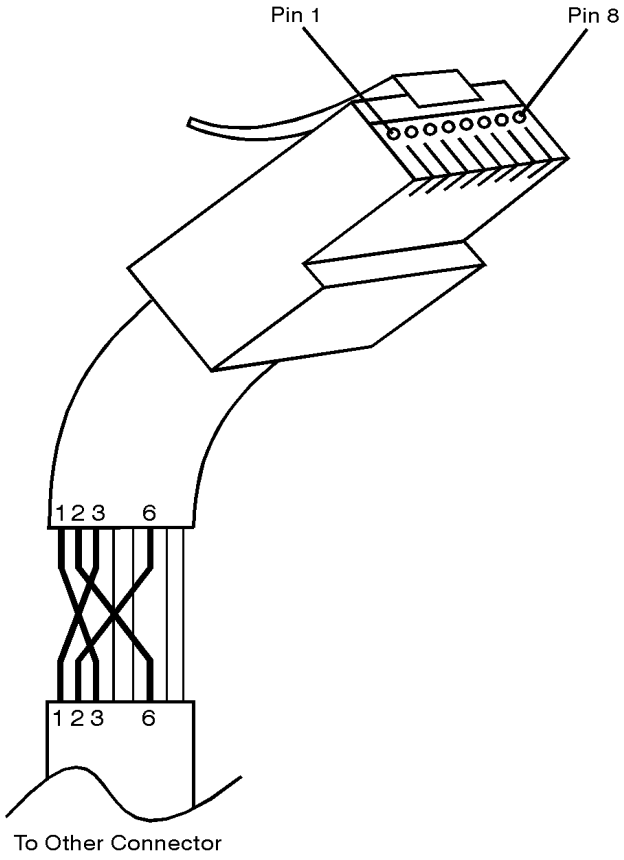
a. Either end of this cable can be used at the analyzer or LAN device. The connector names are a convention useful during cable construction only.

This cable can be used to cascade hubs or to make point-to-point connections without a LAN hub.

NOTE A convenient way to make a cross-over adapter is to use two RJ-45 *jacks* wired according to Table 9-2, above. Standard straight-through patch cables can then be used from the analyzer to the adapter, and from the adapter to other LAN devices. If you use a special-purpose adapter, you will avoid having a cross-over cable mistaken for a standard, straight-through patch cable.

NOTE Some commercially-available cross-over cables do not implement the cross-over wiring required for your analyzer. Please refer to Table 9-2, above, and verify all connections before using cables not made by Agilent Technologies.

Figure 9-1 Cross-Over Patch Cable Wiring (cross-over end)



The TELNET Command

Synopsis

`telnet` [`host` [`port`]]

Description

The `telnet` command is used to communicate with another host using the TELNET protocol. When `telnet` is invoked with `host` or `port` arguments, a connection is opened to `host`, and input is sent from the user to `host`.

NOTE Standard UNIX commands are described here. Please see your `telnet` documentation for specific information.

NOTE Only a portion of the available commands and options are described. See your `telnet` documentation for more complete information.

Options and Parameters

`telnet` operates in line-by-line mode or in character-at-a-time mode. In line-by-line mode, typed text is first echoed on the screen. When the line is completed by pressing the Enter key, the text line is then sent to `host`. In character-at-a-time mode, text is echoed to the screen and sent to `host` as it is typed.

In some cases, if your telnet connection is in “line-by-line” mode, there is no local echo. This means you will not be able to see the characters you are typing on your computer's display until *after* you press the Enter key. Only a portion of the available commands and options are described. See your telnet documentation for more complete information.

To remedy this, you need to change your telnet connection to “character-by-character” mode. This can be accomplished in most systems by escaping out of telnet to the `telnet>` prompt and then typing `mode char`. Consult your telnet program's documentation for how to change to “character-by-character” mode.

The FTP Command

Synopsis

```
ftp [-g] [-i] [-n] [-v] [server-host]
[-B DataSocketBufferSize]
```

Description

The `ftp` command is used to transfer files using the File Transfer Protocol. `ftp` transfers files over a network connection between a local machine and the remote `server-host`.

NOTE Standard UNIX commands are described here. Please see your `ftp` documentation for specific information.

NOTE Only a portion of the available commands and options are described. See your `ftp` documentation for more complete information.

Options and Parameters

When ftp is invoked with a server-host specified, a connection is opened immediately. Otherwise, ftp waits for user commands.

The following options are supported:

`-g`	disables expansion of shell metacharacters in file and directory names
`-i`	disables prompts during multiple-file operations
`-n`	disables automatic log-in
`-v`	enables verbose output
`-B`	specifies a new `DataSocketBufferSize`
`server-host`	the name or address of the remote host.

Table 9-3 lists the available user commands.

Table 9-3 ftp Commands

Command	Description
ascii	Sets the file transfer type to ASCII.
binary	Sets the file transfer type to binary.
bye	Closes the connection to the host and exits ftp.
cd *remote_directory*	Sets the working directory on the host to *remote_directory*.
delete *remote_file*	Deletes *remote_file* or empty *remote_directory*.
dir [*remote_directory*]	Lists the contents of the specified *remote_directory*. If *remote_directory* is unspecified, the contents of the current remote directory are listed.
get *remote_file* [*local_file*]	Copies *remote_file* to *local_file*. If *local_file* is unspecified, ftp uses the *remote_file* name as the *local_file* name.
help	Provides a list of ftp commands.
help *command*	Provides a brief description of *command*.
lcd [*local_directory*]	Sets the local working directory to *local_directory*.
ls [*remote_directory*]	Lists the contents of the specified *remote_directory*. If the *remote_directory* is unspecified, the contents of the current remote directory are listed.
mget *remote_file [local_file]*	Copy *remote_file* to the local system. If *local_file* is unspecified, ftp uses the *remote_file* name as the *local_file* name.
mput *local_file [remote_file]*	Copies *local_file* to remote file. If *remote_file* is unspecified, ftp uses the *local_file* name as the *remote_file* name.
put *local_file* [*remote_file*]	Copies *local_file* to *remote file*. If *remote_file* is unspecified, ftp uses the *local_file* name as the *remote_file* name.
quit	Closes the connection to the host and exits ftp.

The PING Command

Synopsis

`ping [-r] [-v] [-o] host [packetsize] [count]`

Description

The `ping` command sends an echo request packet to the `host` once per second. Each echo response packet that is returned is listed on the screen, along with the round-trip time of the echo request and echo response.

NOTE Standard UNIX commands are described here. Please see your `ping` documentation for specific information.

Options and Parameters

`-r`	Bypasses the routing tables, and sends the request directly to the host.
`-v`	Reports all packets that are received, including the response packets.
`-o`	Requests information about the network paths taken by the requests and responses.
`host`	The host name or IP address.
`packetsize`	The size of each packet (8 bytes - 4096 bytes).
`count`	The number of packets to send before ending ping ($1-(2^{31}-1)$). If `count` is not specified, `ping` sends packets until interrupted.

Dynamic Data Disk Contents

Table 9-4 Contents of the Dynamic Data Disk

File	File Type	Description
readme.txt	ASCII	This file contains a brief description of each file in this directory.
state.sta	binary	This file contains the analyzer's current instrument state settings.
cal.sta	binary	This file contains the analyzer's current calibration and instrument state settings.
data.sta	binary	This file contains the measurement data for both measurement channels.
tset_cal.cal	binary	*For use with multiport test sets only.* This file contains the test set calibration data that currently resides on the analyzer's non-volatile RAM disk.
prog.bas	ASCII	This file contains the currently loaded IBASIC program.
prog_run.bas	ASCII	This file accepts a copy of an IBASIC program, copies it to prog.bas, and immediately runs the program.
prog_run.scp	ASCII	This file accepts a copy of a file containing SCPI commands and immediately executes the commands.
screen.hgl	ASCII	This file contains the current screen image in HP-GL format. It is available for uploading to a file on your computer.
screen.gif	binary	This file contains the current screen image in GIF format. It is available for uploading to a file on your computer
screen.pcx	binary	This file contains the current screen image in PCX format. It is available for uploading to a file on your computer.
screen_m.hgl	ASCII	This file contains the current screen image, as well as the current softkey menu, in HP-GL format.
screen_m.pcx	binary	This file contains the current screen image, as well as the current softkey menu, in PCX format.

File	File Type	Description
screen_m.gif	binary	This file contains the current screen image, as well as the current softkey menu, in GIF format.
parm_all.txt	ASCII	This file contains a listing of all of the instrument's operating parameters in ASCII text format.
parm_screen.txt	ASCII	This file contains the information in the current operating parameters screen in ASCII text format.
trace1.prn	ASCII	This file contains the measurement channel 1 measurement data in ASCII spreadsheet format.
trace2.prn	ASCII	This file contains the measurement channel 2 measurement data in ASCII spreadsheet format.
trace1.s1p	ASCII	This file contains the measurement channel 1 measurement data in Touchstone format.
trace2.s1p	ASCII	This file contains the measurement channel 2 measurement data in Touchstone format.

Agilent Technologies Sales and Service Offices

Table 9-5 Sales and Service Offices

UNITED STATES		
Instrument Support Center Agilent Technologies, Inc. (800) 403-0801		
EUROPEAN FIELD OPERATIONS		
Headquarters Agilent Technologies S.A. 150, Route du Nant-d'Avril 1217 Meyrin 2/ Geneva Switzerland (41 22) 780.8111	France Agilent Technologies France 1 Avenue Du Canada Zone D'Activite De Courtaboeuf F-91947 Les Ulis Cedex France (33 1) 69 82 60 60	Germany Agilent Technologies GmbH Agilent Technologies Strasse 61352 Bad Homburg v.d.H Germany (49 6172) 16-0
Great Britain Agilent Technologies Eskdale Road, Winnersh Triangle Wokingham, Berkshire RG41 5DZ England (44 118) 9696622		

Continued on next page.

INTERCON FIELD OPERATIONS		
Headquarters Agilent Technologies 3495 Deer Creek Rd. Palo Alto, CA 94304-1316 USA (650) 857-5027	Australia Agilent Technologies Australia Ltd. 31-41 Joseph Street Blackburn, Victoria 3130 (61 3) 895-2895	Canada Agilent Technologies (Canada) Ltd. 17500 South Service Road Trans-Canada Highway Kirkland, Quebec H9J 2X8 Canada (514) 697-4232
Japan Agilent Technologies Japan, Ltd. Measurement Assistance Center 9-1, Takakura-Cho, Hachioji-Shi Tokyo 192-8510, Japan TEL (81) -426-56-7832 FAX (81) -426-56-7840	Singapore Agilent Technologies Singapore (Pte.) Ltd. 150 Beach Road #29-00 Gateway West Singapore 0718 (65) 291-9088	Taiwan Agilent Technologies Taiwan 8th Floor, H-P Building 337 Fu Hsing North Road Taipei, Taiwan (886 2) 712-0404
China China Agilent Technologies 38 Bei San Huan X1 Road Shuang Yu Shu Hai Dian District Beijing, China (86 1) 256-6888		

Glossary

Glossary

10Base-T A physical network connection that uses twisted-pair cables with RJ-45 connectors.

absolute pathname (The specification of a node (file or directory) in a hierchical file system relative to the root directory (the topmost node)—It is the full path name of a file or directory, including all the directories leading to it, starting with the root (/) and ending with the file or directory name itself.

authentication The verification of an entity (person or process) for the purpose of granting access to files or directories, or to verify the source of a message.

Automount The automatic mounting of a remote file system.

BOOTP (Bootstrap Protocol) A protocol for passing configuration information on a TCP/IP network. Specified by RFC-951.

bridge A device that moves traffic from one network to another.You use a bridge to connect networks of the same type together.

datagram An independent piece of data comprising sufficient information to be routed to the destination, without reliance on previous messages.

client A computer system or process that requests services from a server.

Depending on the protocol, your analyzer is either a client or a server:

Protocol	Function
BOOTP	client
FTP	server
NFS	client
SICL LAN	server

client/server A distributed computing system, with tasks split between a client and a server. Clients send requests to servers, asking for information or actions.

The client and server must share a common protocol for client-server communication to occur.

Ethernet A network that adheres to the IEEE 802.3 Local Area Network standard.

Ethernet address A hexadecimal number which is used to identify a machine on a network. Each analyzer is assigned a unique Ethernet address at the factory and it is stored in the analyzer's ROM.

Ethertwist See **10Base-T**.

FTP (File Transfer Protocol) A service that allows you to remotely transfer files among different operating systems.

ftp (File Transfer Program) A file transfer program that uses file transfer protocol.

gateway A generic term usually referring to a **router**.

group ID In UNIX, a unique number between 0 and 32767 that identifies a set of users. Each file has a group ID associated with it, to indicate the group to which its group permissions apply. Group IDs are stored in `/etc/passwd` and `/etc/group` databases.

host A computer or device on a network.

host name A unique name that is used to identify each host machine on a network. The host name is created by the user or the system administrator. The hostname is directly linked to a specific IP address, and can usually be used in place of the harder-to-remember IP address.

http (HyperText Transfer Protocol) A protocol used to carry World Wide Web (WWW) traffic.

internet The connection of two or more distinct networks. Often a gateway or router is used to make the connection.

Internet The largest **internet** (see above) in the world, connecting millions of networks. The Internet uses the TCP/IP protocol.

IP address (Internet Protocol Address) A unique number that is assigned to each device which is to be connected to a TCP/IP network. Before using your analyzer on a network, your network administrator will need to assign an IP address.

An IP address consists of a 32-bit value presented in decimal dot notation: 4 octets (bytes) separated by a dot.

For example, the binary address 10000000 00000111 00001111 00000001 has the decimal dot notation of 128.7.15.1

network administrator

Similar to "system administrator".

network logging The collection and recording of network performance measures and other parameters and statistics.

NFS (Network File System) A standard network protocol for file sharing among different operating systems.

ping A utility that allows you to determine the status of the connections between devices and a network. The ping utility is usually included with software packages that provide networking services. Your analyzer has a ping utility included in its firmware.

protocol A set of conventions that specify how information will be formatted and transmitted on a network, and how machines on a network will communicate.

relative pathname The specification of a node (file or directory) in a hierarchical file system relative to the current (context-dependent) directory.

RPC (Remote Procedure Call) A client/server protocol used to obtain a service from a remote process, over a network. RPC is a fundamental part of NFS.

router A device that moves traffic from one network to another. Routers are used to connect different types of networks together.

server A device that is configured to provide a service to other devices on a network, such as shared access to a file system or a printer. See client.

Depending on the protocol, your analyzer is either a client or a server:

Protocol	Function
BOOTP	client
FTP	server
NFS	client
SICL LAN	server

SICL LAN A LAN protocol using the Standard Instrument Control Library (SICL). It provides control of instruments over the LAN, using a variety of computing platforms, I/O interfaces and operating systems.

socket An endpoint for communication over a network. A socket, consisting of a port number and a network address, is part of a mechanism for creating a virtual connection between two processes.

subnet mask (Also called an *address mask*) A bit mask that identifies the bits corresponding to the network address and subnet address portions of the IP address. The mask has ones in positions of the IP address corresponding to the network and subnet addresses, and zeros in the host address positions.

system administrator A person who manages systems and machines on a network. The system administrator is responsible for installing software and hardware on the network and assigning addresses and names to machines.

TCP/IP (Transmission Control Protocol/Internet Protocol) A set of standards for communications between computers and between networks.

TFTP (Trivial File Transfer Protocol Protocol) A very simple protocol for file transfer over a network. TFTP uses an unreliable data protocol without user identification or directory visibility. TFTP is specified by RFC 1350.

telnet A protocol that allows users to create a session to run programs on or transfer information to and from a remote computer.

ThinLAN A physical network connection that uses coax cables with BNC connectors.

time out A period of system inactivity during which the system awaits user or network response. If there is no response by the end of the period, the system takes an action.

UDP (User Datagram Protocol) A protocol for passing data on an IP network. UDP does not guarantee delivery, and does not require a connection. It is a lightweight and efficient protocol, but all error processing and retransmission of data must be done by the application program. Specified by RFC-768.

VISA (Virtual Instrument Software Architecture) VISA is an I/O library used to develop I/O applications and instrument drivers that comply with *plug&play* standards. Applications and instrument drivers developed with VISA work on any system that has the VISA I/O layer. This allows software from many sources to work together.

Index

Index

Index